



Informatyka

Jak zaprojektować stronę WWW? Do czego przydaje się dokumentacja projektu? Jak zaprogramować aplikację? Po odpowiedzi na te pytania zapraszamy do e-podręcznika.

- [Informacje o projektowej koncepcji podręcznika](#)
- [Technologie informatyczne](#)
- [Projekt stanowiska pracy i wybór ról w projekcie](#)
- [Dokumentacja projektu \(edycja dokumentów\)](#)
- [Programowanie aplikacji](#)
- [Ćwiczenie – Przykład Przewodnika](#)
- [Analiza danych aplikacji \(Bazy danych\)](#)
- [Strona internetowa projektu \(Internet\)](#)
- [Słowniczek](#)
- [Pojęcia](#)

Informacje o projektowej koncepcji podręcznika

Wstęp

Koncepcja tego podręcznika polega na zastosowaniu metodologii nauki przez realizację projektu (ang. Project Based Learning, PBL). Project-Based Learning (PBL) jest alternatywnym sposobem prowadzenia zajęć szkolnych. Taka forma pozwala na przyswojenie szerokiej bazy wiedzy, poprawiając zarazem komunikację, i umożliwia nabycie umiejętności społecznych i interpersonalnych. Efektem proponowanej metody jest wykorzystywanie zdobytej wiedzy do realizacji konkretnych zadań, a nie nauczanie się definicji i formułek. Planowanie i przeglądanie informacji, w celu znalezienia rozwiązania konkretnego problemu, pozwala lepiej zrozumieć daną problematykę (jak powiedział Konfucjusz „Powiedz mi, a zapomnę. Pokaż mi, a zapamiętam. Pozwól mi zrobić, a zrozumiem”).

Wszystkie lekcje będą miały postać jednego spójnego projektu aplikacji interaktywnej (np. gry komputerowej lub innej aplikacji multimedialnej o tematyce zaproponowanej przez was i uzgodnionej z nauczycielem) rozplanowanego na 30 godzin, który będzie zawierał wszystkie treści programowe, jakie należy znać na tym etapie edukacji.

Przygotowanie stanowiska pracy

Już wiesz

Z podręcznika do gimnazjum, przypomnijcie sobie rozdział:
”Parametry wpływające na ergonomię pracy z komputerem”.

Ćwiczenie 1

Przygotujcie swoje stanowiska pracy z komputerem. Zastanówcie się, na co należy zwrócić uwagę. Przedyskutujcie z kolegami, jakie mogą wystąpić zagrożenia i skutki niewłaściwego korzystania z komputera.

Wskazówka

Można zwiększyć komfort pracy z komputerem, dostosowując niektóre parametry urządzeń do własnych predyspozycji, takich jak wzrok, właściwości ruchowe ręki (prawej lub lewej) lub ostrość słuchu.

Praca przy komputerze w niewłaściwych warunkach może być powodem dolegliwości i chorób.

Technologie informatyczne

Wstęp

Pierwszym etapem projektu realizowanego w ramach tego podręcznika będzie wybór tematyki aplikacji. W zależności od dokonanego wyboru, dalsze zadania wykonywane w ramach projektu mogą nieznacznie różnić się od siebie.

Już wiesz

Z podręcznika dla szkoły podstawowej przypomnijcie sobie następujące rozdziały:

- ”Pojęcie i rodzaje systemów operacyjnych”,
- ”Elementy zestawu komputerowego”,
- ”Podstawowe działania w systemach operacyjnych”.

Temat projektu

Zastanówcie się, jaką aplikację interaktywną będziecie realizować w ramach zajęć. Może być to gra komputerowa, kalkulator lub aplikacja multimedialna prezentująca wybrane zagadnienia, np. z historii, biologii lub geografii. Przedyskutujcie z nauczycielem wasze pomysły i propozycje na ten temat.

Propozycje:

- [Gra komputerowa](#),
- [Multimedialny przewodnik geograficzny](#),
- [Symulator lotów kosmicznych](#).

Sposób realizacji projektu zostanie omówiony na przykładzie aplikacji multimedialnej w formie przewodnika po miastach Polski. W przypadku realizacji zadania przez jedną osobę można go wykorzystać jako szablon do własnej aplikacji. Przy pracy w grupie możliwa będzie realizacja bardziej rozbudowanej aplikacji. Praca w grupie może więc być bardziej wydajna i efektywna. Możliwość koncentracji na wąziej zdefiniowanych zagadnieniach, wykorzystanie ukierunkowanych zdolności członków grupy jak i wzajemna mobilizacja działań sprzyja powstawaniu lepszych projektów. Niezależnie od sposobu realizacji projektu, każde z zadań powinno obejmować:

- tworzenie dokumentacji projektu,
- prezentację koncepcji projektu,

- opracowanie systemu wymiany informacji pomiędzy członkami grupy, lub pomiędzy uczniem i nauczycielem (wykorzystanie chmury),
- projektowanie interfejsu aplikacji,
- programowanie aplikacji,
- analizę danych (bazy danych),
- projekt strony internetowej promującej aplikację,
- stworzenie promocji aplikacji w Internecie.

Zdefiniowanie podzadań

Przystępując do wykonania projektu, należy zdefiniować niezbędne zasoby, które będą potrzebne w dalszym etapie projektowania. Należy wybrać platformę informatyczną, a więc rodzaj komputera, oraz system operacyjny, który umożliwi jego obsługę, a także potrzebne urządzenia peryferyjne (skaner, kamera, aparat cyfrowy lub inne), które będą potrzebne do realizacji projektu. Przy pracy w grupie ważne jest też określenie zakresu prac, który powinien odpowiadać liczbie osób w zespole.

Liczba członków zespołu zależy zwykle od wielkości i stopnia skomplikowania zadania. Planując projekt informatyczny, należy opracować strukturę, która powinna powstać na podstawie rozdzielenia zadań dla pojedynczych członków zespołu w zadanym czasie (dla całego projektu mamy semestr).

W przypadku szacowania zadań profesjonalnego projektu informatycznego wprowadza się:

- czas trwania projektu - realny czas w którym projekt powinien zostać ukończony (najczęściej podany w miesiącach)
- nakład pracy – ilość środków potrzebnych na zrealizowanie zadania (najczęściej mierzone w osobo-miesiącach)
- obciążenie ludzi – ile osób w funkcji czasu jest przydzielona do projektu

Koncepcję aplikacji i wybór platformy informatycznej dla projektu, należy zdefiniować odpowiadając na pytania określające projekt, które będą definiować kontekst i sposób użycia aplikacji.

1. OKREŚLENIE CELU – WIZJA TWORZONEGO SYSTEMU

O czym ma być projektowany system?

Jakie są cele tworzonego projektu?

1. FORMUŁOWANIE SZCZEGÓŁOWYCH ZADAŃ

Jak można zdefiniować dobrze wykonaną podobną aplikację?

Co będzie sukcesem wykonywanej aplikacji? W jaki sposób można stwierdzić, że odniosła sukces?

Jak można opisać aplikację?

Jakie są cele ze względu na właściciela?

Jakie są cele ze względu na użytkownika?

1. OKREŚLENIE ODBIORCÓW – UŻYTKOWNIKÓW

Kto będzie użytkownikiem aplikacji?

Jak można opisać przeciętnego użytkownika? (np., wiek, doświadczenie, wykształcenie, itp.)

Dlaczego użytkownicy będą używać aplikacji? (Potrzeby użytkowników, zainteresowania i cele)

W jaki sposób użytkownicy będą mieli dostęp do aplikacji (np., prędkość łącza, rozdzielczość, itp.)

1. PRZEPROWADZENIE ANALIZY ZADAŃ I PRIORYTETÓW ZADAŃ

Co użytkownicy robią z aplikacją?

Jakie zadania mają decydujące znaczenie dla sukcesu projektowanego systemu?

Jakie zadania są najistotniejsze dla tworzonego systemu?

Które funkcje najczęściej będą używane przez użytkowników?

Jakie zadania mają decydujące znaczenie dla sukcesu aplikacji?

Jak często aplikacja będzie uruchamiana przez tego samego użytkownika?

Co skłoni użytkownika do ponownego uruchomienia aplikacji?

1. OKREŚLENIE MIERZALNYCH WARTOŚCI TWORZONEJ APLIKACJI

*Jakie zadania będą mogli wykonać użytkownicy z najmniejszą ilością błędów ?
(wydajność)*

*Jakie zadania użytkownicy będą mogli wykonać jak najszybciej i najsprawniej?
(efektywność) Jaki poziom zadowolenia będą mieli użytkownicy obsługujący aplikację?
(zadowolenie)*

2. DYSKUSJA NA TEMAT OCZEKIWAŃ, WYMAGAŃ I PREFERENCJI

*Co powinna robić aplikacja? Jak powinna wyglądać aplikacja ?iEva2MpCjj_d374e376Czy istnieją jakieś ograniczenia i ścisłe wytyczne dotyczące aplikacji?
iEva2MpCjj_d374e376Czy istnieją aplikacje na których można się wzorować ?*

(ewentualnie można wymienić te aplikacje) iEVa2MpCjj_d374e376 Jakie będą podstawowe cechy i atrybuty tworzonej aplikacji? iEVa2MpCjj_d374e376

3. IDENTYFIKACJA DOSTĘPNYCH ZASOBÓW

Czy i jak stworzona aplikacja będzie aktualizowana i konserwowana? Kto zajmuje się umieszczaniem treści? Kto jest odpowiedzialny za aktualizację treści? Kto jest odpowiedzialny za promocję stworzonej aplikacji?

4. POTREZBY TECHNOLOGICZNE

Określenie potrzeb hostingowych? Zakup lub zaadoptowanie istniejącej domeny? W jakiej technologii zostanie wykonana aplikacja? Jakie dodatkowe technologie będą potrzebne ze względu na sposób działania aplikacji? Czy wykorzystane biblioteki i programy wymagają nakładu finansowego, czy licencja zobowiązuje nas do uiszczenia, w przypadku sukcesu komercyjnego, opłat licencyjnych ?

iEVa2MpCjj_d374e426

Odpowiedzi na te pytania będą stanowiły podstawę dalszych działań. Niech będą one drogowskazem dalszego działania. Ważne jest, by na każdym etapie pamiętać o postawionych celach, a wcześniej określone potrzeby pozwolą na odpowiednie zaprojektowanie stanowiska pracy.

Ponieważ projekt jest realizowany w grupie, odpowiedzi na pytania powinny zostać sformułowane na podstawie dyskusji wszystkich jej członków (*burza mózgów*)

Rodzaje licencji

Każdego użytkownika programu komputerowego obowiązują ściśle określone zasady, o których stanowi licencja - umowa między producentem (autorem lub wydawcą), a użytkownikiem, dotycząca zasad użytkowania produktu. Do najpopularniejszych licencji należą:

Freeware - są to programy, których można używać i rozpowszechniać za darmo. Nie wolno ich jednak sprzedawać, ani dokonywać w nich zmian, umieszczając wewnątrz na przykład własną reklamę. Mogą być rozpowszechniane wyłącznie w niezmienionej formie. Zazwyczaj aplikacje te nie nakładają na użytkownika obowiązku rejestracji. Niektóre programy posiadają jeszcze dodatkowo rozbudowaną wersję Pro, za którą trzeba już zapłacić.

Public domain - licencja „dla dobra publicznego”, czyniąca z oprogramowania własność ogółu, w myśl której autor lub autorzy oprogramowania zrzekają się praw do upowszechniania oprogramowania na rzecz ogółu użytkowników. Dozwolona jest dalsza dystrybucja oprogramowania bez zgody autora

Shareware - oprogramowanie udostępniane bezpłatnie do testów. Każdy potencjalny nabywca, przed podjęciem decyzji o zakupie, może gruntownie sprawdzić działanie,

zazwyczaj w pełni funkcjonalnej, wersji.

Licencja GNU GPL (General Public Licence) lub po prostu GPL – zasady licencyjne określone przez konsorcjum Free Software Foundation. Jeśli ktoś wprowadza do obiegu oprogramowanie zawierające jakąkolwiek część podlegającą licencji GPL, to musi udostępnić wraz z każdą dystrybucją binarną jej postać źródłową.

Źródło: Piotr Grzyb „Licencjonowanie Oprogramowania” Portal php.pl – Marzec 2004

Projekt stanowiska pracy i wybór ról w projekcie

Wstęp

W zależności od tematyki zadania potrzebne będą odmienne urządzenia współpracujące z komputerem (np. fotograficzny aparat cyfrowy, kamera cyfrowa, czytniki). Biorąc pod uwagę zdefiniowane wcześniej podstawowe potrzeby, należy określić, co powinno zawierać stanowisko pracy. W przypadku przewodnika multimedialnego, wykorzystane zostaną materiały z Internetu, dlatego bardzo istotny jest dostęp do Internetu.

Wykorzystywany Processing jest zaimplementowany w języku Java. Powstały kod jest niezależny od systemu operacyjnego i procesora, a wykonuje go tzw. wirtualna maszyna Javy, która tłumaczy kod uniwersalny na kod dostosowany do specyfiki konkretnego systemu operacyjnego i procesora. W tej chwili wirtualna maszyna Javy jest już dostępna dla większości systemów operacyjnych i procesorów. Wybór Processingu pozwala na stworzenie aplikacji międzyplatformowej (działającej w systemach Windows, Linux i Mac OS).

Już na tym etapie należy wyszukać i wybrać źródła informacji z różnych dziedzin niezbędne do wykonania projektu. Dla wybranego przykładu – przewodnik multimedialny należy:

- wyznaczyć miasta, które będą opisane w przewodniku (Gdańsk, Wrocław, Poznań, Łódź, Warszawa, Kraków),
- określić rysunki, które powinny zostać wykonane dla aplikacji, ikony programu,

Przed przystąpieniem do wykonywania projektu, należy na papierze naszkicować wygląd ekranu i przedyskutować działanie aplikacji. Pierwszy szkic pozwoli na określenie programów potrzebnych do realizacji zadania (programy do obróbki grafiki wektorowej i rastrowej i program do wykonania interakcji wraz z niezbędnymi [bibliotekami](#)).

Ważne!

Jeśli będą pobierane grafiki lub utwory muzyczne z Internetu, należy zwrócić szczególną uwagę na przestrzeganie praw autorskich. Należy korzystać z darmowych źródeł i pamiętać o podaniu, skąd pochodzą pobrane pliki.

Główne zadania w projekcie – Przewodnik Multimedialny

- przygotowanie treści (opis miasta) i ilustracji (zdjęcia i filmy ciekawych obiektów charakterystycznych dla każdego miasta),
- projekt interfejsu,
- przygotowanie ikon i elementów interfejsu,
- programowanie zadań aplikacji ,
- prezentacja projektu,
- projekt strony Internetowej promującej przewodnik.

Po ustaleniu zadań i programów do wykonania można przydzielić role członkom grupy i przydzielić im odpowiednie zadania związane z przypisanymi rolami.

Kolejne podzadanie polega na podziale ról pełnionych w projekcie. Umiejętne zbudowanie zespołu, składającego się z osób równomiernie obciążonych zadaniami będzie decydować o powodzeniu całego przedsięwzięcia.

W przypadku realizacji projektu w grupie ważnym staje się tzw. [Praca grupowa](#)

Jednym z istotnych elementów jest przypisanie ról członkom zespołu. Według Modelu Belblina, – na podstawie doświadczeń, można przypisać członkom zespołu projektowego jedną z 9 ról :

- **Kreator**, tworzy nowe propozycje i rozwiązuje problemy. Osoba taka powinna wyróżniać się inteligencją i kreatywnością. Dzięki niemu możliwe jest inicjowanie nowych etapów projektu
- **Ewaluator**, potrafi sprawnie analizować postawione problemy, rozwija koncepcje. Osoba taka powinna być opanowana i pragmatyczna
- **Koordynator**, odpowiada za zespół. Potrafi tworzyć nie porozumienia pomiędzy wszystkimi członkami zespołu, odnajdując potencjał w każdym z członków grupy w kontekście postawionych wyzwań. Osoba taka powinna być entuzjastyczna, ufna i opanowana.
- **Implementer**, bardzo sumiennie wykonuje postawione zadania. Osoba taka powinna być zorganizowana, sumienna i bardzo praktyczna.
- **Perfekcjonista**, sprawdza każdy szczegół, pilnuje harmonogramu i wymusza na pozostałych członkach zespołu trzymania się ram czasowych. Osoba taka powinna być opanowana, zdyscyplinowana o dużej świadomości postawionych zadań.
- **Lokomotywa**, wprowadza zmiany bez wchodzenia w szczegóły interpersonalne. Osoba taka powinna być odporna na stres, efektywna, towarzysząca i lubiana przez grupę.
- **Dusza Zespołu**, zapobiega konfliktom pomiędzy członkami zespołu. Osoba taka powinna być komunikatywna i entuzjastyczna, ufna i przyjazna.
- **Poszukiwacz Źródeł**, szuka rozwiązań w źródłach zewnętrznych. Prowadzi negocjacje i zajmuje się kontaktami zewnętrznymi. Osoba taka powinna być zaradna, komunikatywna i towarzysząca.
- **Specjalista**, posiada wiedzę niezbędną do realizacji postawionych zadań. Osoba taka powinna posiadać umysł analityczny, powinna być rozważna i zaangażowana w swoje działania.

Każdy w projekcie może opisać swoje zadania za pomocą określonej funkcji. Funkcje mogą się nieznacznie różnić w zależności od wybranego wariantu. Wykaz funkcji związanych z wykonywaniem zadań może obejmować: [grafika](#), [webdesignera](#), [kreatora treści](#), [programistę](#) czy [projektanta systemu](#).

Przy specyfikacji zadań i ich rozdziale należy zwrócić szczególną uwagę na następujące problemy:

- podstawowe obowiązki poszczególnych członków zespołu,
- czy istnieje specjalna odpowiedzialność za sprzęt,
- bezpieczeństwo innych,
- główne trudności i uciążliwości,
- z jakimi osobami będzie mieć kontakt osoba wykonująca zadanie,
- zakładany stopień nadzoru,
- swobody w podejmowaniu decyzji,
- czy istnieją inne niedogodności tej pracy.

Ćwiczenie 1

Proszę przypisać kilka z wyżej postawionych ról, kolegom i koleżankom z klasy. Proszę również opisać siebie, cechy charakterystyczne swojej osoby. Następnie proszę porównać swoje propozycje z propozycjami kolegów z klasy.

Wskazówka

W literaturze opisany jest szczegółowo ten model w książce M. Belbin, „Twoja rola w zespole”, Wydawnictwo GWP, Gdańsk 2002

Mając określone cechy charakterologiczne i umiejętności można przystąpić do specyfikacji i rozdziału zadań. Należy przydzielić zadania wszystkim członkom grupy oraz określić ich odpowiedzialność. Należy założyć, kto jest odpowiedzialny za sprzęt. Ważne jest by planując role i zadania uwzględnić bezpieczeństwo wszystkich członków grupy.

Większość zadań związanych z koordynacją działań można przydzielić nauczycielowi.

Uczniowie będą przede wszystkim wykonawcami, którym w zależności od predyspozycji i zdolności można przypisać odpowiednie role. Poniżej znajduje się przykładowa tabela ról dla Projektu – **Przewodnik Multimedialny**.

Nazwisko	Imie	Funkcja
Imię 1	Nazwisko 1	Grafik
Imię 2	Nazwisko 2	Webdesigner
Imię 3	Nazwisko 3	Programista
Imię 4	Nazwisko 4	projektant systemu

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Kryteria oceny projektu:

	Grafika	Pomysł i oryginalność	Implementacja	ocena zespołu
wyjaśnienie	Ocena estetyki, doboru grafik i optymalizacji pod względem urządzenia.	Bardzo ważna jest kreatywność, pomysłowość i oryginalność prezentowanego rozwiązania	Oceniane jest tu działanie aplikacji – czy jest wolne od błędów jak i sposób implementacji – przejrzystość kodu	Członkowie zespołu powinni wystawić sobie nawzajem oceny.
wagi	10%	10%	10%	20%

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Dokumentacja projektu (edycja dokumentów)

Wstęp

Już wiesz

Przypomnijcie sobie rozdział:
"Opracowywanie tekstów za pomocą komputera"
z podręcznika do gimnazjum

Aby sprawnie realizować kolejne etapy wybranego projektu, należy zastanowić się nad **harmonogramem pracy**, podziałem obowiązków i sposobem komunikacji w zespole. W tym celu należy za pomocą pakietu aplikacji biurowych utworzyć dokumentację projektu. Można wykonać także prezentację multimedialną w postaci pokazu slajdów, aby przedstawić projekt pozostałym osobom w klasie. Powinno się również skorzystać z technologii chmury, aby współdzielić utworzone pliki ze wszystkimi członkami grupy.

Wykonanie dokumentacji projektowej

Dokumentacja projektowa powinna zawierać zakres i podział prac oraz ich harmonogram. Może ona obejmować:

- plany, harmonogramy, szacunki;
- diagramy, schematy i algorytmy, związane z fazami realizacji zadań,
- raporty;
- kod źródłowy;
- wymianę informacji między członkami zespołu na temat realizacji poszczególnych zadań.

Dokumentacja powinna powstać na bazie pytań i odpowiedzi sformułowanych na początku zajęć.

Wybór formatu pliku, tworzenie i zapis dokumentów

Już wiesz

Z podręcznika do gimnazjum przypomnijcie sobie rozdziały:
"Zapisywanie dokumentu tekstowego w różnych formatach",

"Opracowywanie dokumentów kilkustronicowych".

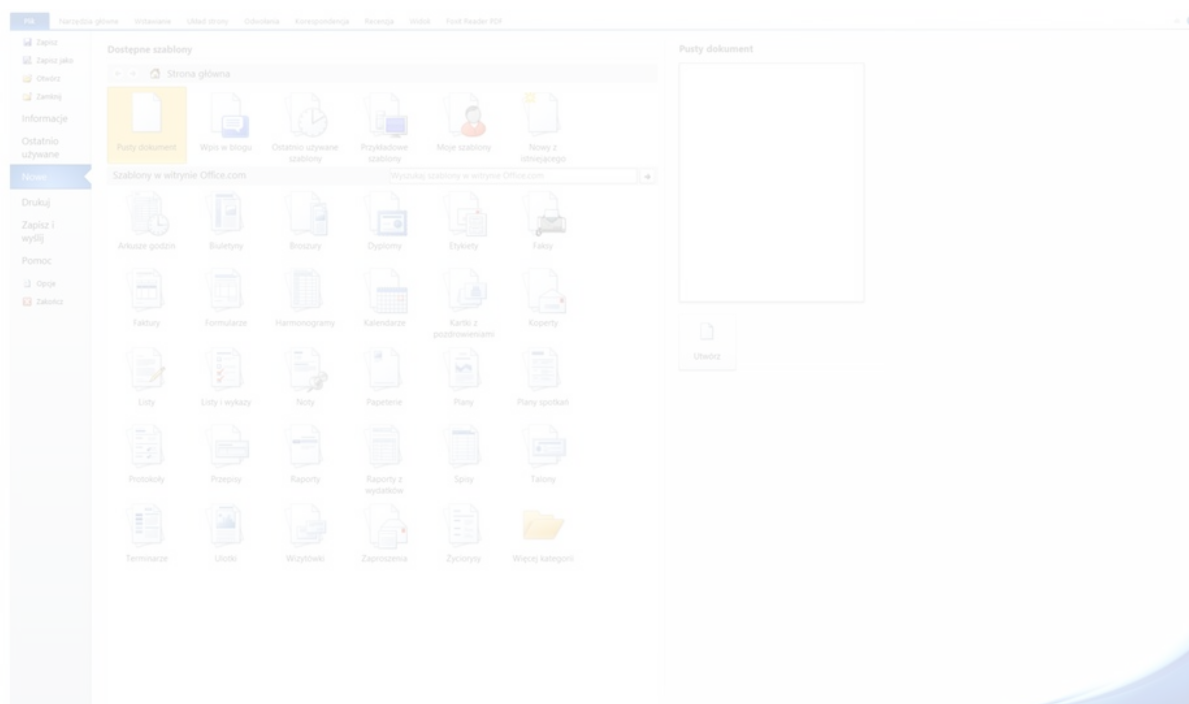
W celu sprawnej komunikacji i wymiany danych należy ustalić format pliku, w jakim będzie zapisana dokumentacja projektowa.

Kolejne rozdziały dokumentacji projektowej powinny powstawać na podstawie stworzonej przez nas wcześniej listy pytań i zadań.

W celu stworzenia dokumentacji dla przewodnika multimedialnego należy przeanalizować odpowiedzi na pytania z rozdziału „Zdefiniowanie podzadań”. Na podstawie dyskusji z kolegami z grupy lub po samodzielnej analizie, odpowiedzi na pytania powinny zwięźle scharakteryzować projektowany program komputerowy. W przypadku działania w grupie należy przydzielić zadania każdemu z członków grupy. W przypadku działania samodzielnego należy jedynie zaplanować kolejne działania w celu poprawnej realizacji zadania.

Popularne edytory tekstu dostarczają szereg różnorodnych szablonów, umożliwiających szybkie tworzenie dokumentów. Szablony zawierają wstępnie skonfigurowane ustawienia, takie jak tekst, liniatury, ustawienia formatowania oraz układy stron, dzięki czemu umożliwiają szybsze tworzenie dokumentów niż w przypadku rozpoczynania od pustego arkusza.

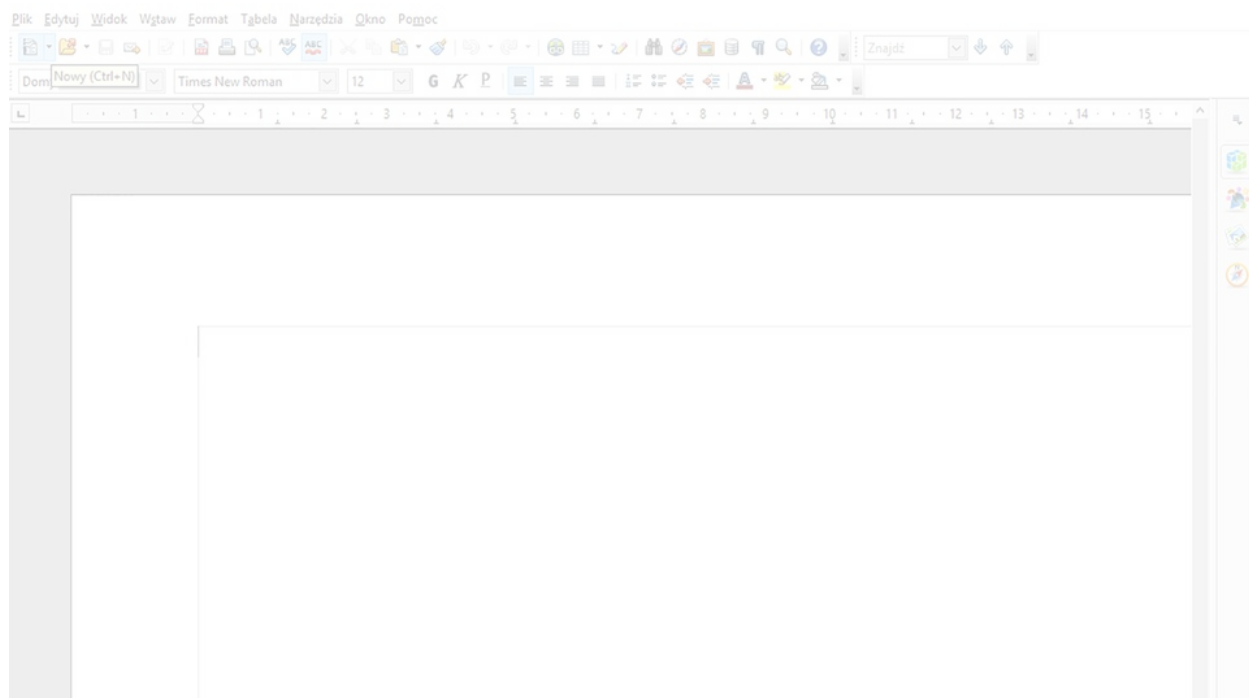
Przykład 1



Zasób interaktywny dostępny pod adresem <https://zpe.gov.pl/a/DUZA12PPC>

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Przykład 2



Zasób interaktywny dostępny pod adresem <https://zpe.gov.pl/a/DUZA12PPC>

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Dokumentacja zazwyczaj zawiera następujące składniki:

- Cel Projektu.
- Harmonogram Projektu.
- Organizacja Projektu.
- Formuła realizacji.

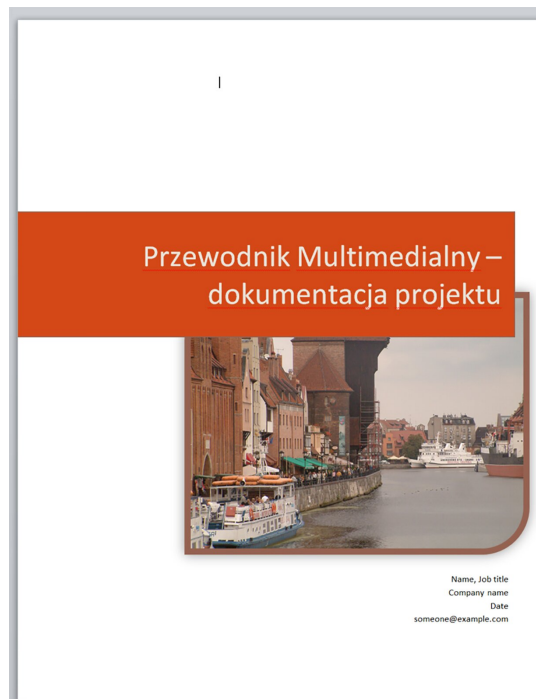
Na podstawie powyższych punktów należy stworzyć rozdziały dokumentu projektowego wykorzystując do tego odpowiednie [style](#) (dla każdego punktu będzie to nagłówek).

Na podstawie odpowiednio przygotowanego formatu nagłówków można stworzyć automatycznie spis treści na początku dokumentu projektu.

Spis treści zostaje utworzony na podstawie stylu nagłówków. Na podstawie wybranych nagłówków, zostaje sformatowany tekst haseł spisu treści.

W dokumencie utworzonym za pomocą szablonu w miejscach przeznaczonych do wprowadzania tekstu jest umieszczony przykładowy tekst. Wystarczy podmienić ten tekst na własny, a przykładowe ilustracje, na ilustracje dotyczące wykonywanego projektu.

Poniżej znajduje się przykładowa ilustracja dokumentu wygenerowanego przez szablon:



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Po stronie tytułowej można wpisać nazwę grupy projektowej i wypisać jej członków wraz z ich rolami w projekcie.

Nazwa Grupy Projektowej

1. Imię Nazwisko 1 – grafik
2. Imię Nazwisko 2 – webdesigner
3. Imię Nazwisko 3 – programista
4. Imię Nazwisko 3 – projektant systemu



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

W rozdziale zatytułowanym “Cel Projektu” powinny zostać opisane cele i założenia projektu. W przypadku przewodnika multimedialnego należy napisać kto będzie użytkownikiem projektu i jakie treści będzie taka osoba mogła znaleźć w tej aplikacji.

Cel Projektu

Celem projektu jest|.



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

W rozdziale „Harmonogram Projektu” umieszczone są terminy realizacji działań projektowych oraz czas dostarczenia i odbioru prac cząstkowych (pojedynczych zadań) w postaci dokładnego harmonogramu.

Przypomnijcie sobie rozdział z podręcznika do gimnazjum zatytułowany „Modyfikacja układu tekstu zawierającego obrazy i tabele”

Nazwisko	Imię	funkcja

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Przyznanie ról członkom grupy projektowej pozwoli na odpowiednie i równomierne rozdzielenie prac nad projektem. W przypadku samodzielnie wykonywanego projektu podpunkt ten można pominąć i skoncentrować się wyłącznie na zdefiniowaniu zadań. Należy zastanowić się nad czasem realizacji projektu, z uwzględnieniem rozkładu zajęć w roku szkolnym, by jak najefektywniej wykorzystać czas lekcji. W dokumentacji można umieścić rozkład roku szkolnego, co pozwoli na optymalne zaplanowanie czasu wykonania poszczególnych zadań.

Wrzesień							Październik							Listopad						
Pn	Wt	Śr	Cz	Pt	Sb	N	Pn	Wt	Śr	Cz	Pt	Sb	N	Pn	Wt	Śr	Cz	Pt	Sb	N
	1	2	3	4	5	6				1	2	3	4							1
7	8	9	10	11	12	13	5	6	7	8	9	10	11	2	3	4	5	6	7	8
14	15	16	17	18	19	20	12	13	14	15	16	17	18	9	10	11	12	13	14	15
21	22	23	24	25	26	27	19	20	21	22	23	24	25	16	17	18	19	20	21	22
28	29	30					26	27	28	29	30	31		23	24	25	26	27	28	29

Grudzień							Styczeń							Luty						
Pn	Wt	Śr	Cz	Pt	Sb	N	Pn	Wt	Śr	Cz	Pt	Sb	N	Pn	Wt	Śr	Cz	Pt	Sb	N
	1	2	3	4	5	6				1	2	3		1	2	3	4	5	6	7
7	8	9	10	11	12	13	4	5	6	7	8	9	10	8	9	10	11	12	13	14
14	15	16	17	18	19	20	11	12	13	14	15	16	17	15	16	17	18	19	20	21
21	22	23	24	25	26	27	18	19	20	21	22	23	24	22	23	24	25	26	27	28
28	29	30	31				25	26	27	28	29			29						

Marzec							Kwiecień							Maj						
Pn	Wt	Śr	Cz	Pt	Sb	N	Pn	Wt	Śr	Cz	Pt	Sb	N	Pn	Wt	Śr	Cz	Pt	Sb	N
	1	2	3	4	5	6					1	2	3							1
7	8	9	10	11	12	13	4	5	6	7	8	9	10	2	3	4	5	6	7	8
14	15	16	17	18	19	20	11	12	13	14	15	16	17	9	10	11	12	13	14	15
21	22	23	24	25	26	27	18	19	20	21	22	23	24	16	17	18	19	20	21	22
28	29	30	31				25	26	27	28	29	30		23	24	25	26	27	28	29

Czerwiec							Lipiec							Sierpień						
Pn	Wt	Śr	Cz	Pt	Sb	N	Pn	Wt	Śr	Cz	Pt	Sb	N	Pn	Wt	Śr	Cz	Pt	Sb	N
			1	2	3	4					1	2	3	1	2	3	4	5	6	7
6	7	8	9	10	11	12	4	5	6	7	8	9	10	8	9	10	11	12	13	14
13	14	15	16	17	18	19	11	12	13	14	15	16	17	15	16	17	18	19	20	21
20	21	22	23	24	25	26	18	19	20	21	22	23	24	22	23	24	25	26	27	28
27	28	29	30				25	26	27	28	29	30	31	29	30	31				

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Następnie należy zdefiniować kolejne zadania, które powinny zostać wykonane w celu prawidłowej realizacji projektu. W przypadku Przewodnika Multimedialnego będą to:

- określenie miast które mają być w przewodniku,
- zebranie fotografii,
- zebranie filmów,
- zebranie opisów miast (teksty),
- zebranie danych statystycznych,
- opracowanie layoutu programu,
- zaprojektowanie menu,
- zaprojektowanie przycisków sterowania,
- zaprojektowanie akcji działania aplikacji,
- zebranie danych statystycznych w bazie danych,
- zaprojektowanie relacyjnej bazy danych,
- oprogramowanie aplikacji,
- projekt strony WWW,
- implementacja strony www,
- przygotowanie prezentacji.

W rozdziale „Organizacja Projektu” należy przydzielić zadania konkretnym członkom grupy i zaplanować je w czasie. Można to wykonać w postaci [wykresu gantta](#).

Przykładowy wykres Gantta:

Tygodnie		1	2	3	4	5	6	7	8	9	10	11	12	13	14
zadanie	2014-10-25	2014-11-01	2014-11-08	2014-11-15	2014-11-22	2014-11-29	2014-12-06	2014-12-13	2014-12-20	2014-12-27	2015-01-03	2015-01-10	2015-01-17	2015-01-24	2015-01-31
Określenie liczby uczestników															
Sporządzenie listy specjalnych gości															
Znalezienie Sali															
Zatrudnienie zespołu muzycznego															
Wysłanie zaproszeń															
Wynajęcie firmy cateringowej															
Udekorowanie Sali															
Opracowanie programu artystycznego															

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

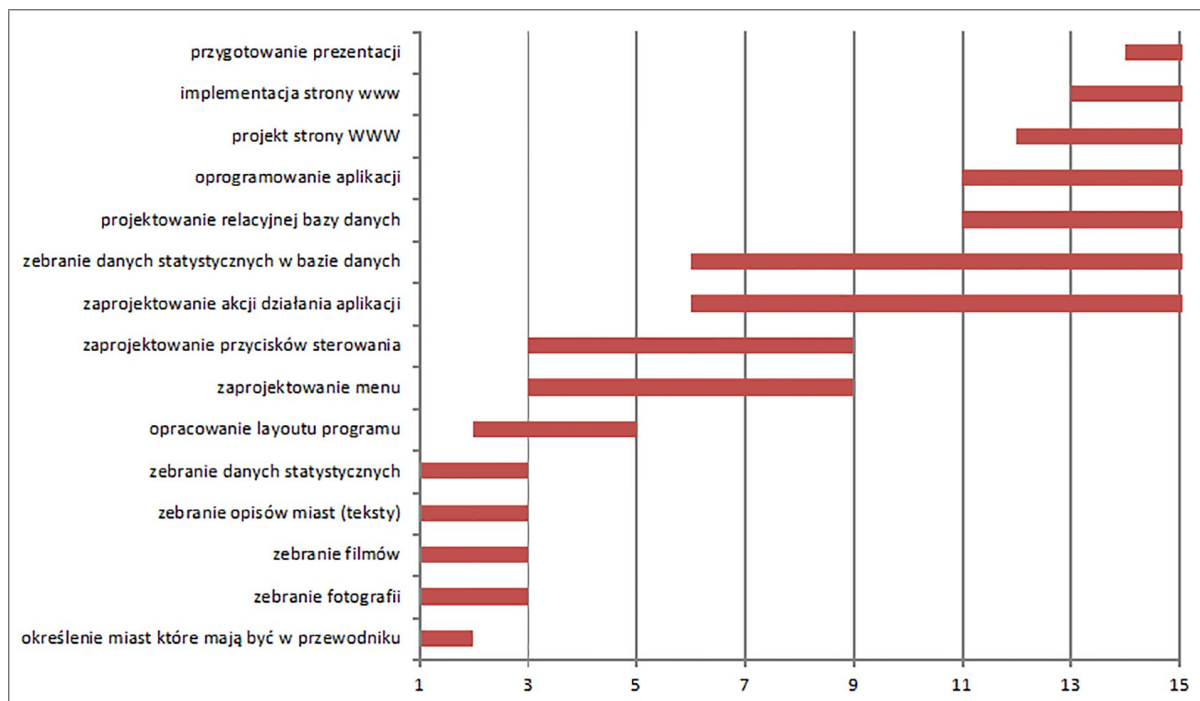
Chociaż nie we wszystkich wersjach arkuszy kalkulacyjnych udostępniono wykresy Gantta, można wykonać symulację takiego wykresu, dostosowując skumulowany wykres słupkowy tak, aby były na nim wyświetlane zadania, czas trwania zadań oraz hierarchia. Najczęściej używaną jednostką czasu na osi poziomej wykresu Gantta w programie Excel są dni. Mogą to również być numery lekcji, które wypadają w kolejnych tygodniach.

W celu stworzenia wykresu Gantta należy wykonać tabelę zawierającą wszystkie zadania związane z realizacją projektu. Dla Przewodnika Multimedialnego będzie ona następująca:

zadanie	początek	czas trwania
określenie miast które mają być w przewodnik	1	1
zebranie fotografii	1	2
zebranie filmów	1	2
zebranie opisów miast (teksty	1	2
zebranie danych statystycznych	1	2
opracowanie layoutu programu	2	3
zaprojektowanie menu	3	6
zaprojektowanie przycisków sterowania	3	6
zaprojektowanie akcji działania aplikacji	6	13
zebranie danych statystycznych w bazie danych	6	10
projektowanie relacyjnej bazy danych	11	13
oprogramowanie aplikacji	11	12
projekt strony WWW	12	13
implementacja strony www	13	14
przygotowanie prezentacji	14	15

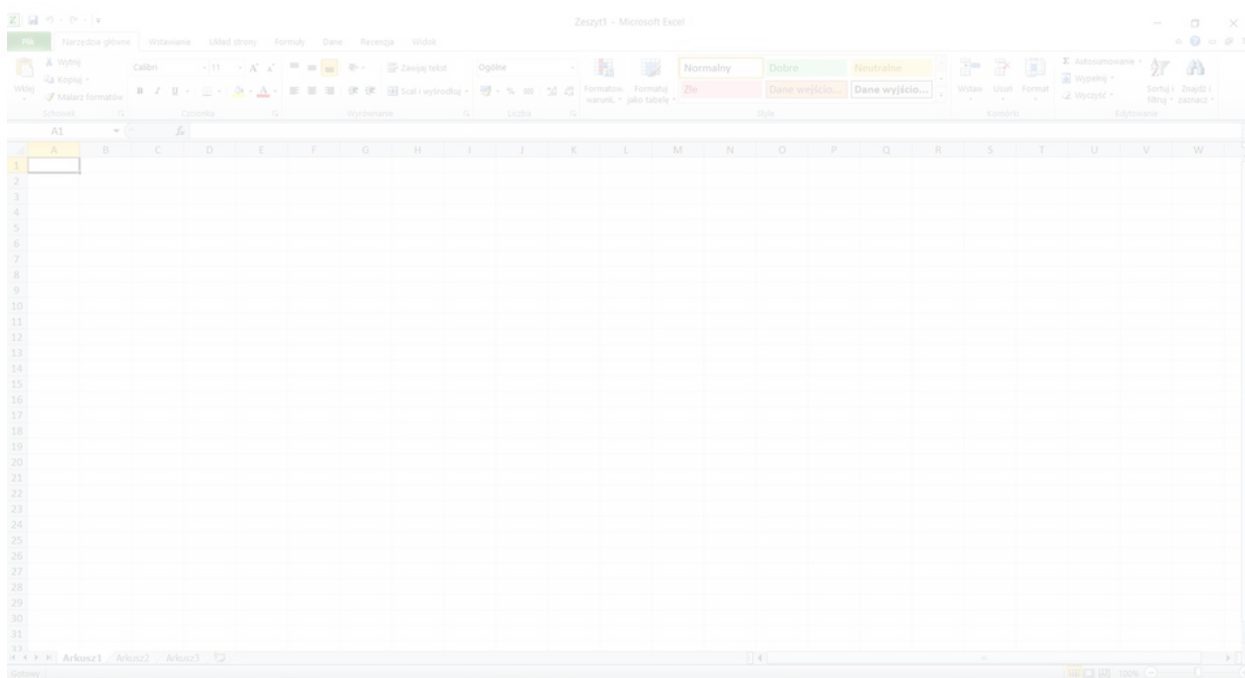
Źródło: Piotr Napieralski <Piotr.napieralski@gmail.com>, licencja: CC BY 3.0.

Przykładowy wykres dla opisywanego zadania będzie wyglądał następująco:



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Wideo tutorial:



Zasób interaktywny dostępny pod adresem <https://zpe.gov.pl/a/DUZA12PPC>

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Tak zaprojektowane dane z arkusza można przenieść do dokumentacji w edytorze tekstu. Można w tym celu skopiować zaznaczone dane lub wykresy w programie arkusza kalkulacyjnego, a następnie użyć poleceń Wklej lub Wklej specjalnie w programie edytor tekstu.

Dalsze działanie polegać będzie na opisie organizacji projektu. W poniższym rozdziale należy opisać zadania, obowiązki i role przypisane członkom zespołu.

Opis ról dla grupy wieloosobowej można przedstawić w tabeli. Bardzo ważny jest jasny podział ról i zadań w trakcie realizacji projektu. Poniższej znajduje się przykład opisu ról w projekcie dla grupy wieloosobowej.

Zadanie	Termin realizacji	Osoby zaangażowane w realizację zadania	Wynik działań
określenie miast które mają być w przewodniku	Pierwsza lekcja	Wszyscy	Nazwy wybranych miast
zebranie fotografii	Na drugą lekcję	Imię Nazwisko 4	Zdjęcia ze stron internetowych
zebranie filmów	Na drugą lekcję	Imię Nazwisko 4	Filmy z Internetu
zebranie opisów miast (teksty)	Na drugą lekcję	Imię Nazwisko 4	Opracowanie opisów z Internetu
zebranie danych statystycznych	Na drugą lekcję	Imię Nazwisko 4	Dane statystyczne od nauczyciela geografii
opracowanie layoutu programu	Na piątą lekcję	Imię Nazwisko 1	Layout programu
zaprojektowanie menu	Na dziewiątą lekcję	Imię Nazwisko 1	Wygląd menu
zaprojektowanie przycisków sterowania	Na dziewiątą lekcję	Imię Nazwisko 1	Przyciski
zaprojektowanie akcji działania aplikacji	Do końca zajęć	Imię Nazwisko 1	Podstawowe działanie aplikacji
zebranie danych statystycznych w bazie danych	Do końca zajęć	Imię Nazwisko 1	Zbiór danych statystycznych
projektowanie relacyjnej bazy danych	Do końca zajęć	Imię Nazwisko 1	Relacyjna baza danych wraz z jej wizualizacją
oprogramowanie aplikacji	Do końca zajęć	Imię Nazwisko 1	Działająca aplikacja
projekt strony WWW	Do końca zajęć	Imię Nazwisko 1 Imię Nazwisko 1	Strona projektu
implementacja strony www	Do końca zajęć	Imię Nazwisko 1	Strona projektu
przygotowanie prezentacji	Do końca zajęć	Imię Nazwisko 4 Imię Nazwisko 1	Prezentacja multimedialna projektu

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

W przypadku zadania realizowanego samodzielnie należy tylko opisać działania, które mają zostać podjęte w celu realizacji projektu i ich harmonogram.

Przedostatni rozdział to “Formuła realizacji” dotycząca sposobu realizacji projektu. Formuła decyduje o tym, które zadania wykonane zostaną samodzielnie, a które polegać będą na wykorzystaniu zewnętrznych materiałów, lub wykonane zostaną przy pomocy nauczyciela. Ostatni rozdział dotyczy dokumentacji projektu i będzie rozwijany wraz z postępem prac nad projektem. Powinien on być technicznym opisem czynności, które należy wykonać tworząc projekt.

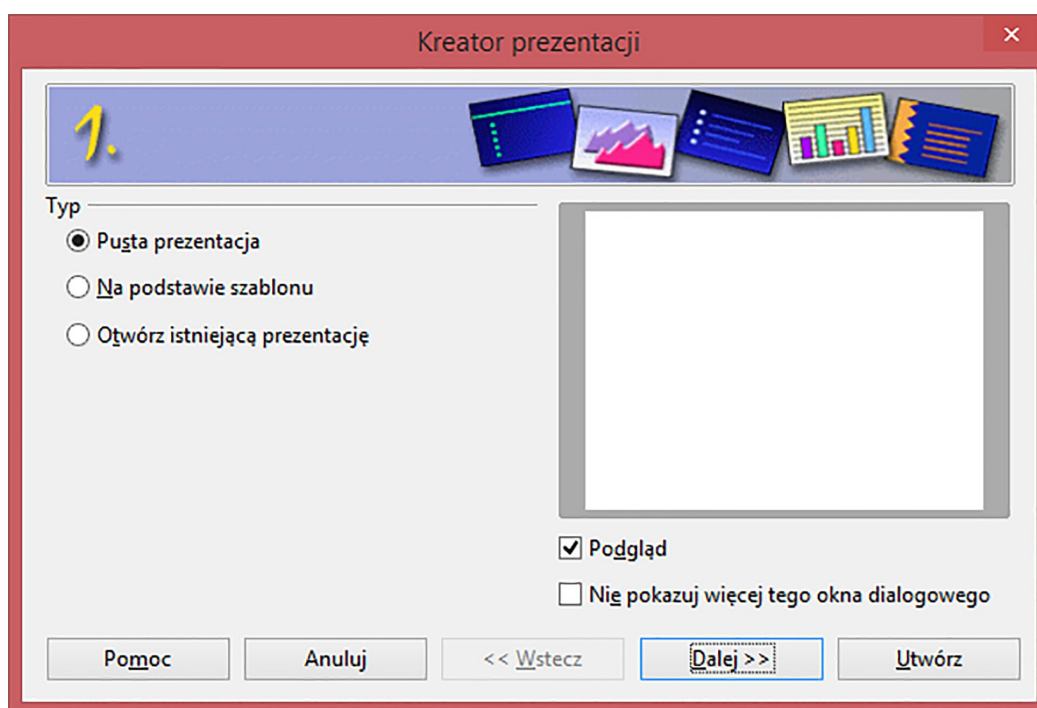
Utworzenie prezentacji do pokazu multimedialnego

Już wiesz

Przypomnijcie sobie rozdział „Tworzenie prezentacji” z podręcznika do szkoły podstawowej

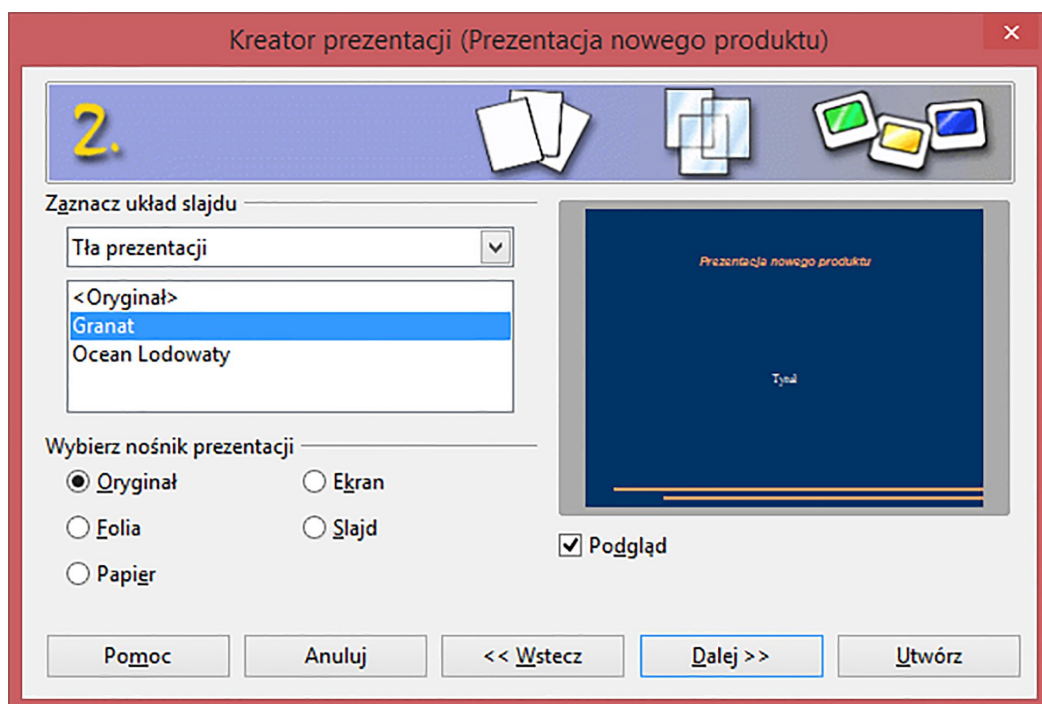
Pokaz multimedialny pozwoli na przedstawienie pomysłu aplikacji w szerszym gronie, a za sprawą konstruktywnej krytyki, w obiektywny sposób ocenić dotychczasową pracę i wprowadzić ewentualne poprawki.

Do stworzenia prezentacji można wykorzystać jeden z dostępnych na rynku programów specjalnie do tego przeznaczonych. Podobnie jak w przypadku edytorów tekstu, w sieci można znaleźć wiele gotowych szablonów.

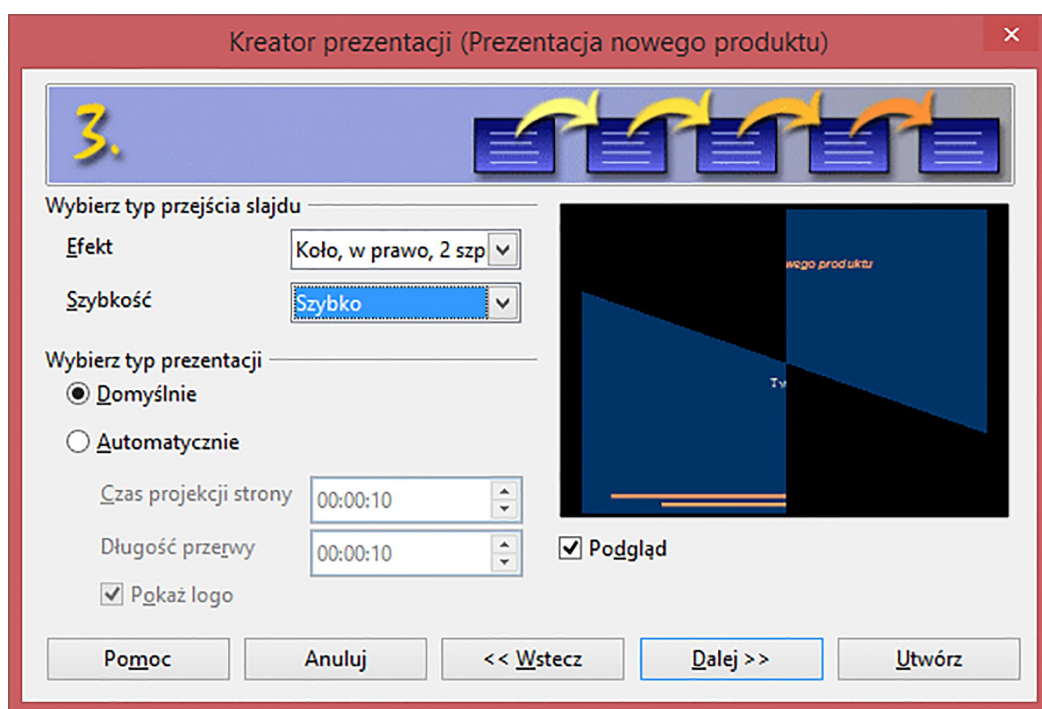


Źródło: Piotr Napieralski, licencja: CC BY 3.0.

W przykładzie wykorzystany zostanie jeden z szablonów. Kolejne etapy pozwolą nam na wybranie tła i stylu dla opracowywanej prezentacji:



Źródło: Piotr Napieralski, licencja: CC BY 3.0.



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Kreator prezentacji (Prezentacja nowego produktu)

4.

Przedstaw główne idee

Podaj swoje nazwisko lub nazwę swojej firmy.

Jan Kowalski

Co jest tematem prezentacji?

Prezentacja Multimedialnego Przewodnika po Polsce

Dodatkowe pomysły do zaprezentowania?

1. Prezentacja członków grupy
2. Prezentacja produktu
3. Podsumowanie

Pomoc Anuluj << Wstecz Dalej >> Utwórz

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Kreator prezentacji (Prezentacja nowego produktu)

5.

Wybierz strony

- ☒ Tytuł
- ☒ Cel długoterminowy
- ☒ Życzenia klientów
- ☒ Spełnienie życzeń klientów
- ☒ Analiza kosztów
- ☒ Walory i zalety
- ☒ Następne kroki

☒ Utwórz podsumowanie

☒ Podgląd

Prezentacja Multimedialnego Przewodnika po Polsce

Jan Kowalski

1. Prezentacja członków grupy
2. Prezentacja produktu
3. Podsumowanie

Pomoc Anuluj << Wstecz Dalej >> Utwórz

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Prezentacja Multimedialnego Przewodnika po Polsce

Jan Kowalski

1. Cel
2. Prezentacja członków grupy
3. Prezentacja produktu
4. Podsumowanie

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Cel

- Stworzenie interaktywnej aplikacji wspomagającej proces nauczania geografii dla szkół ponadpodstawowych

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

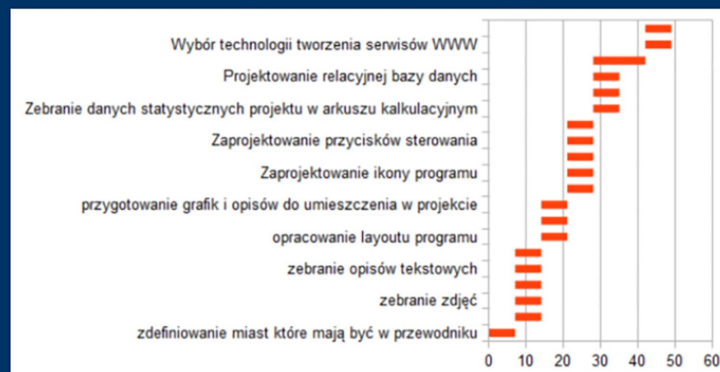
Podobne aplikacje

- Polska AudioPrzewodnik - aplikacja na urządzenie mobilne
- Przewodnik multimedialny Muzeum Wsi Radomskiej
- Multimedialny przewodnik Łódzkiej Biblioteki Publicznej im. Marszałka Józefa Piłsudskiego w Łodzi



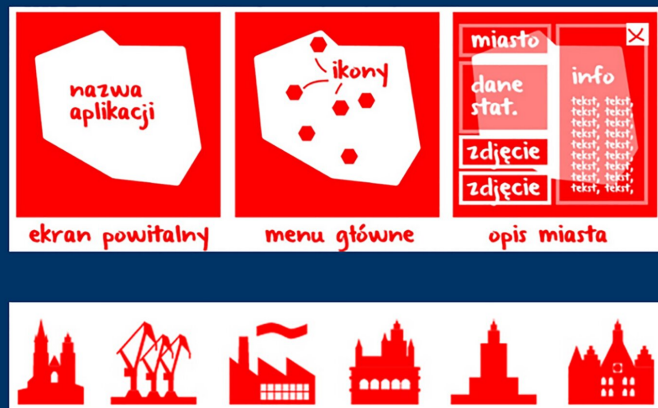
Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Zadania do wykonania



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Pomysł na wygląd aplikacji



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Prezentacja materiałów do prezentacji

- Przedstawiamy pokaz slajdów zdjęć które będą użyte w projektowanej aplikacji



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

- Tutaj przedstawiamy dotychczasowy etap prac



Umieszczenie dokumentów w chmurze

Przypomnijcie sobie rozdział „Praca w chmurze cyfrowej” z podręcznika do gimnazjum.

Należy utworzyć konto dla opracowywanego projektu na wirtualnym dysku. W przykładzie posłużono się technologią firmy Microsoft OneDrive. Aplikacja OneDrive jest wbudowana w systemy Windows 8.1 i Windows RT 8.1. W systemach Windows 7 lub Windows Vista, należy zainstalować klasyczną aplikację OneDrive dla systemu Windows. Na komputerze zostanie wtedy utworzony folder OneDrive, który będzie synchronizowany z usługą OneDrive. W przypadku komputera z systemem Mac należy zainstalować OneDrive na komputerze i automatycznie synchronizować pliki z folderem OneDrive na wszystkich urządzeniach.

W celu instalacji Onedrive dla Linux w konsoli należy wykonać następujące komendy

```
$ git clone
```

```
$ cd onedrive-d
```

```
$ ./inst install
```

Po instalacji, należy skonfigurować usługę tak by program *onedrive-d* uzyskał dostęp do istniejącego konta *OneDrive*.

Najpierw, należy utworzyć folder lokalny który będzie używany do synchronizacji ze zdalnym kontem *OneDrive*.

W tym celu należy wpisać komendę:

```
$ mkdir ~/onedrive
```

Wpisanie tej komendy rozpocznie konfigurację początkową.

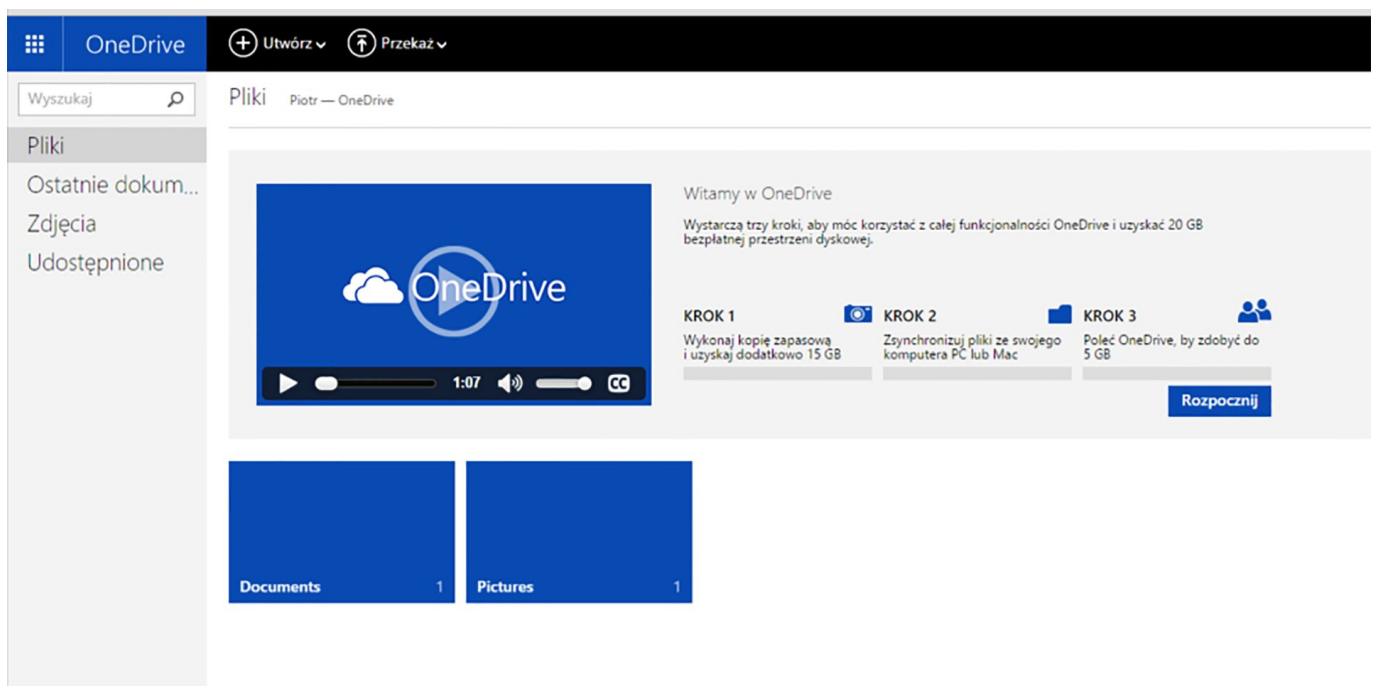
W polu *Location*, należy podać ścieżkę do wcześniej stworzonego folderu. W polu *Authentication*, pojawi się informacja *You have not authenticated OneDrive-d yet*.

Teraz należy zaznaczyć opcję *Connect to OneDrive.com*.

Po zaznaczeniu tych opcji pokaże się okno pozwalające na skonfigurowanie konta *OneDrive.com*

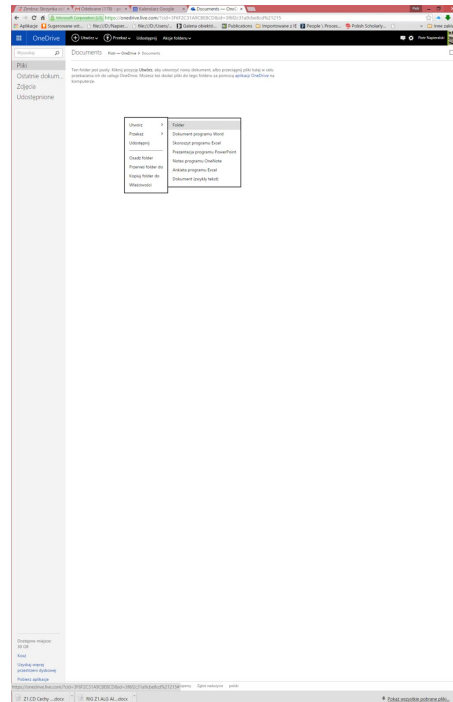
Po wpisaniu opcji konfiguracyjnych konta (podobnie jak w pozostałych systemach), w oknie ustawień, status zostanie zmieniony na *You have connected to OneDrive.com*. Wybranie przycisku „OK” kończy konfigurację.

Następnie należy udostępnić utworzony folder wszystkim członkom zespołu projektowego oraz nauczycielowi i umieścić w nim utworzoną wcześniej dokumentację projektu oraz prezentację. Oprócz stworzonej dokumentacji projektowej, na założonym wirtualnym dysku w chmurze można również w przyszłości przechowywać wszystkie inne pliki i dane realizowanego projektu. Należy wybrać jedną z dostępnych na rynku metod wymiany dokumentów w chmurze.

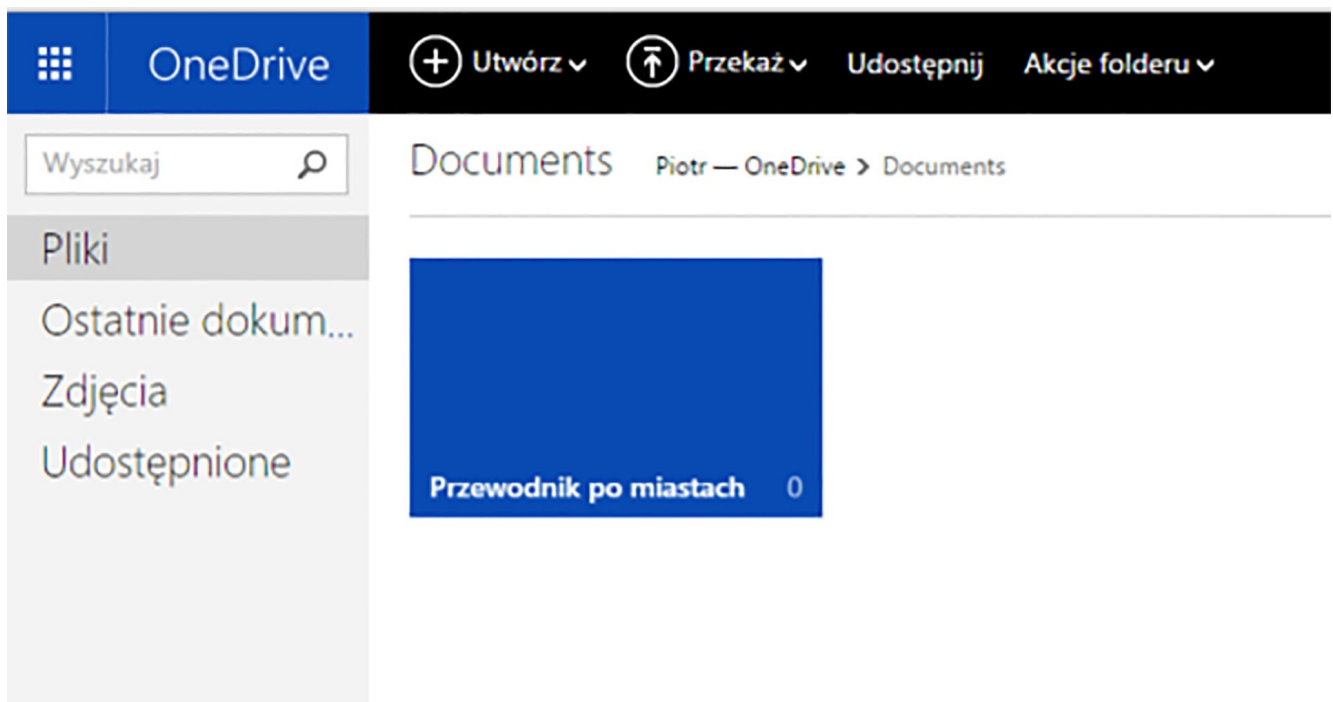


Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Utwórzmy folder zawierający dane projektu:

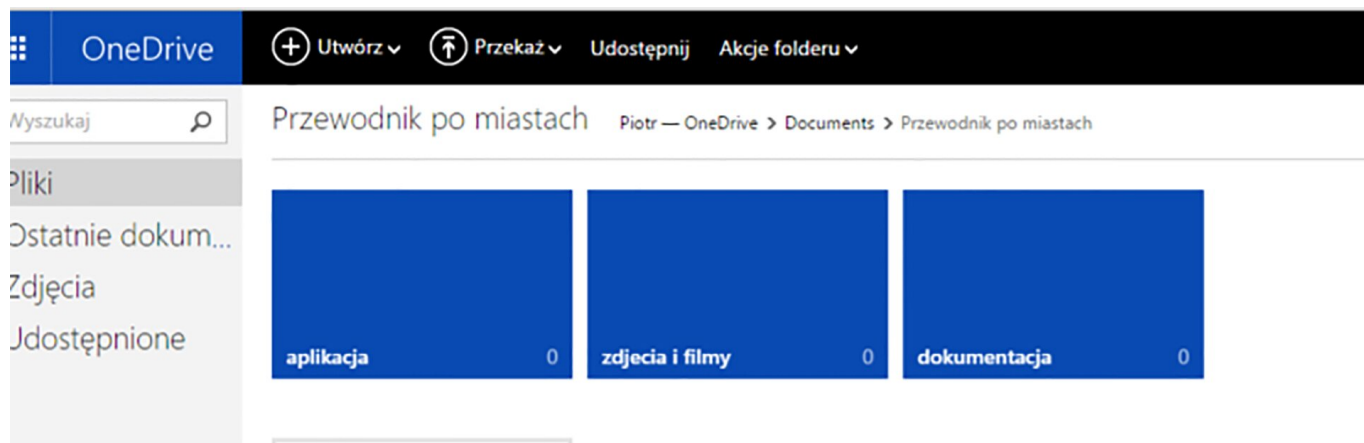


Źródło: Piotr Napieralski, licencja: CC BY 3.0.



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

W folderze można przechowywać wszystkie pliki związane z projektem. Za pomocą odpowiedniego polecenia należy importować do stworzonego folderu powstałe wcześniej pliki. Teraz każdy z uczestników w projekcie będzie mógł równocześnie edytować dokumenty.



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Wygląd aplikacji (Grafika)

Każda aplikacja powinna mieć przemyślany i funkcjonalny interfejs graficzny (ikony, przyciski, tło). Oprócz elementów interfejsu, w zależności od przeznaczenia aplikacji, potrzebne mogą być dodatkowe ilustracje, fotografie, itp. Przykładowo, realizując przewodnik geograficzny, konieczne będzie przygotowanie map i fotografii. Jeśli tworzymy grę komputerową będziemy musieli wykonać obrazki planszy, bohaterów. W przypadku symulatora lotów kosmicznych potrzebne będą ilustracje rakiety i planet. Zadanie te można zrealizować dowolnym programem do grafiki rastrowej. Najpopularniejszym darmowym programem do obróbki grafiki, jest program GIMP który można pobrać z Internetu. Aby zrealizować ten etap projektu należy przypomnieć sobie informacje z gimnazjum dotyczące:

- tworzenie i modyfikowanie grafiki rastrowej, tworzenie fotomontaży (praca na warstwach), wykonywanie retuszu, modyfikowanie kolorystyki,
- tworzenie prostych obrazów grafiki wektorowej,
- łączenie grafiki rastrowej i wektorowej (wstawianie napisów do grafiki rastrowej),
- formaty zapisów plików graficznych,
- modele kolorów.

O interfejsach słów kilka...

Na podstawie przygotowanej wcześniej dokumentacji projektowej aplikacji należy zdecydować jak będzie wyglądał jej **interfejs**.

Przemyślany interfejs powinien uwzględniać kontekst i sposób wykorzystania aplikacji przez jej użytkowników. W przypadku realizowanego Przewodnika po największych miastach w Polsce, graficzny interfejs powinien być oparty na symbolicznym odwołaniu do terytorium naszego kraju oraz barw narodowych.

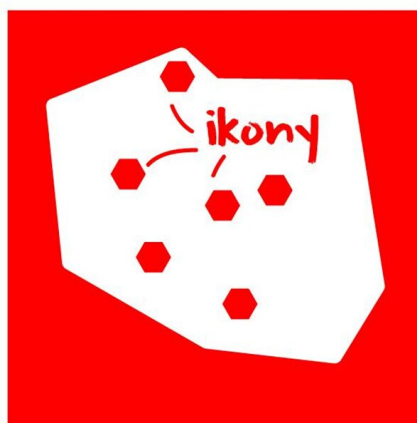
Na podstawie listy zadań aplikacji należy zastanowić się jakie elementy graficzne będą potrzebne do jej realizacji. Oprócz grafiki wyświetlanej na ekranie powitalnym aplikacji, będą potrzebne również ikony służące jako przyciski, dzięki którym będzie można wyświetlać informacje o poszczególnych miastach naszego przewodnika - zarówno informacje tekstowe jak i zdjęcia prezentujące wybrane miejsca i zabytki.

Przed przystąpieniem do wykonania poszczególnych elementów graficznych, należy zaplanować ich wymiary, rozmieszczenie a także możliwość wywołania lub ukrycia. Innymi słowy, należy stworzyć projekt układu aplikacji (ang. Layout). Okno programu Przewodnika po najważniejszych miastach w Polsce będzie wyświetlało trzy główne ekrany:

- ekran powitalny - prezentujący nazwę aplikacji,
- ekran zawierający menu główne - pozwalające wybrać konkretne miasto,
- ekran opisujący dane miasto - wyświetlający informacje tekstowe, dane statystyczne oraz zdjęcia.



ekran powitalny



menu główne



opis miasta

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Przed wykonaniem tych elementów należy przypomnieć sobie z podręcznika do gimnazjum jak używa się programów do projektowania grafiki wektorowej i rastrowej:

Projektowanie elementów interfejsu aplikacji

Dysponując listą potrzebnych elementów graficznego interfejsu aplikacji można przystąpić do ich wykonania za pomocą odpowiednich narzędzi do tworzenia i edycji grafiki wektorowej

Ekran powitalny

Już wiesz

Z podręcznika do gimnazjum przypomnijcie sobie rozdziały:

”Rozdzielczość obrazu drukowanego i wyświetlanego na monitorze”,

”Zaawansowana obróbka grafiki cyfrowej z wykorzystaniem warstw”.

Pracę należy rozpocząć od wykonania ekranu powitalnego aplikacji, który w przypadku projektowanego Przewodnika po największych miastach będzie przedstawiał zarys Polski, wewnątrz którego należy umieścić nazwę aplikacji:



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

- Pierwszym krokiem po uruchomieniu programu do tworzenia grafiki wektorowej będzie utworzenie nowego dokumentu o rozmiarach planowanego okna aplikacji czyli 700 na 700 pikseli, w rozdzielczości przeznaczonej do wyświetlania na monitorze czyli 72 dpi.
- Następnie trzeba utworzyć nową warstwę, na której należy umieścić obraz przedstawiający mapę Polski. Warstwie tej trzeba nadać delikatną przezroczystość.

- Na nowej warstwie należy wykonać obrys granic Polski za pomocą narzędzia do tworzenia i edycji krzywych. Narzędzie to działa i wygląda podobnie w każdym programie do grafiki wektorowej.

Ćwiczenie 1

Proszę utworzyć ekran powitalny aplikacji projektowej

Wskazówka

Przypomnijcie sobie z podręcznika do gimnazjum „Zaawansowana obróbka grafiki cyfrowej z wykorzystaniem warstw” Narzędzie ścieżek może mieć wiele zastosowań (tworzenie obrysów, czy skomplikowanych linii, a także przydaje się w tworzeniu selekcji i szparowaniu).

Przypomnijcie sobie z podręcznika do gimnazjum „Rozdzielczość obrazu drukowanego i wyświetlanego na monitorze”

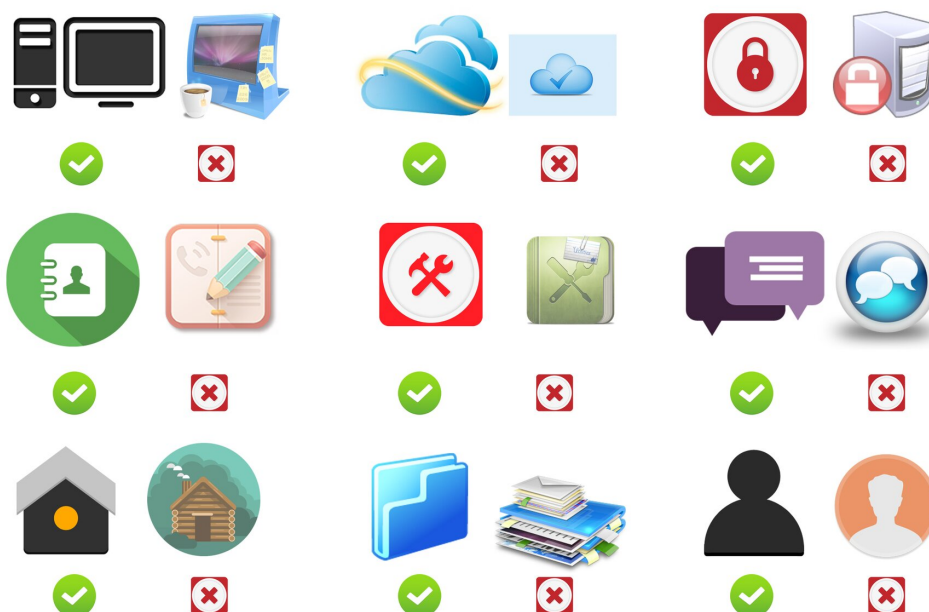
Ikony aplikacji

Zgodnie z zaplanowanym układem przewodnika, główne okno aplikacji będzie zawierało sześć ikon symbolizujących poszczególne miasta i pełniących funkcję przycisków umożliwiających wyświetlenie informacji i zdjęć na temat wybranego miasta.

Ważne!

Tworząc ikonę trzeba pamiętać o kilku zasadach, które mają na celu zapewnienie czytelności znaku:

- **kontrast** pomiędzy symbolem/tekstem a tłem ikony,
- rozmiar ikony, aby wygodnie z niej skorzystać,
- mało skomplikowany symbol ikony, aby pozostał czytelny.



Źródło: Ewa Dobrzyńska, licencja: CC BY 3.0.

Ćwiczenie 2

Utwórz ikonę dla każdego z miast projektowanego przewodnika

Wskazówka

Większość programów do obróbki grafiki wektorowej i rastrowej operuje na ścieżkach. Dzięki odpowiednim zabiegom na ścieżkach i obiektach można uzyskać dowolne kształty. Zarówno w Illustrator, Corel jak i Inkscape posiada opcje edycji ścieżek. Znajdź narzędzia do ścieżek programie, z którego korzystasz. Możesz sprawdzić działanie ścieżek w dokumentacji programu którego używasz (plik pomocy lub strona internetowa).

Edycja obrazów rastrowych na potrzeby projektu

Już wiesz

Przypomnijcie sobie z podręcznika do gimnazjum rozdziały:

"Rozdzielczość obrazu drukowanego i wyświetlanego na monitorze",

"Bezpieczne pobieranie informacji oraz plików ze źródeł internetowych".

Następnym etapem prac nad wizualną oprawą aplikacji będzie przygotowanie jej zawartości. Każda z proponowanych aplikacji zawiera inne charakterystyczne elementy graficzne. W przypadku przewodnika, należy przygotować zdjęcia czy filmy, ilustrujące odpowiednie działy. Gra, niezależnie od jej gatunku, będzie korzystać z grafiki składającej się na scenografię, czyli z tak zwanych assetów (ASSET jest to plik graficzny, filmowy, dźwiękowy, który jest używany w grze.)

Symulator lotów, podobnie jak gra komputerowa, potrzebować będzie elementów scenerii, plansz, czy też projektu statku kosmicznego.

Obrazy cyfrowe potrzebne do wykonania powyższych elementów aplikacji możemy pozyskać w różny sposób, np. importując wykonane zdjęcia z aparatu cyfrowego lub skanując znalezione ilustracje za pomocą skanera komputerowego.

Alternatywnym źródłem obrazów cyfrowych, które można wykorzystać w projekcie są internetowe banki zdjęć.

(np. www.shutterstock.com, www.istockphoto.com)

Tego typu serwisy internetowe udostępniają zdjęcia i filmy wideo z licencją do wykorzystania ich za niewielką opłatą lub za darmo do użytku własnego. Zbiory te zapisane są w internetowej bazie danych, która pozwala użytkownikowi na szybkie odnalezienie odpowiedniej ilustracji na podstawie właściwych kryteriów. Wybrany zasób może zostać zdalnie pobierany na dysk użytkownika. Serwisy te dają możliwość uiszczenia opłaty licencyjnej, która zakłada komercyjne wykorzystanie obrazów zarówno w publikacjach drukowanych jak i elektronicznych. Nowocześni dystrybutorzy stock photography zazwyczaj oferują nie tylko zdjęcia, ale również filmy i ilustracje.

Po skompletowaniu wszystkich potrzebnych obrazków można przystąpić do ich obróbki, lub w ramach potrzeb, łączenia ich w finalne ilustracje. Przykładowo, z wybranych zdjęć części i mechanizmów można stworzyć rakietę kosmiczną, która będzie sterowana z poziomu odpowiedniej aplikacji.

Zdjęcia zgromadzone na potrzeby „Przewodnika po miastach w Polsce” mogą wymagać odpowiedniej korekty w postaci:

- **poprawienia nasycenia barw**,
- **retuszu**, polegającego na wymazaniu drobnych elementów z kadru,
- **wykadrowania**, czyli przycięcia wykorzystywanego zdjęcia.

Na koniec należy zapisać każde z przygotowanych zdjęć w formacie JPG, w postaci miniaturki o rozmiarach 280 na 186 pikseli, aby pasowało do szablonu tworzonej aplikacji, oraz rozdzielczości przeznaczonej do wyświetlania na monitorze, czyli 72 dpi.

Należy pamiętać, że obrazy mogą być zapisywane w różnych formatach z użyciem różnych rodzajów kompresji

Przygotowanie treści przewodnika

Treść do przewodnika można przygotować na podstawie informacji zawartych na stronach internetowych. Korzystając z nich należy jednak pamiętać, by [powołać się na źródło informacji](#), które zostały wykorzystane. W przypadku korzystania z informacji dostępnych np. w Wikipedia, wystarczy powołać się na źródło (). Jeśli korzysta się ze strony dla której prawa autorskie nie zostały zastrzeżone, trzeba zapytać się właściciela treści o zgodę na wykorzystanie jego utworu.

W przykładowym opisie miast, tekst został zapożyczony ze strony Wikipedia.

Na mapach: 51°46′36″N 19°27′17″E (mapa)

Łódź ^[edytuj]

Ten artykuł dotyczy miasta Łódź. Zobacz też: inne znaczenia tej nazwy.

Łódź (**•** **•** **•** **Łódź⁻¹**) – miasto na prawach powiatu w środkowej Polsce. Siedziba władz województwa łódzkiego, powiatu łódzkiego wschodniego oraz gminy Nowosolna. Przejściowa siedziba władz państwowych w 1945 roku^[5].

Ośrodek akademicki (6 uczelni państwowych oraz 22 prywatne), kulturalny i przemysłowy. Przed **przemianami polityczno-gospodarczymi w 1989 r.** centrum przemysłu włókienniczego i filmowego. Ważny węzeł komunikacji drogowej. Szesć kilometrów od centrum Łodzi znajduje się Międzynarodowy Port Lotniczy **Łódź im. Władysława Reymonta**.

Łódź jest trzecim miastem w Polsce pod względem liczby ludności zameldowanej (706 004) i czwartym pod względem powierzchni (293,25 km²)^{[2][1]}.

W 2015 r., niektóre obszary miasta zostały uznane za „Pomnik historii” pod wspólną nazwą „Łódź – wielokulturowy krajobraz miasta przemysłowego”^[6].

Miasto należy do Unii Metropolii Polskich^[7] i Związku Miast Polskich^[8], a także do stowarzyszenia Eurocities^[9].

Łódź była miastem duchowym^[10].

Spis treści [ukryj]
1 Położenie
2 Środowisko naturalne
3 Klimat
4 Nazwa miasta
5 Historia
5.1 Łódź rolnicza
5.2 Łódź przemysłowa
5.3 Łódź wielkoprzemysłowa
5.4 Łódź podczas II wojny światowej
5.5 Łódź powojenna

Łódź

Płac Wolności,
Centrum Manufaktura, Politechnika Łódka,
Ulica Piotrkowska, Pałac Poznańskiego,
Łódski Manhattan, CKD,
Akademia Sztuk Pięknych,
Biała Fabryka

Herb Łodzi Flaga Łodzi

Dewiz: Ex navicula novi
(Z łódzkiej łódz)

Państwo Polska

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

W przewodniku zostaną wykorzystane materiały filmowe do piosenki Pharrella Williamsa „Happy”. Radosny przebój zainspirował ludzi na całym świecie do tworzenia własnych wersji piosenki. Teledyski powstały m.in. w Paryżu, Amsterdamie i Krakowie. Dla każdego z miast wybranych do przewodnika można znaleźć odpowiednią wersję tego teledysku. Aby pobrać filmy, skopiuj poniższe linki do paska przeglądarki:

<https://www.youtube.com/watch?v=L8yLIGmUiHE>

<https://www.youtube.com/watch?v=KsPLYpCNcTM>

<https://www.youtube.com/watch?v=jDz1ZZptIso>

<https://www.youtube.com/watch?v=B8JaqpHdBuI>

<https://www.youtube.com/watch?v=NUaJA9myEns>

<https://www.youtube.com/watch?v=jwMczNEOpvs>

Programowanie aplikacji

Wstęp

Po przygotowaniu oprawy graficznej oraz wszystkich materiałów potrzebnych do realizacji projektu, można przystąpić do wykonania aplikacji w wybranym języku programowania. Na potrzeby przykładowego Przewodnika po największych miastach w Polsce zostanie wykorzystany język [Processing](#).

Pierwszym krokiem powinno być pobranie i zainstalowanie zintegrowanego środowiska programistycznego. (IDE - ang. [Integrated Development Environment](#))

Processing rozwijany jest jako projekt Open Source, natomiast Processing IDE jest rozpowszechniane na licencji GPL. Biblioteki oraz kod dostarczony przez producentów objęte są licencją LGPL, co pozwala autorom na rozpowszechnianie swoich prac pod dowolną licencją, bez ograniczeń. Wszystkie wersje środowiska Processing znajdują się na stronie <http://processing.org>.

Instalacja środowiska

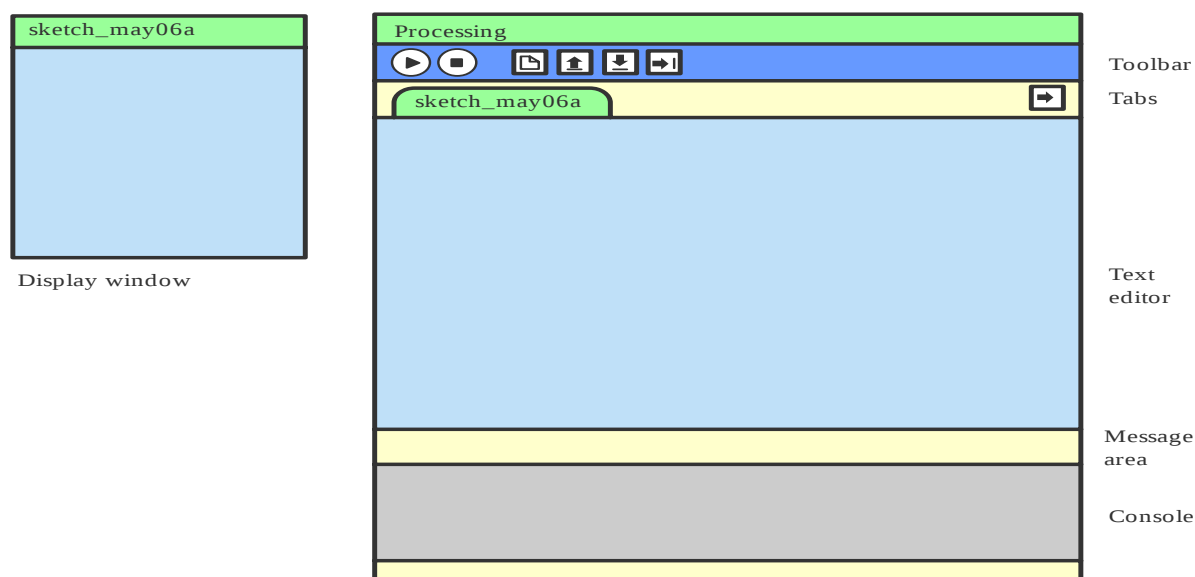
Dla każdego systemu operacyjnego wystarczy pobrać i uruchomić środowisko programistyczne wybierając odpowiednią zakładkę na stronie oprogramowania processing (<https://processing.org/download/?processing>).

Na stronie tej można także znaleźć wszelkie informacje dotyczące tego języka, takie jak dokumentacja, przykłady i tutoriale, oraz dodatkowe biblioteki. Jeśli na komputerze zainstalowane jest środowisko uruchomieniowe Java (JRE) oraz JDK, to można wybrać plik instalacyjny „Windows (Without Java)”. Następnie należy wypakować pobrane pliki do wybranego folderu. Kończy to proces instalacji. Warto jednak przed przystąpieniem do pisania pierwszego programu dokonać jeszcze konfiguracji środowiska.

Programowanie w Processing

Twórcy środowiska Processing zalecają, by swoje projekty przechowywać wewnątrz folderu nazwanego przez nich szkicownikiem (ang. sketchbook). Dzięki temu projekty będą dostępne w łatwy sposób z menu File -> Sketchbook. Folder ten jest ważny jednak przede

wszystkim z innego powodu. To wewnątrz niego musi się znajdować folder z zewnętrznymi bibliotekami, z których chcemy korzystać (instalacja zewnętrznych bibliotek omówiona jest poniżej). Można jednak samodzielnie wybrać inny folder, korzystając z opcji z menu File->Preferences. Pierwszą rzeczą, jaką należy ustawić w oknie Preferences, jest położenie wybranego folderu szkicownika. Powinien on być wybrany tak, aby był do niego łatwy dostęp przy instalowaniu dodatkowych bibliotek. Drugą ważną opcją jest maksymalna dostępna pamięć. W sytuacji, gdy operujemy na dużej ilości plików (zwłaszcza graficznych, wideo czy audio) możemy mieć do czynienia z błędem `java.lang.OutOfMemoryError`, którego można uniknąć poprzez zwiększenie maksymalnej pamięci dla środowiska Processing (co jest analogiczne do zwiększania ilości pamięci dla maszyny wirtualnej Javy, bo przecież Processing powstał jako opakowanie właśnie na Javę, a obecnie rozwinął się do obsługi Android SDK oraz JavaScript). Pozostałe opcje nie są już tak istotne i mogą być dowolnie konfigurowane.



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Przykłady wykorzystania języka processing

Prostota składni języka Processing, pozwoliła wielu osobom, które nie miały doświadczenia programistycznego w realizacji projektów bez konieczności zatrudniania specjalistów z branży IT. To czysto akademickie rozwiązanie umożliwiło wielu badaczom opracować prototypy i sprawdzić ich działanie w środowisku o bardzo dużych możliwościach

graficznych i interakcyjnych bez ponoszenia znacznych nakładów finansowych.

W Internecie można znaleźć dużo przykładów zastosowania tej technologii przez różne grupy zawodowe. Wystarczy wspomnieć o projekcie, który wzbudził spore zainteresowanie świata sztuki. Artyści Daniel Franke i Cedric Kiefer opracowali pomysł wirtualnej ruchomej rzeźby z piasku. Rzeźba powstała na podstawie ruchu tancerki, czytanego przez kontroler Kinect. Animacja została zrealizowana na podstawie analizy w czasie rzeczywistym właśnie w Processing.



Źródło: Daniel Frank, Cedric Kiefer, licencja: CC BY 3.0.

Naukowcy ze znanego niemieckiego instytutu Max Planck, zwizualizowali za pomocą Processing dane około 94,000 publikacji z ostatnich dziesięciu lat zgromadzonych w portalu „SciVerse Scopus”. Zaprojektowana i zaimplementowana dynamiczna sieć pozwoliła na zwizualizowanie połączeń pomiędzy instytucjami naukowymi.

Sieć ta pozwoliła na stworzenie interaktywnego środowiska prezentującego aktywność naukową i współpracę instytutu z innymi jednostkami na całym świecie.



Źródło: Max Planck Science Gallery, licencja: CC BY 3.0.

Stephan Thiel z uniwersytetu w Adelajdzie zrealizował projekt polegający na nowoczesnym sposobie interpretacji dzieł Shakespeare. Użycie Processing pozwoliło na zliczenie słów i znalezienie istniejących relacji i powiązań pomiędzy dziełami autora.



The Comedy of Errors
3576 views on flickr



The Taming of the Shrew
886 views on flickr



The Two Gentlemen of Verona
914 views on flickr



Love's Labour's Lost
554 views on flickr



A Midsummer Night's Dream
1967 views on flickr

Źródło: Stephana Thiela, licencja: CC BY 3.0.

Przedstawione przykłady miały za zadanie zilustrować szerokie zastosowanie technologii którą poznacie podczas najbliższych lekcji. Zdobyta wiedza pozwoli Wam zrealizować ciekawe a przede wszystkim kreatywne pomysły.

Typy danych

Zmienna jest to miejsce w pamięci, w którym możemy przechowywać dane. W zależności od rodzaju danych zmienna zajmować będzie odpowiedni zakres pamięci. Do najczęstszych typów danych możemy zaliczyć liczby całkowite, liczby rzeczywiste, znak, oraz zmienne logiczne.

typ	ilość bajtów	zakres wartości	przykład	opis
int	4	od -2 ³¹ do 2 ³¹ -1	a = 1;	Liczby całkowite
float	4	3.4E +/- 38 (7 cyfr)	a = 0.1;	Liczby zmiennoprzecinkowe
byte	1	od 0 do 255	a = -1;	Liczby jedno bajtowe
boolen	1	false lub true	tak = true;	Typ logiczny zmienna może przyjmować wartość prawda lub fałsz (true lub false)
color	4	Od #000000 do #FFFFFF lub od 0 do 255 dla każdej składowej koloru	n = color(255,0,0) lub n = #FF0000;	Służy do przechowywania wartości koloru w postaci 3 wartości składowych RGB
char	1	od -128 do 127	znak = 'a';	Typ znakowy, znak musi być ujęty w cudzysłów
string			tekst = „Ala”;	Typ tekstowy, zmienna jest łańcuchem znaków, przy deklaracji tekst musi być ujęty w cudzysłów

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Można określić początkową wartość zmiennej już przy jej deklaracji, można też zrobić to później, za pomocą znanego z matematyki operatora przypisania::

nazwa_zmiennej = nowa_wartość_dla_zmiennej;

Operator = służy do przypisywania nowej wartości zmiennej. Wartość jaką przechowuje zmienna, możesz zmieniać wielokrotnie w trakcie działania aplikacji.

Kolejną rzeczą, jest nazewnictwo zmiennych. Nazwy zmiennych nie mogą zawierać polskich znaków i nie mogą zaczynać się od liczby. Wartość zmiennej można zmieniać wielokrotnie w czasie trwania programu.

Paradygmaty programowania i instrukcje sterujące

Paradygmat programowania o sposób przepływu sterowania i wykonywania programu komputerowego. Do najpopularniejszych technik programowania należy programowanie obiektowe i programowanie funkcyjne. W programowaniu obiektowym paradygmat polega na traktowaniu programu jako zbioru współpracujących ze sobą obiektów.

W programowaniu funkcyjnym określamy funkcje, których wykonanie ma prowadzić do wykonania zadania.

Dijkstra proponował użycie tylko trzech rodzajów struktur sterujących:

- Sekwencja (lub konkatencja) — czyli po prostu wykonanie instrukcji w określonej kolejności. W wielu językach rolę „operatora konkatencji instrukcji” spełnia niepozorny średnik...

- Wybór – czyli wykonanie jednej z kilku instrukcji zależnie od stanu programu. Przykładem jest `if-then-else` i `switch/case`.
- Iteracja, czyli powtarzanie instrukcji tak długo, jak długo spełniony (lub niespełniony) jest dany warunek. Chodzi oczywiście o pętle, np. `while`, `repeat-until`, `for` itp.

Processing dostarcza również standardowe możliwości, które dostępne są w każdym języku programowania takie jak :

- komentarze: `//`, `/* ... */`

To, co następuje po dwóch ukośnikach to tzw. komentarz. Processing tego nie czyta, bo są to tylko informacje dla programistów. Komentarz taki obejmuje wszystko, co znajduje się za znakiem podwójnego ukośnika do końca linii, lub między znakami `/*...*/`

- pętle: *for*, *while*,

Instrukcja `for` będzie powtarzać ciąg instrukcji tak długo, jak długo zdefiniowane warunki będą prawdziwe.

```
for (int i = 40; i < 80; i = i+5) {
  line(30, i, 80, i);
}
```

- instrukcje warunkowe: *if*, *if...else*,

Podstawowym rodzajem instrukcji warunkowej spotykanej w większości języków programowania jest *if-then*. Instrukcja ta pozwala na wykonanie określonego bloku kodu w przypadku spełnienia określonego warunku. Jeśli warunek nie jest spełniony zostanie wykonany blok alternatywny. Pomiędzy różnymi językami programowania występują w tej instrukcji nieznaczne różnice składniowe. W Processing instrukcja ta wygląda następująco:

```
if(warunek)
{
  ciąg instrukcji wykonywany jeśli warunek jest spełniony
}
else
{
  ciąg instrukcji wykonywany jeśli warunek nie jest spełniony
}
```

- instrukcję wyboru: *switch*,

Podobnie jak *if-then-else*, jest to instrukcja warunkowa, tzw instrukcja wielokrotnego wyboru. W przypadku niej możemy podejmować decyzje na podstawie wartości jednej zmiennej. Najczęściej wykorzystywana jest ona do obsługi zdarzeń (np. sterowanie za pomocą klawiatury). W zależności od wartości parametru uruchamiana jest odpowiednia akcja.

- operatory

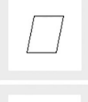
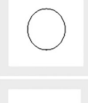
W poniższej tabeli znajdują się najważniejsze operatory pozwalające wykonywać obliczenia, oraz zmiany stanu zmiennych. Operatory te pozwalają też porównywać zmienne ze sobą.

Operator	Opis	Przykłady
Równy (==)	Zwraca true jeżeli operandy są równe.	Otrzymamy prawdę w przypadku gdy: 4 == zmien1 "4" == zmien1 4 == '3'
Różny (!=)	Zwraca true jeżeli operandy są różne.	Otrzymamy prawdę w przypadku gdy: zmien1 != 1 zmien1 != "1"
Identyczny (===)	Zwraca true jeżeli operandy są równe i tego samego typu.	Otrzymamy prawdę w przypadku gdy: 2 === zmien1
Nieidentyczny (!==)	Zwraca true jeżeli operandy nie są równe i/lub nie są tego samego typu.	Otrzymamy prawdę w przypadku gdy: zmien1 !== 5 5 !== '5'
Większy niż (>)	Zwraca true jeżeli lewy operand jest większy od prawego operandu.	Otrzymamy prawdę w przypadku gdy: zmien2 > zmien1
Większy - równy (>=)	Zwraca true jeżeli lewy operand jest większy lub równy prawemu operandowi.	Otrzymamy prawdę w przypadku gdy: zmien2 >= zmien1 zmien1 >= 3
Mniejszy niż (<)	Zwraca true, jeśli lewy operand jest mniejszy, niż prawy.	Otrzymamy prawdę w przypadku gdy: zmien1 < zmien2
Mniejszy - równy (<=)	Zwraca true jeżeli lewy operand jest mniejszy lub równy prawemu operandowi.	Otrzymamy prawdę w przypadku gdy: zmien1 <= zmien2 zmien2 <= 3
operator koniunkcji &&	Koniunkcja jest to inaczej iloczyn	true && true daje true , true && false daje false , false && true daje false , false && false daje false
operator alternatywy	Alternatywa to inaczej suma	true true daje true , true false daje true , false true daje true , false false daje false
Operator negacji !	Negacja jeśli występuje przed operandem, powoduje jego negację.	!(true) daje false
Operator sumy +	Dodawanie	2+1=3
Operator odejmowania -	Odejmowanie	2- 1=1
Operator modulo %	Modulo (reszta z dzielenia)	14%5=4
Operator dzielenia /	Dzielenie	12/2=6
Operator mnożenia *	Mnożenie	2*3=6
Operator inkrementacji ++	Zwiększenie wartości o jeden	2++ = 3

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Kształty

Najprostsze figury geometryczne dostępne w processingu to:

	point(x,y);
	line(x1, y1, x2, y2);
	triangle(x1, y1, x2, y2,x3,y3);
	quad (x1, y1, x2, y2,x3,y3,x4,y4);
	rect(x,y,width,height);
	ellipse(x,y,width,height);
	bezier(x1, y1, cx1, cy1, cx2, cy2, x2, y2);

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Kolor

Już wiesz

Przypomnijcie sobie o modelach koloru, z rozdziału „Grafika i multimedia”, z podręcznika do gimnazjum.

Wartości jakie przyjmuje każda z funkcji odpowiedzialna za generowanie koloru to np.: `background( R, G, B)`; Gdzie R – jest wartością z przedziału od (0-255) dla koloru czerwonego G – jest wartością z przedziału od (0-255) dla koloru niebieskiego B – jest wartością z przedziału od (0-255) dla koloru zielonego

		
<code>background(0);</code>	<code>background(125);</code>	<code>background(255);</code>

będzie równoważne z:

		
<code>background(0,0,0);</code>	<code>background(125, 125, 125);</code>	<code>background(255, 255,255);</code>

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Oprócz wartości RGB stosowana jest też czwarta wartość – „alpha – A”, określająca przezroczystość koloru. Każda z funkcji odpowiedzialna za kolor może przyjmować więc wartości (R, G, B, A);

Metoda Setup

Processing posiada kilka predefiniowanych metod, które określają strukturę programu i odpowiadają za najważniejsze dla każdego projektu części. Pierwszą taką metodą jest funkcja `setup`:

```
void setup() {  
}
```

Jest to metoda wejściowa dla naszego programu i wewnątrz niej definiujemy wszelkie ustawienia programu (takie jak rozmiar okna, silnik renderujący itp.), inicjalizujemy obiekty klas i zmienne. Po skompilowaniu programu otrzymamy okno aplikacji o domyślnym rozmiarze 100 na 100 pikseli. Uruchomić program można na trzy sposoby:

- poprzez pierwszą od lewej ikonkę na górnym pasku (ikona „play”),
- poprzez menu Sketch->Run,
- poprzez skrót klawiaturowy Ctrl+R.

Można również samemu ustawić rozmiar okna poprzez następujące wywołanie wewnątrz funkcji setup:

```
size(700, 700);
```

Jeśli nie określimy silnika renderującego, domyślnie zostanie ustawiony silnik JAVA2D, który pozwala na operacje dwuwymiarowe. Jest on dokładny, lecz nieco wolniejszy niż silnik P2D. Dostępne opcje silników (od wersji 2.0 środowiska Processing.org) to:

- **P2D** (Processing 2D) – silnik dwuwymiarowy wykorzystujący możliwości kart graficznych wspierających OpenGL 2.0.
- **P3D** (Processing 3D) – silnik trójwymiarowy wykorzystujący możliwości kart graficznych wspierających OpenGL 2.0.
- **PDF** – silnik pozwalający zapisać grafikę dwuwymiarową bezpośrednio do pliku PDF, zamiast wyświetlać ją na ekranie. Co więcej, elementy wektorowe zostaną zapisane również jako wektorowe, co umożliwia późniejszą manipulację nimi w programach typu Adobe Illustrator. Przydatny, gdy potrzebujemy wysokiej jakości grafiki do druku czy edycji.

W starszych wersjach środowiska dostępny jest również:

OPENGL – silnik trójwymiarowy, do którego potrzebny jest sprzęt, który wspiera OpenGL, wykorzystuje bibliotekę JOGL (Java for OpenGL) i pozwala na rendering trójwymiarowy na stronach www.

P2D oraz **P3D** – były to sprzętowe implementacje nie wykorzystujące silnika OpenGL.

Aby korzystać z możliwości oraz zapisu do PDF, należy zaimportować odpowiednią bibliotekę. Podobnie jest w przypadku OpenGL'a w starszych wersjach środowiska (od wersji 2.0 OpenGL jest podstawową składową środowiska). Rodzaj silnika określa się poprzez podanie trzeciego parametru metody size np.

```
size(400, 200, P3D);
```

lub:

```
size(400, 200, PDF, „nazwa_pliku.pdf”);
```

Powyższy kod (pierwszy przykład) jest wszystkim, czego potrzeba, by uruchomić okno dla OPENGL'a

Metoda draw

Drugą istotną funkcją jest funkcja draw:

```
void draw( ){ }
```

Jest ona wywoływana cyklicznie w każdej klatce działania programu, zatem w niej należy umieścić kod odpowiedzialny za rysowanie i wszelkie operacje, które powinny odbywać się

co klatkę. Można prześledzić to na przykładzie. Należy dodać dwie zmienne typu całkowitego do kodu programu, które będą przechowywać położenie rysowanego obiektu:

```
int posx = 0, posy = 0;
```

Linie tę trzeba umieścić poza omawianymi wcześniej metodami. Teraz w metodzie *draw* można dodać rysowanie kwadratu i zwiększenie położenia jego wyrysowania: *posx =*

```
posy++;
```

```
rect(posx, posy, 50, 50);
```

Całość powinna wyglądać następująco:

```
int posx = 0, posy = 0;
```

```
void setup(){
```

```
size(400, 200);
```

```
}
```

```
void draw(){
```

```
posx = posy++;
```

```
rect(posx, posy, 50, 50);
```

```
}
```

Aby wyrysować tylko jedną klatkę lub zatrzymać animację należy skorzystać z metody *noLoop*. Jeśli umieści się ją wewnątrz metody *setup*, program wywoła metodę *draw* tylko raz:

```
void setup(){
```

```
size(400, 200);
```

```
noLoop();
```

```
}
```

Funkcją o działaniu odwrotnym do *noLoop* jest funkcja *loop*, która jest domyślnie wywoływana przy starcie programu, (jeśli nie zadeklaruje się inaczej). Jeśli do kodu zostanie dopisane wywołanie tej funkcji na końcu metody *draw*, to otrzymany efekt będzie taki, jakby w ogóle nie dodano wywołania pary *loop* i *noLoop*. Aby powrócić do widoku animacji należy usunąć oba wpisy (*loop* oraz *noLoop*).

W powyższym przykładzie kwadrat rysowany jest od lewego górnego narożnika, jednak można to zmienić, przez dodanie, dla wbudowanych funkcji rysujących kształty, ustawienia, czy ma on być rysowany od narożnika, czy względem środka. Dla metody *rect* wygląda to następująco:

```
rectMode(CENTER);
```

By powrócić do domyślnego ustawienia, należy skorzystać z metody:

```
rectMode(CORNER);
```

Analogiczne metody istnieją dla pozostałych metod rysujących – jest to omówione w dokumentacji programu Processing.

Kolor obrysu i wypełnienia zostały ustawione na wartości domyślne. Zmienić je można przy pomocy następujących metod:

```
fill(0);
```

```
stroke(255);
```

W powyższym przykładzie kolor wypełnienia ustawiony jest na czarny, a kolor obrysu na

biały. Można także użyć innego koloru, ale zostanie to omówione później. Jeśli wywołanie tych metod zostaną umieszczone w metodzie `setup`, będą dotyczyły wszystkich rysowanych obiektów tak długo, póki nie zostaną zmienione. Można je także umieścić wewnątrz funkcji `draw`. Wtedy dotyczyć będą obiektów narysowanych po ich wywołaniu. Dlatego można określić Processing jako maszynę stanową – raz ustawiony stan jest używany globalnie, aż do jego zmiany. Jeżeli chce się korzystać jedynie z wypełnienia lub tylko z obrysu, należy skorzystać z jednej z poniższych metod:

```
noStroke();
```

```
noFill();
```

Wypełnienie i/lub obrys będą wyłączone do ponownego wywołania metody ustawiającej ich kolor (czyli *fill* lub *stroke*).

Metoda exit

Ostatnią główną metodą (lecz nie ostatnią dostępną), jest funkcja wyjścia. W niej należy określić wszystkie czynności konieczne do wykonania przed zamknięciem programu. Metoda ta nazywa się *exit*. W metodzie tej można wyświetlić komunikat podczas zamykania programu:

```
void exit(){  
println(„Koniec programu”);  
}
```

Po uruchomieniu programu komunikat ten jest wyświetlany, ale program nie zostanie zamknięty. Należy samemu obsłużyć zamknięcie okna aplikacji, przez dopisanie na końcu programu, instrukcji:

```
super.exit();
```

Spowoduje to wywołanie domyślnej obsługi metody *exit*. Metody tej można również użyć w dowolnym miejscu kodu np. w reakcji na działanie użytkownika. Jej wywołanie spowoduje zamknięcie aplikacji. Dla testów można dopisać jej wywołanie np. na końcu metody *draw*, która teraz powinna wyglądać następująco:

```
void draw(){  
background(100);  
posx = posy++;  
rect(posx, posy, 50, 50);  
exit();  
}
```

Aby obejrzeć efekt działania programu, należy jednak usunąć tę linijkę

Własne funkcje i błędy kompilacji.

W celu poprawienia czytelności kodu można rysowanie kwadratu przenieść do osobnej funkcji, i jej wywołanie umieścić w metodzie *draw*:

```

void draw(){
background(100);
posx = posy++;
rysujKwadrat(posx, posy);
}
void rysujKwadrat(int _x, int _y){
rect(_x, _y, 50, 50);
}

```

Teraz należy zapisać program. Należy pamiętać, by plik z kodem źródłowym programu znajdował się w folderze o tej samej nazwie co nazwa pliku (czyli plik „test.pde” powinien znajdować się w folderze „test”) – inaczej otrzyma się błąd. Jeśli chodzi o błędy kompilacji w środowisku Processing, warto wspomnieć o jednej, nie do końca oczywistej, właściwości tego środowiska. By to zobrazować, należy wygenerować celowy błąd, poprzez wywołanie nieistniejącej funkcji. Na dole okna pojawia się wtedy treść powstałego błędu oraz dodatkowo numer linii, w której on się pojawił (liczba „10” w prawym dolnym rogu okna). Należy jednak uważać, bowiem w tym miejscu standardowo pojawia się numer bieżącej linii, w programie. Po kliknięciu gdziekolwiek na kod, zniknie zaznaczenie linii błędu i jej numer. Żeby się dowiedzieć o którą linię chodziło, należy ponownie uruchomić program.

Tryby pracy

Od wersji 2.0 środowisko Processing.org oferuje kilka trybów pracy (a nowe są cały czas tworzone i dodawane). Są to:

- **Standard** – podstawowy i domyślny tryb dostępny w Processing'u. Pozwala tworzyć i eksportować aplikacje desktop'owe jednocześnie na różne platformy (Windows, Mac OS, Linux). Umożliwia także zapis projektu jako aplet w środowisku java uruchamiany w przeglądarce (wraz ze wsparciem dla 3D), jednak to rozwiązanie nie jest już rozwijane (zamiast tego zaleca się wykorzystanie trybu JavaScript). Tryb ten zalecany jest jako domyślny podczas nauki i pisania pierwszych programów w środowisku Processing.org.
- **Android** – zaprezentowany po raz pierwszy w wersji 1.5 środowiska Processing.org, był pierwszym dostępnym, poza standardowym, trybem. Umożliwia tworzenie i uruchamianie projektów na urządzeniach z systemem Android. Począwszy od wersji 2.0 środowiska, wspierane są tylko wersje wyższe niż Android 2.3.3 (Gingerbread, API 10).
- **JavaScript** – pozwala tworzyć aplikacje uruchamiane bezpośrednio z przeglądarki przy użyciu biblioteki Processing.js (<http://processingjs.org/>). To rozwiązanie zastępuje aplety znane z wcześniejszych wersji środowiska.

Tryb można zmienić poprzez menu rozwijane w prawym górnym obszarze okna programu (Ilustracja 5).

Instalacja zewnętrznych bibliotek

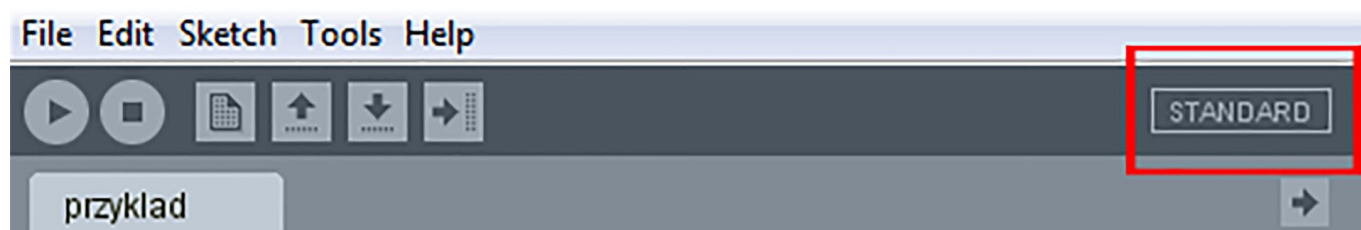
Gdy w API Processing'u nie można znaleźć potrzebnej metody czy klasy, warto poszukać dodatkowych bibliotek. Biblioteki takie znajdują się na stronie środowiska - <http://processing.org/reference/libraries/> , a także na wielu innych stronach. Samo środowisko zawiera już kilka przydatnych bibliotek jak np. Video, Network czy PDF Export. Jedyne co trzeba zrobić, by zainstalować pobraną bibliotekę, to przekopiować wszystkie pliki (zachowując strukturę folderów stworzoną przez autora biblioteki) do odpowiedniego folderu. Od wersji 1.0 i wyżej, wszelkie biblioteki potrzebne do wykonania projektu umieszcza się w podfolderze „libraries” w folderze „szkicownika” (czyli jako podfolder folderu sketchbook, którego lokalizacja została ustalona wcześniej podczas konfiguracji). By dodać bibliotekę do projektu, wystarczy, że wykonać jeden z poniższych kroków:

- skorzystać z menu Sketch->Import Library wybierając żadaną bibliotekę,
- wpisując na początku programu słowo „import” i po spacji nazwę biblioteki wraz z pakietem, w którym się znajduje.

Niezależnie od wybranego sposobu, otrzymuje się identyczny efekt. Przykładowo, jeśli chcemy zaimportować bibliotekę do obsługi wideo, na początku pliku należy wpisać:

```
import processing.video.*;
```

co pozwoli na korzystanie z klas i metod tej biblioteki.



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Obsługa klawiatury

Za obsługę klawiatury odpowiada kolejna główna metoda, a mianowicie:

```
void keyPressed() {}
```

Ustawienie wewnątrz niej obsługi klawiatury zapewnia, że będzie ona wykonywana w każdej klatce. Można jednak pominąć tę metodę i umieścić obsługę w każdej innej metodzie (np. draw), gdyż nie zawiera ona żadnych parametrów. Zdarzenie klawiatury nie jest przekazywane do metody, a zachowywane w zmiennych globalnych, które zostaną omówione w dalszej części. Ta metoda gwarantuje jednak wywołanie w każdej klatce i estetykę kodu. Zamiast niej można sprawdzać flagę naciśnięcia klawisza o tej samej nazwie co metoda (czyli keyPressed), w następujący sposób:

```
void draw() {  
  if(keyPressed) {
```

```
//tutaj właściwa obsługa klawiatury  
{  
}
```

Zaleca się jednak korzystanie z metody `keyPressed`. Aby sprawdzić, który klawisz został wciśnięty, należy użyć zmiennej globalnej `key`:

```
switch(key){  
case 'a': bgColor = 0; break;  
case 's': bgColor = 100; break;  
case 'd': bgColor = 200; break;  
}
```

Można dodać także zmienną `bgColor`, która będzie odpowiadać za zmianę koloru tła i przekazać ją jako parametr do funkcji `background` w metodzie rysującej. Dzięki temu, po naciśnięciu klawisza „a” tło zmieni się na czarne, po naciśnięciu klawisza „s” na ciemnoszare, a naciśnięcie klawisza „d” zmieni je na jasnoszare. Co jednak zrobić, gdy jest to klawisz specjalny, jak np. strzałka czy enter? Należy wówczas skorzystać z kolejnej zmiennej globalnej – `keyCode`. Zawiera ona kod ASCII klawisza, który został wciśnięty. Aby więc dopisać obsługę zatrzymania i wznowienia animacji przy pomocy strzałek na klawiaturze należy napisać kod:

```
if (key == CODED) {  
  switch(keyCode){  
  case UP: loop(); break;  
  case DOWN: noLoop(); break;  
  }  
}
```

Powyższy przykład zawiera pewne stałe zdefiniowane w języku Processing. Łącząc oba przypadki poprzez dodanie słowa kluczowego „else” przed obsługą zwykłych klawiszy, otrzymamy kompletną obsługę klawiatury.

Obsługa myszy

Obsługa myszy, tak jak w przypadku klawiatury, możliwa jest na dwa sposoby. Najczęściej wykorzystuje się zmienne globalne (dostępne w każdym miejscu programu) przechowujące położenie kursora myszy: `mouseX` oraz `mouseY`. Po zakomentowaniu wywołania utworzonej poprzednio funkcji rysującej kwadrat, można napisać wywołanie rysowania z położeniem kursora myszy jako pozycją tworzonego kwadratu. Dodając do tego poznaną wcześniej metodę, by rysować kwadrat od środka, otrzyma się efekt podążania kwadratu za kursorem myszy (kursor w środku obiektu). Fragment kodu za to odpowiedzialny powinien wyglądać następująco:

```
//rysujKwadrat(posx, posy);  
rectMode(CENTER);  
rect(mouseX, mouseY, 50, 50);
```

Główną metodą, odpowiadającą za obsługę kliknięcia myszy jest `mousePressed` bez żadnych parametrów. Jeśli umieści się ją w programie i przeniesie do niej napisane przed chwilą dwie linie rysujące kwadrat, to otrzyma się białe tło i „mignięcie” kwadratu przy kliknięciu przycisku myszy. Dlaczego tak się dzieje? Co należy zmienić w kodzie, by rysowany kwadrat pozostawał widoczny? Trzeba zakomentować (lub usunąć) pierwszą linię w metodzie `draw`, czyli wypełnienie tła kolorem. Alternatywnie można ją przenieść do metody `setup`, dzięki czemu będzie wywołana tylko raz z zadaniem kolorem jako parametrem. Teraz rysowany na kliknięcie kwadrat powinien pozostawać na obrazie. Dodatkowo można sprawdzić, który przycisk został wciśnięty. Do tego celu służy kolejna globalna zmienna – `mouseButton`. Najlepiej zobrazuje to poniższy przykład:

```
switch(mouseButton) {  
case LEFT: println(„Wciśnięto LEWY przycisk”); break;  
case RIGHT: println(„Wciśnięto PRAWY przycisk”); break;  
case CENTER: println(„Wciśnięto ŚRODKOWY przycisk”); break;  
}
```

Inne przydatne podczas obsługi myszy metody to:

- `mouseClicked()` – dla obsługi kliknięcia (w przeciwieństwie do naciśnięcia jest to złożenie dwóch akcji – wciśnięcia i puszczenia przycisku myszy),
- `mouseDragged()` – dla obsługi przeciągania,
- `mouseMoved()` – dla obsługi ruchu myszą (bez wciśnięcia przycisku myszy),
- `mouseReleased()` – odwrotne niż `mousePressed()` – puszczenie przycisku myszy.

Operowanie na plikach wektorowych

Transformacje w języku Processing są analogiczne do znanych z OpenGL, a mianowicie przekształcany jest cały obecny układ współrzędnych. Zatem narysować obiekt w punkcie (100, 200) można na dwa sposoby:

```
rect(100, 200, 50, 50);
```

lub

```
translate(100, 200);  
rectMode(CENTER);  
rect(0, 0, 50, 50);
```

Pierwszy przypadek jest oczywisty, natomiast w drugim najpierw przenosi się cały układ do zadanego punktu, a następnie rysuje obiekt. Aby narysować kwadrat również w punkcie (100, 200), ale obrócony o 45 stopni, trzeba skorzystać ze złożenia przekształceń – translacji oraz obrotu – w następujący sposób:

```
translate(100, 200);  
rotate(PI/4);  
rectMode(CENTER);  
rect(0, 0, 50, 50);
```

Zastosowana metoda obrotu jako argument przyjmuje kąt obrotu w radianach. Jeśli kąt

obrotu określony jest w stopniach, to należy skorzystać z metody *radians()*, konwertującej stopnie na radiany. Obrazuje to kod:

```
rotate(radians(45));
```

Można przenieść teraz ten kod do funkcji rysującej na kliknięcie myszy tak, by zamiast stałego punktu, kwadrat rysował się w miejscu, w którym zostanie wciśnięty przycisk myszy (oczywiście obrócony o 45 stopni). Efekt widoczny jest na ilustracji 1, na której można zauważyć, że narysowane kwadraty są „poszarpane” na brzegach (widoczne zwłaszcza na powiększeniu). Aby tego uniknąć należy zastosować metodę *smooth()*, którą trzeba dodać do ustawień programu (czyli metody *setup()*).

Trzecią przydatną transformacją jest skalowanie – metoda *scale*, której przekazuje się liczbę skali wzdłuż osi X oraz Y:

```
scale(2, 1);
```

Dopisując powyższą linię do kodu (po translacji i obrocie), otrzyma się obrócony prostokąt. Aby narysować obrócony kwadrat (jak powyżej) w zadanym punkcie, a potem kwadrat (nie obrócony) w punkcie (0, 0) należy napisać kod:

```
pushMatrix();  
translate(100, 200);  
rotate(radians(45));  
rectMode(CENTER);  
rect(0, 0, 50, 50);  
popMatrix();  
rect(0, 0, 50, 50);
```

Operowanie na obrazach rastrowych.

Aby pracować na poziomie pikseli obrazu, należy wykonać kilka kroków, a mianowicie: załadować piksele, dokonać zmian na nich, zaktualizować piksele na obrazie. Ten proces pokazany jest na przykładzie funkcji zmieniającej wyświetlany obraz na negatyw.

```
void invertColors(){  
  loadPixels();  
  for(int x=0; x<width; x++){  
    for(int y=0; y<height; y++){  
      float R = 255 - red(pixels[x+y*width]);  
      float G = 255 - green(pixels[x+y*width]);  
      float B = 255 - blue(pixels[x+y*width]);  
      pixels[x+y*width] = color(R, G, B);  
    }  
  }  
  updatePixels();  
}
```

Metoda *loadPixels()* służy do załadowania pikseli obrazu do tablicy *pixels*, którą później

wykorzystuje się do przeprowadzenia zmian. Jest to tablica jednowymiarowa, a więc można zastosować następującą pętlę:

```
for(int i=0; i<pixels.length; i++){  
float R = 255 - red(pixels[i]);  
float G = 255 - green(pixels[i]);  
float B = 255 - blue(pixels[i]);  
pixels[i] = color(R,G,B);  
}
```

Jednak częściej trzeba poruszać się po obrazie w dwóch wymiarach. Wymaga to indeksowania tablicy jednowymiarowej tak, jakby była dwuwymiarową. Globalne właściwości `width` oraz `height` odnoszą się do wymiarów obrazu (są identyczne z wartością ustawioną poprzez `size()`). Metody `red`, `green` oraz `blue` pozwalają pobrać odpowiednie składowe koloru, by je zmodyfikować. Po dokonaniu zmian (odwróceniu koloru), można zapisać kolor piksela przy wykorzystaniu metody `color` z parametrami, odpowiednio – składową czerwoną, zieloną i niebieską koloru. Ostatnim krokiem jest zaktualizowanie pikseli wyświetlanych na ekranie – `updatePixels()`. Jest to technika podobna do podwójnego buforowania w OpenGL. By uniknąć nieprzyjemnych efektów zmiany pikseli na obrazie podczas jego wyświetlania, należy je zmodyfikować w osobnym buforze (tablica `pixels`), po czym zamienić te tablice.

By zobaczyć efekt, trzeba dopisać wywołanie utworzonej funkcji na końcu metody obsługującej wciśnięcie przycisku myszy (zatem po każdorazowym narysowaniu kwadratu powinno się otrzymać negatyw obrazu). Analogicznie można pracować na pikselach wczytanych obrazów. Należy w tym celu utworzyć zmienną globalną, która będzie przechowywała wczytany obraz.

```
PImage img;
```

Typ `PImage` to typ danej obrazu. W metodzie `setup` należy wczytać przykładowy obraz do zmiennej:

```
img = loadImage("test.jpg");
```

W tym miejscu można użyć dowolnego obrazu (obsługiwane typy plików to `.gif`, `.jpg`, `.tga`, `.png`). Przy tak podanej ścieżce musi się on znajdować w folderze projektu (tam gdzie plik `.pde`). Aby ustawić go jako tło programu, należy dodać następującą linię:

```
image(img, 0, 0);
```

Kod ten umieszcza się w metodzie `setup` zaraz po załadowaniu obrazka. Gdyby został on umieszczony w metodzie `draw`, to za każdym razem przysyłane byłyby kwadraty rysowane przy wciśnięciu klawisza myszy. W ten sposób wskazany obrazek zostanie wyrysowany począwszy od punktu (0, 0). Oprócz położenia, jako kolejne dwa parametry można przekazać do metody `image` rozmiar, do jakiego należy przeskalować obrazek, np.

```
image(img, 0, 0, 100, 100);
```

Dodatkowe informacje na temat tej metody można znaleźć w dokumentacji

http://processing.org/reference/image_.html

Na pikselach obrazka również można pracować w sposób analogiczny do opisanego powyżej (który dotyczył pikseli całego obrazu). Różnica polegać będzie na wskazaniu

odpowiedniej tablicy pixels, np.

```
img.loadPixels();  
for(int i=0; i<img.pixels.length; i++){  
float R = 255 - red(img.pixels[i]);  
float G = 255 - green(img.pixels[i]);  
float B = 255 - blue(img.pixels[i]);  
img.pixels[i] = color(R,G,B);  
}  
img.updatePixels();
```

Tak zmodyfikowany obraz należy ponownie wyświetlić przy użyciu metody *image*.

Wczytywanie i zapis plików

Aby zapisać utworzony plik obrazu stosuje się instrukcję:

```
saveFrame( „output.png” );
```

Zapisanie więcej niż jednego obrazu (jednej klatki) lub serii klatek, można wykonać poprzez odpowiednią konstrukcję nazwy pliku wyjściowego:

```
saveFrame( „output-###.png” );
```

Spowoduje to zapis każdej klatki do pliku z odpowiednią nazwą tj. output-001.png, output-002.png itd. Jednak przed uruchomieniem powyższego przykładu należy dodać w metodzie setup jeszcze ustawienie ilości klatek na sekundę:

```
frameRate(1);  
= loadStrings( „plik.txt” );
```

Kolejne linie pliku zostaną wczytane jako osobne elementy tablicy typu String. Załóżmy, że każda linia zawiera dane oddzielone od siebie pewnym charakterystycznym znakiem, np. „:”. By uzyskać dostęp do poszczególnych treści (czyli rozdzielić linię na fragmenty oddzielone dwukropkami) można się posłużyć metodą *splitTokens*, w następujący sposób:

```
String[] daneLinii = splitTokens(linie, „:” );
```

podając jako parametr ciąg znaków do podziału oraz znak (lub znaki, bo możemy podać ich kilka naraz) podziału. Zapis treści tekstowych dokonuje się następująco:

```
saveStrings( „nazwa_pliku.txt”, tresc_do_zapisu);
```

gdzie tresc_do_zapisu to tablica (tablica String[]), w której każdy element to kolejna linia pliku, który zapisujemy. Aby wybrać plik do wczytania podczas działania programu należy napisać następujący kod:

```
String sciezka_do_pliku = selectInput();  
if (sciezka_do_pliku == null) {  
println(„Nie wybrano żadnego pliku...”);  
}
```

W pierwszej linii zostaje otworzone okno dialogowe do wyboru pliku i wskazana zmienna, do której ma być przypisana ścieżka wybranego pliku. Następnie należy sprawdzić, czy plik został wybrany, czy naciśnięto przycisk „anuluj” lub pojawił się inny błąd, co spowodowało,

że zmienna jest pusta. Wtedy można wyświetlić stosowny komunikat (jak w powyższym przykładzie). Jeśli nie ma błędu, można przekazać ścieżkę do pliku do odpowiedniej metody wczytującej (omawianej wcześniej). Wczytywać dane można z wielu innych formatów plików oraz źródeł – wystarczy zajrzeć do dokumentacji lub przejrzeć dodatkowe biblioteki. Często wykorzystywany jest format XML.

Ćwiczenie – Przykład Przewodnika

W przypadku przewodnika multimedialnego, bardzo istotne jest wykorzystanie plików multimedialnych (dźwięk, video i obraz). Należy do tego celu wykorzystać zewnętrzne biblioteki, znajdujące się na stronie środowiska – <http://processing.org/reference/libraries/> – takie jak:

```
import processing.video.*; //biblioteka video
import ddf.minim.*; //biblioteka dźwięku
```

W programie należy zadeklarować zmienne tekstowe – np. do opisów miast, zmienne pomocnicze typu całkowitego *int* oraz zmienne odpowiedzialne za obraz.

```
//zmienne
int ekran;
int miasto;
int ikona_foto_over;
int ikona_video_over;
int ikona_stat_over;
int muzyka;
String nazwa;
String opis;
PImage intro;
PImage tlo_menu;
String url = „statystyki/index.html”;
```

Dodatkowo trzeba zadeklarować zmienne odpowiedzialne za obrazy ikon i zdjęć:

```
//ikony
PImage ikona_lodz;
PImage ikona_warszawa;
PImage ikona_krakow;
PImage ikona_poznan;
PImage ikona_gdansk;
PImage ikona_wroclaw;
PImage ikona_zamknij;
PImage ikona_foto;
PImage ikona_video;
PImage ikona_muzyki;
PImage ikona_statystyk;
//zdjęcia
PImage foto1s;
PImage foto2s;
PImage foto1b;
```

oraz obiekt typu movie z biblioteki processing.video.

```
//video
```

```
Movie movie;
```

Elementy związane z obsługą dźwięku pochodzą z biblioteki *ddf.minim*. Aby rozpocząć korzystanie z *Minim* należy najpierw zadeklarować instancję obiektu *minim*, którą można następnie wykorzystać do załadowania plików audio lub obsługi wejść i wyjść:

```
//muzyka
Minim minim;
AudioPlayer player;
```

Funkcja setup()

Funkcja ta wywoływana jest tylko raz przy uruchamianiu programu, ustalając wszystkie niezbędne warunki początkowe do działania aplikacji.

Pierwsze polecenie określa rozdzielczość okna aplikacji ():

```
size(700, 700);
```

W funkcji zostanie wywołane polecenie *loadImage(url, „png”)*, które powoduje wczytanie obrazka typu png z adresu url.

Wstawienie obrazka, którego za lewy górny róg będzie w punkcie obszaru roboczego o współrzędnej (x,y) realizuje polecenie:

image(obrazek, x, y). Jeśli nie planuje się dalszej pracy na obrazku, można użyć szybszego polecenia przenoszącego obrazek na obszar roboczy: *set(x, y, obrazek)*. Jeśli chcemy mieć możliwość manipulacji na zmiennych, należy wczytać obrazy do wcześniej zadeklarowanych zmiennych, poleceniem:

```

intro = loadImage(„grafika/intro.jpg”);
tlo_menu = loadImage(„grafika/tlo_menu.jpg”);
ikona_lodz = loadImage(„grafika/ikona_lodz.png”);
ikona_warszawa = loadImage(„grafika/ikona_warszawa.png”);
ikona_krakow = loadImage(„grafika/ikona_krakow.png”);
ikona_poznan = loadImage(„grafika/ikona_poznan.png”);
ikona_gdansk = loadImage(„grafika/ikona_gdansk.png”);
ikona_wroclaw = loadImage(„grafika/ikona_wroclaw.png”);
ikona_zamknij = loadImage(„grafika/ikona_zamknij.png”);
ikona_foto = loadImage(„grafika/ikona_foto.png”);
ikona_video = loadImage(„grafika/ikona_video.png”);
ikona_muzyki = loadImage(„grafika/ikona_muzyki.png”);
ikona_statystyk = loadImage(„grafika/ikona_statystyk.png”);

```

Dla plików muzycznych należy dokonać inicjalizacji obiektu `minim` oraz załadować plik muzyczny `music.mp3`:

```

minim = new Minim(this); //inicjalizacja obiektu minim do obsługi muzyki
player = minim.loadFile(„music.mp3”); //ładowanie pliku z muzyką z katalogu data
player.play();

```

Ostatnie linijki funkcji `setup` służą do zainicjowania stanów zmiennych:

```

ekran = -1;
miasto = 0;
muzyka = 1;
ikona_foto_over=0;
ikona_video_over=0;

```

Funkcja `draw()`

Funkcja `draw()` będzie wykonywana ciągle, zaraz po wykonaniu instrukcji `setup()`, do czasu kiedy zostanie przerwane wykonywanie programu lub będzie uruchomiona inna instrukcja. W przypadku projektowanego przewodnika należy utworzyć różne funkcje rysujące poszczególne ekrany dla każdego miasta, w zależności od wybranej pozycji z menu. Będą tutaj wywoływane w pętli warunki, które będą sprawdzały jakie okno powinno zostać uruchomione, w zależności od wartości zmiennej `ekran`:

```

void draw() {
  switch(ekran) {
    case -1:
      ekran_powitalny();
      break;
    case 0:
      ekran_menu();
      break;
    case 1:
      ekran_miasta();
      break;
    case 2:
      ekran_foto();
      break;
    case 3:
      ekran_video();
      break;
  }
}

```

Funkcje okien/ekranów aplikacji

Poniżej znajdują się deklaracje poszczególnych ekranów:

Ekran powitalny:

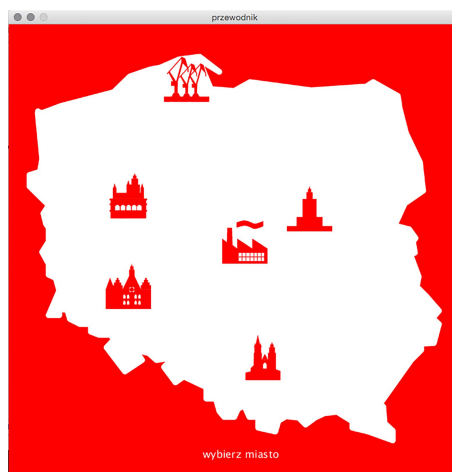


Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Kod ekranu powitalnego:

```
1
2 void ekran_powitalny(){
3     background(intro);
4     fill(255, 255, 255);
5     textSize(16);
6     text("kliknij myszką", 300, 670);
7 }
```

Ekran menu głównego z miastami



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Kod ekranu menu główne, rozmieszcza wcześniej zaprojektowane ikony na wcześniej przygotowanym tle (kontur Polski):

```
void ekran_menu(){
    image(tlo_menu, 0, 0, width, height);
    image(ikona_lodz, 330, 300, 70, 70);
    image(ikona_warszawa, 430, 250, 70, 70);
    image(ikona_krakow, 360, 480, 70, 70);
    image(ikona_poznan, 150, 230, 70, 70);
    image(ikona_gdansk, 240, 50, 70, 70);
    image(ikona_wroclaw, 150, 370, 70, 70);

    image(ikona_zamknij, 630, 20, 50, 50);
    image(ikona_muzyki, 570, 20, 50, 50);

    fill(255, 255, 255);
    textSize(16);
```

```
    text(„wybierz miasto”, 300, 670);  
}
```

Ekran miasta

Ekran poszczególnych miast, będą się różnić w zależności od wczytanego miasta. Dlatego do zmiennych odpowiedzialnych za film, obraz i statystyki zostaną przypisane inne obiekty. Wykorzystany zostanie do tego celu warunek *switch(miasto)*:

```
void ekran_miasta(){  
  
    switch(miasto) {  
    case 1:  
        nazwa=„Łódź”;  
        opis = „Łódź – miasto na prawach powiatu w środkowej Polsce. Siedziba władz województwa łódzkiego”;  
        foto1s = loadImage(„grafika/lodz1s.jpg”);  
        foto2s = loadImage(„grafika/lodz2s.jpg”);  
        foto1b = loadImage(„grafika/lodz1b.jpg”);  
        movie = new Movie(this, „lodz.mp4”);  
        url = „http://www.lodz.pl”;  
        break;  
    case 2:  
        nazwa=„Warszawa”;  
        opis = „informacje o wybranym mieście, informacje o wybranym mieście, informacje o wybranym mieście”;  
        foto1s = loadImage(„grafika/warszawa1s.jpg”);  
        foto2s = loadImage(„grafika/warszawa2s.jpg”);  
        foto1b = loadImage(„grafika/warszawa1b.jpg”);  
        movie = new Movie(this, „warszawa.mp4”);  
        url = „http://www.warszawa.pl”;  
        break;  
    case 3:  
        nazwa=„Kraków”;  
        opis = „informacje o wybranym mieście, informacje o wybranym mieście, informacje o wybranym mieście”;  
        foto1s = loadImage(„grafika/krakow1s.jpg”);  
        foto2s = loadImage(„grafika/krakow2s.jpg”);  
        foto1b = loadImage(„grafika/krakow1b.jpg”);  
        movie = new Movie(this, „krakow.mp4”);  
        url = „http://www.krakow.pl”;  
        break;  
    case 4:  
        nazwa=„Poznań”;  
        opis = „informacje o wybranym mieście, informacje o wybranym mieście, informacje o wybranym mieście”;  
        foto1s = loadImage(„grafika/poznan1s.jpg”);  
        foto2s = loadImage(„grafika/poznan2s.jpg”);  
        foto1b = loadImage(„grafika/poznan1b.jpg”);  
        movie = new Movie(this, „poznan.mp4”);  
        url = „http://www.poznan.pl”;  
        break;  
    case 5:  
        nazwa=„Gdańsk”;  
        opis = „informacje o wybranym mieście, informacje o wybranym mieście, informacje o wybranym mieście”;  
        foto1s = loadImage(„grafika/gdansk1s.jpg”);  
        foto2s = loadImage(„grafika/gdansk2s.jpg”);  
        foto1b = loadImage(„grafika/gdansk1b.jpg”);  
        movie = new Movie(this, „gdansk.mp4”);  
        url = „http://www.gdansk.pl”;  
        break;  
    case 6:  
        nazwa=„Wrocław”;  
        opis = „informacje o wybranym mieście, informacje o wybranym mieście, informacje o wybranym mieście”;  
        foto1s = loadImage(„grafika/wroclaw1s.jpg”);  
        foto2s = loadImage(„grafika/wroclaw2s.jpg”);  
        foto1b = loadImage(„grafika/wroclaw1b.jpg”);  
        movie = new Movie(this, „wroclaw.mp4”);  
        url = „http://www.wroclaw.pl”;  
    }
```

```
    break;
}
```

W przykładzie wpisane zostały jedynie informacje dla miasta Łodzi. Informacje o innych miastach można znaleźć w Internecie i podmienić tekst przypisany do zmiennej opis, odpowiednio dla każdego miasta.

Pozostałe elementy takie jak ramki, zdjęcia czy tekst są wspólne dla każdego miasta.. Zmienne, dzięki wcześniejszemu poleceniu, będą miały odmienne wartości. Dzięki temu, w spójny sposób choć charakterystyczny dla każdego miasta, zostaną rozmieszczone elementy menu:

```
image(tlo_menu,0,0,width,height);

// ramka na nazwę
fill(255, 0, 0, 150);
stroke(255, 255, 255);
strokeWeight(10);
rect(20,20,280,80);

fill(255, 255, 255);
textSize(36);
text(nazwa, 50, 70);

// ramka na dane statystyczne
image(ikona_statystyk,20,100,280,186);
noFill();
stroke(255, 255, 255);
strokeWeight(10);
rect(20,100,280,186);

// zdjęcie 1
image(foto1s,20,300,280,186);
if(ikona_foto_over == 1){
    image(ikona_foto,20,300,280,186);
}
noFill();
stroke(255, 255, 255);
strokeWeight(10);
rect(20,300,280,186);

// zdjęcie 2
image(foto2s,20,494,280,186);
if(ikona_video_over == 1){
    image(ikona_video,20,494,280,186);
}
noFill();
stroke(255, 255, 255);
strokeWeight(10);
rect(20,494,280,186);

// ramka na tekst
fill(255, 0, 0, 150);
stroke(255, 255, 255);
strokeWeight(10);
rect(320,20,360,660);

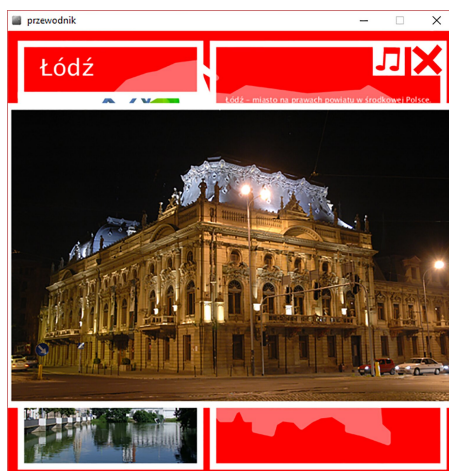
fill(255, 255, 255);
textSize(12);
text(opis, 340, 100, 340, 660);

image(ikona_zamknij,630,20,50,50);
image(ikona_muzyki,570,20,50,50);
```

```
1
2 }
```

Ekran foto i video

Dwa pozostałe ekrany odpowiadają za wyskakujące okienka z obrazkiem i materiałem video:



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Poniżej znajduje się kod dla każdego z okienek:

```
void ekran_foto(){  
  
    image(foto1b, width/2 - (foto1b.width)/2 , height/2 - (foto1b.height)/2 , foto1b.width, foto1b.height);  
    noFill();  
    stroke(255, 255, 255);  
    strokeWeight(10);  
    rect(width/2 - (foto1b.width)/2, height/2 - (foto1b.height)/2 , foto1b.width, foto1b.height);  
}  
  
void ekran_video(){  
  
    image(movie, width/2 - (movie.width)/2, height/2 - (movie.height)/2);  
    noFill();  
    stroke(255, 255, 255);  
    strokeWeight(10);  
    rect(width/2 - (movie.width)/2, height/2 - (movie.height)/2 , movie.width, movie.height);  
}
```

Funkcje odpowiedzialne za interakcje

Funkcje odpowiedzialne za interakcje pozwolą obsłużyć reakcje aplikacji na zachowania myszy. Pierwsza funkcja odpowiada za reakcję na kliknięcie przycisku myszy:

```
1  
2 void mousePressed(){  
3     // kliknięcie gdziekolwiek w obszarze ekranu powitalnego  
4     if(ekran == -1){  
5         ekran = 0;  
6     }  
7     // kliknięcie w ekranie menu głównego  
8     else if(ekran == 0){  
9         if(mouseX>330 && mouseX <400 && mouseY>300 && mouseY <370){  
10            ekran=1;  
11            miasto=1; // wybór Łodzi  
12        }  
13        else if(mouseX>430 && mouseX<500 && mouseY>250 && mouseY<320){  
14            ekran=1;  
15            miasto=2; // wybór Warszawy  
16        }  
17        else if(mouseX>360 && mouseX<430 && mouseY>480 && mouseY<550){  
18            ekran=1;  
19            miasto=3; // wybór Krakowa  
20        }  
21    }  
22 }
```

```

21     else if(mouseX>150 && mouseX<220 && mouseY>230 && mouseY<300){
22         ekran=1;
23         miasto=4; // wybór Poznań
24     }
25     else if(mouseX>240 && mouseX<310 && mouseY>50 && mouseY<120){
26         ekran=1;
27         miasto=5; // wybór Gdańska
28     }
29     else if(mouseX>150 && mouseX<220 && mouseY>370 && mouseY<440){
30         ekran=1;
31         miasto=6; // wybór Wrocławia
32     }
33     else if(mouseX > 630 && mouseX < 680 && mouseY > 20 && mouseY < 70){
34         exit(); // zamknięcie programu
35     }
36     else if(mouseX > 570 && mouseX < 620 && mouseY > 20 && mouseY < 70){
37         if(muzyka == 1){
38             player.pause(); // wstrzymanie muzyki
39             muzyka=0;
40         }
41         else{
42             player.play(); // wznowienie muzyki
43             muzyka=1;
44         }
45     }
46 }
47 // kliknięcie w ekranie miasta
48     else if(ekran == 1){
49         if(mouseX > 630 && mouseX < 680 && mouseY > 20 && mouseY < 70){
50             ekran=0; // powrót do menu
51         }
52         else if(mouseX > 570 && mouseX < 620 && mouseY > 20 && mouseY < 70){
53             if(muzyka == 1){
54                 player.pause(); // wstrzymanie muzyki
55                 muzyka=0;
56             }
57             else{
58                 player.play(); // wznowienie muzyki
59                 muzyka=1;
60             }
61         }
62         else if(mouseX > 20 && mouseX < 300 && mouseY > 300 && mouseY < 486){
63             ekran=2; // powiększenie zdjęcia
64             ikona_foto_over=0; // wyświetl miniaturkę zdjęcia
65         }
66         else if(mouseX > 20 && mouseX < 300 && mouseY > 494 && mouseY < 680){
67             ekran=3; // ekran video
68             ikona_video_over=0; // wyświetl miniaturkę zdjęcia
69             movie.play(); // odtworzenie video
70         }
71         else if(mouseX > 20 && mouseX < 300 && mouseY > 100 && mouseY < 680){
72             link(url);
73         }
74     }
75 // kliknięcie w ekranie foto
76     else if(ekran == 2){
77         ekran=1;
78     }
79 // kliknięcie w ekranie video
80     else if(ekran == 3){
81         movie.stop();
82         ekran=1;
83     }

```

W przypadku najechania myszą na obszar zdjęcia, wideo lub statystyk, pokazuje się ikona symbolizująca odpowiednią opcję :

```
1
2 void mouseMoved(){
3     //jeśli jesteśmy w ekranie miasta
4     if(ekran == 1){
5         // i najedziemy kursorem na zdjęcie
6         if(mouseX > 20 && mouseX < 300 && mouseY > 300 && mouseY < 486){
7             ikona_foto_over=1; // wyświetl ikonę foto zamiast miniaturki zdjęcia
8             ikona_video_over=0;
9         }
10        else if(mouseX > 20 && mouseX < 300 && mouseY > 494 && mouseY < 680){
11            ikona_video_over=1; // wyświetl ikonę video zamiast miniaturki zdjęcia
12            ikona_foto_over=0;
13        }
14        else{
15            ikona_foto_over=0; // wyświetl miniaturkę zdjęcia
16            ikona_video_over=0;
17        }
18    }
19 }
20 void movieEvent(Movie m) {
21     m.read();
22 }
```

Statystyki będą otwierały się w postaci strony internetowej wygenerowanej w rozdziale Bazy danych.

Analiza danych aplikacji (Bazy danych)

Wstęp

Już wiesz

Aby zrealizować ten etap projektu przypomnijcie sobie informacje z gimnazjum dotyczące:

- budowanie i formatowanie tabel
- adresowanie względne, bezwzględne i mieszane
- budowanie formuł
- stosowanie funkcji w formułach (SUMA, ŚREDNIA, JEŻELI, MAX, MIN, LICZ JEŻELI, SUMA JEŻELI)
- budowanie formuł zagnieżdżonych
- formatowanie warunkowe
- tworzenie i modyfikowanie wykresów
- budowanie tabel i wykresów przestawnych
- sortowanie
- filtrowanie

Zrealizowany projekt aplikacji zawiera zbiór danych i zależności pomiędzy nimi. W tym rozdziale nauczycie się w jaki sposób gromadzić, wizualizować i przetwarzać zgromadzone dane za pomocą arkusza kalkulacyjnego oraz projektować proste relacyjne bazy danych.

Zebranie danych statystycznych projektu w arkuszu kalkulacyjnym

Należy się zastanowić, jakie dane związane z projektowaną aplikacją można wyszczególnić. Przykładowo, dla realizowanej gry komputerowej mogą to być rezultaty rozgrywki (np. punktacja i czas). W przypadku multimedialnego przewodnika geograficznego możemy zestawić dane statystyczne dotyczące konkretnego regionu, zaś dla symulatora lotów kosmicznych będą to, na przykład, trajektorie lotów.

Zebrane dane można następnie zapisać w arkuszu kalkulacyjnym, aby przeprowadzić ich statystyczne podsumowanie.

Graficzna prezentacja danych

Zrealizowana aplikacja wiąże ze sobą dużą liczbę danych. Graficzna prezentacja danych w postaci wykresów ułatwia ich zrozumienie i interpretację. W zależności od kontekstu danych, należy umiejętnie dobrać odpowiedni typ wykresu.

Projektowanie relacyjnej bazy danych

Projekt przewodnika multimedialnego zawiera bardzo dużo informacji związanych z opisem ciekawych miejsc w postaci tekstu, danych statystycznych, zdjęć, filmów i dźwięku. Zapisanie tych informacji w postaci bazy danych w arkuszu kalkulacyjnym lub w bazie danych umożliwi ich uporządkowanie oraz szybkie wyszukiwanie potrzebnych danych, również z wykorzystaniem zapytań w języku SQL.

Realizacja tej części projektu będzie wymagała sięgnięcia do informacji z gimnazjum w następującym zakresie:

- tworzenie bazy danych,
- wprowadzanie i edycja rekordów z wykorzystaniem języka SQL,
- wyszukiwanie informacji w bazie danych,
- sortowanie danych według wybranych pól.

W bazie dane są przechowywane w tabelach w postaci rekordów, gdzie każdy rekord składa się z pól o ściśle określonym typie danych.

W tej części e-podręcznika korzystając z umiejętności nabytych w gimnazjum, należy utworzyć bazę danych składającą się z jednej tabeli, a następnie przekształcić ją do modelu relacyjnego.

Baza danych wykorzystywana w projekcie przewodnika multimedialnego będzie zawierała dane statystyczne dotyczące wybranych miast w Polsce.

Materiałem źródłowym, z którego pobierane są informacje jest arkusz kalkulacyjny dostępny poniżej:



Zasób interaktywny dostępny pod adresem <https://zpe.gov.pl/a/DECUGF4hr>

Dane statystyczne zostały pobrane ze stron Głównego Urzędu Statystycznego . Portal ten jest ogromnym źródłem wiedzy i informacji na temat ludności, gospodarki, regionów administracyjnych i innych dziedzin życia publicznego. Stanowi on ogromną bazę, z której można pobierać i wykorzystywać dane.

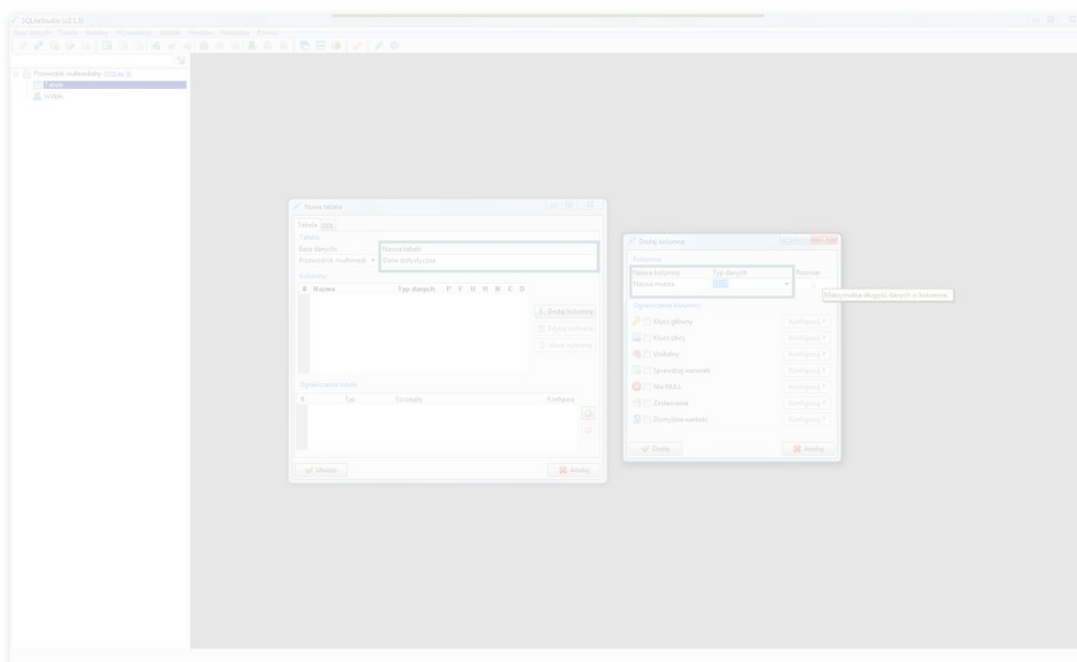
Można również utworzyć bazę danych zawierającą inne informacje.

Tworzona baza danych będzie zawierała tabelę o nazwie *Dane statystyczne* umożliwiającą przechowywanie informacji statystycznych wykorzystywanych w projekcie. Tabela ta może zawierać następujące pola

- nazwa miasta,
- województwo,
- liczba mieszkańców,
- liczba kobiet,
- liczba mężczyzn,
- grupa wiekowa,
- powierzchnia miasta,
- powierzchnia terenów zielonych,
- data aktualizacji danych.

Wprowadzone w bazie danych nazwy pól nie mogą zawierać spacji, a ze względu na stosowane implementacje języka zapytań SQL, należy zrezygnować z polskich znaków diakrytycznych. Czyli zamiast nazwy pola Liczba mieszkańców, w bazie danych powinna być nazwa *Liczba_mieszkancow*, w której zamiast spacji będzie wstawiony znak podkreślenia i nie będą stosowane polskie znaki diakrytyczne.

Proces tworzenia bazy można prześledzić w przedstawionym pokazie slajdów



Źródło: Maciej Kacperski <maciej.kacperski@p.lodz.pl>, licencja: CC BY 3.0.

Przy wprowadzeniu danych do tak przygotowanej tabeli, wprowadzając dane dla różnych grup wiekowych oraz aktualizując dane dotyczące liczby mieszkańców w danym roku, należy wprowadzić również wszystkie pozostałe dane do pozostałych pól tabeli.

Zasób interaktywny dostępny pod adresem <https://zpe.gov.pl/a/DECUGF4hr>

Źródło: Maciej Kacperski <maciej.kacperski@p.lodz.pl>, licencja: CC BY 3.0.

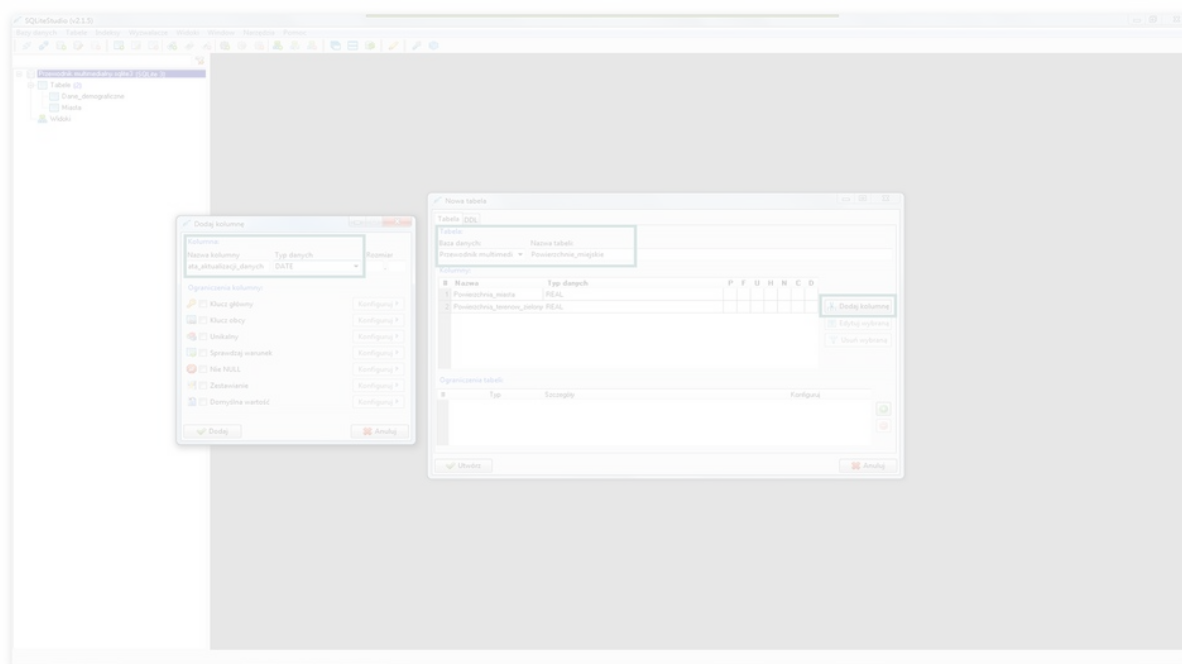
We wpisanych do tabeli rekordach powtarzają się informacje o nazwie miasta, województwie, powierzchni miasta, powierzchni terenów zielonych, a więc tak zbudowana tabela zawiera informacje nadmiarowe.

W relacyjnych bazach danych tworzy się nieco bardziej skomplikowaną strukturę zapisu danych, która składa się z wielu tabel wzajemnie ze sobą powiązanych. Tworzenie relacyjnej bazy danych wymaga wyodrębnienia pól, które utworzą nowe tabele, dzięki czemu ograniczona zostanie nadmiarowość zapisu danych.

Relacyjna baza danych

Relacyjne bazy danych są obecnie najpowszechniej wykorzystywanym systemem baz danych. Baza danych w tym modelu zawiera wiele tabel, w których gromadzone są informacje. Tabele (nazywane inaczej relacjami) mają zawarte w swojej strukturze pól logiczne powiązania pomiędzy danymi znajdującymi się w różnych tabelach. Przykładem realizacji takiego

powiązania może być baza danych **Przewodnik multimedialny** zawierająca informacje statystyczne o wybranych miastach w Polsce. W tabeli *Dane statystyczne*, w polach *Nazwa miasta* i *Województwo* wpisane wartości wielokrotnie się powtarzają przy wprowadzaniu kolejnych rekordów. Ponadto w tabeli *Dane statystyczne* można wyodrębnić trzy grupy pól logicznie ze sobą powiązane. Pierwsza grupa dotyczy opisu miasta, czyli *Nazwa miasta* i jego położenie, czyli *Województwo*. Druga grupa to dane dotyczące ludności, czyli pola *Liczba mieszkańców*, *Liczba kobiet*, *Liczba mężczyzn*, *Grupa wiekowa*. Ostatnia grupa, to dane dotyczące powierzchni miejskich, czyli pola *Powierzchnia miasta* i *Powierzchnia terenów zielonych*. Ponadto, dane statystyczne pobrane z Głównego Urzędu Statystycznego, podawane są na określony dzień, miesiąc i rok. Projektowana baza danych może gromadzić informacje z wielu lat, więc należy dodać pole Data aktualizacji danych w grupach dotyczących informacji demograficznych i powierzchni miejskich. Przedstawiony podział na tak określone grupy danych pozwala uniknąć powtarzania danych. Zamiast tabeli *Dane statystyczne* należy utworzyć trzy tabele i powiązać rekordy między tymi tabelami tak, aby można było otrzymać pełną informację.



Zasób interaktywny dostępny pod adresem <https://zpe.gov.pl/a/DECUGF4hr>

Źródło: Maciej Kacperski <maciej.kacperski@p.lodz.pl>, licencja: CC BY 3.0.

Nowe tabele będą miały następującą strukturę:

tabela **Miasta** zawierać będzie dwa pola

- Nazwa miasta,
- Województwo,

tabela **Dane demograficzne** będzie składała się z następujących pól:

- Liczba mieszkańców,

- Liczba kobiet,
- Liczba mężczyzn,
- Grupa wiekowa,
- Data aktualizacji danych.

tabela **Powierzchnie miejskie** będzie miała pola:

- Powierzchnia miasta,
- Powierzchnia terenów zielonych,
- Data aktualizacji danych.

Projektując bazę danych należy unikać zapisywania danych, które można obliczyć na podstawie danych zapisanych w tym samym rekordzie. W przypadku tabeli *Dane demograficzne*, liczbę mężczyzn można wyznaczyć z liczby mieszkańców i liczby kobiet, dlatego zapisywanie tej informacji jest zbędne.

Utworzone trzy tabele należy wzajemnie powiązać tak, aby jednoznacznie określić, które rekordy z tabeli *Powierzchnie miejskie* i *Dane demograficzne* są powiązane z określonym miastem w tabeli *Miasta*. Stosuje się do tego dodatkowe pola dodawane do wszystkich tabel. W tabeli *Miasta* dodajemy pole *ID_miasta* będące kluczem głównym tej tabeli, jednoznacznie identyfikującym każdy jej rekord. Cechą charakterystyczną klucza głównego jest jego unikalność w obrębie tabeli. W tabeli *Dane demograficzne* i *Powierzchnie miejskie* dodawane są pola o zwykle takiej samej nazwie, czyli *ID_miasta*, będące w tych tabelach kluczem obcym, umożliwiającym właściwe powiązanie rekordów między tabelami. Ostateczna postać struktury bazy danych **Przewodnik multimedialny** jest określona poprzez trzy tabele *Miasta*, *Dane demograficzne* oraz *Powierzchnie miejskie*, w których zastosowano powiązanie za pomocą pola *ID_miasta*. Struktura pól w tabelach, po dokonanych modyfikacjach będzie następująca:

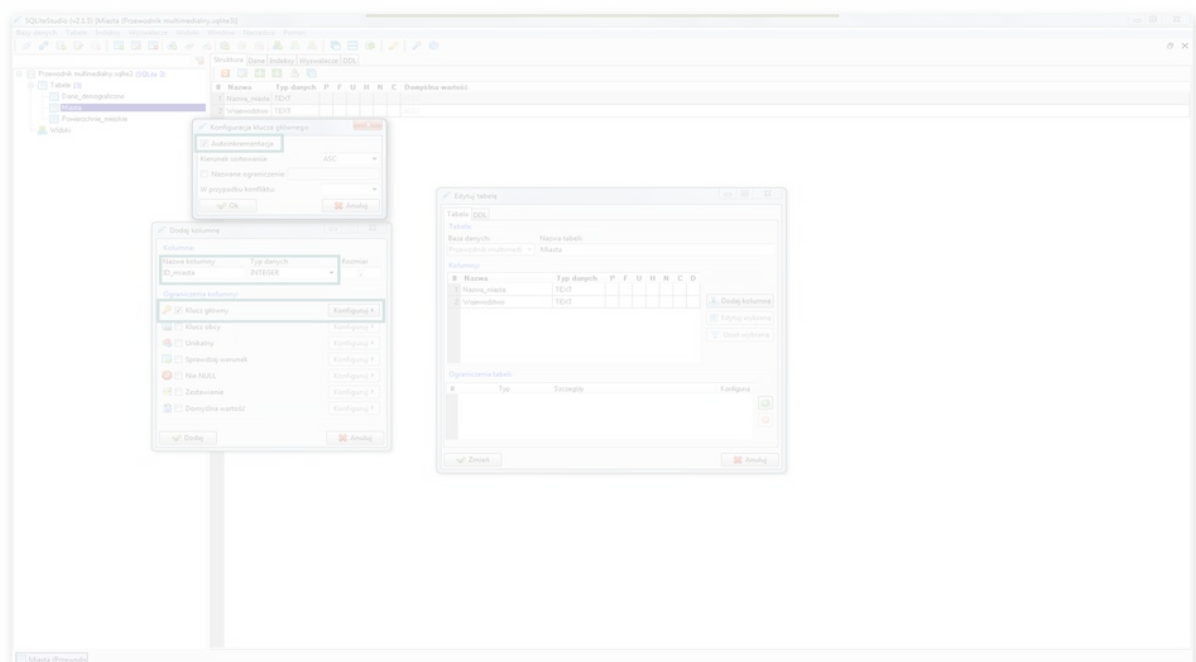


Tabela **Miasta** zawierać będzie trzy pola

- ID_miasta (klucz główny),
- Nazwa miasta,
- Województwo.

Tabela **Dane demograficzne** składać się będzie z następujących pól

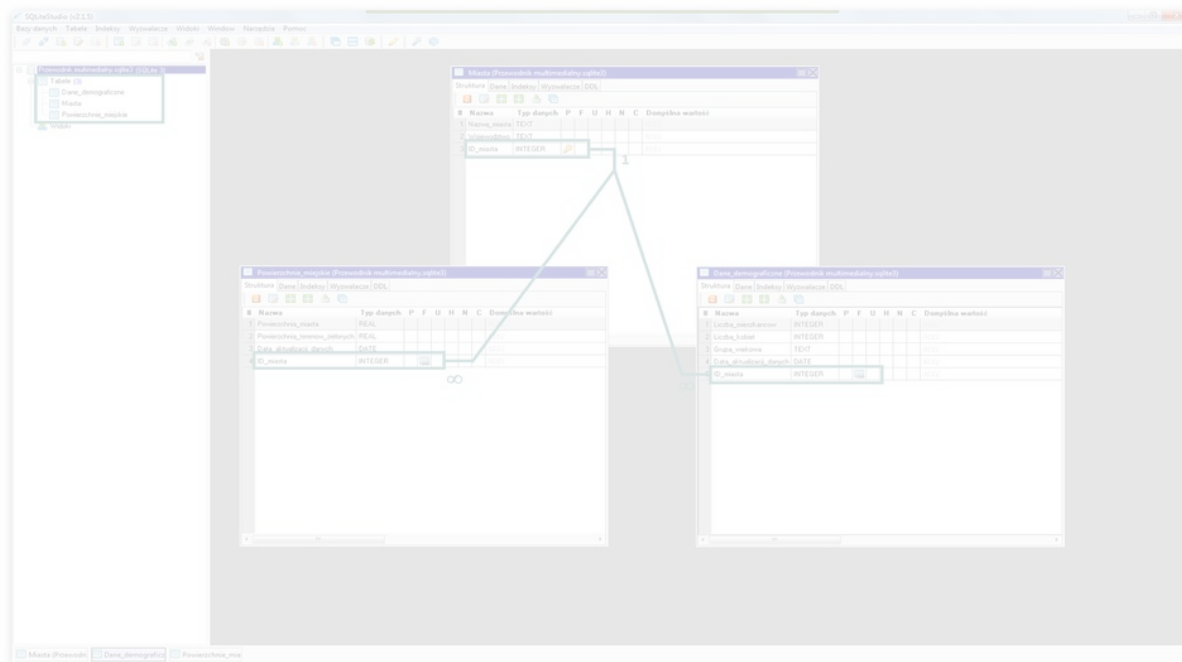
- ID_miasta (klucz obcy),
- Liczba mieszkańców,
- liczba kobiet,
- Grupa wiekowa,
- Data aktualizacji danych.

Tabela **Powierzchnie miejskie** składać się będzie z następujących pól

- ID_miasta (klucz obcy),
- Powierzchnia miasta,
- Powierzchnia terenów zielonych,
- Data aktualizacji danych.

Zdefiniowany typ danych dla pola *ID_miasta* to typ numeryczny –najczęściej jest to liczba całkowita. Wartości w kolumnie pola klucza głównego muszą być niepowtarzalne (unikalne), jednoznacznie identyfikujące każdy rekord. Jedną z metod osiągnięcia niepowtarzalności jest *autoinkrementacja*, polegająca na nadawaniu kolejnym dodawanym do tabeli rekordom wartości kolejnych liczb całkowitych. Istnieją powszechnie stosowane dane, które mogą pełnić rolę klucza głównego. W przypadku danych osobowych jest to najczęściej numer PESEL, numer identyfikacji podatkowej - NIP, numer indeksu studenta lub legitymacji szkolnej.

W tabelach *Dane demograficzne* i *Powierzchnie miejskie* dodane zostało pole *ID_miasta* określane jako klucz obcy. Pole to służy do jednoznacznego powiązania rekordów z tabeli *Miasta* z rekordami w tabelach *Dane demograficzne* i *Powierzchnie miejskie*. W polu *ID_miasta* w tabelach *Dane demograficzne* i *Powierzchnie miejskie* mogą być zapisywane wyłącznie wartości występujące w polu *ID_miasta* tabeli *Miasta*.



Zasób interaktywny dostępny pod adresem <https://zpe.gov.pl/a/DECUGF4hr>

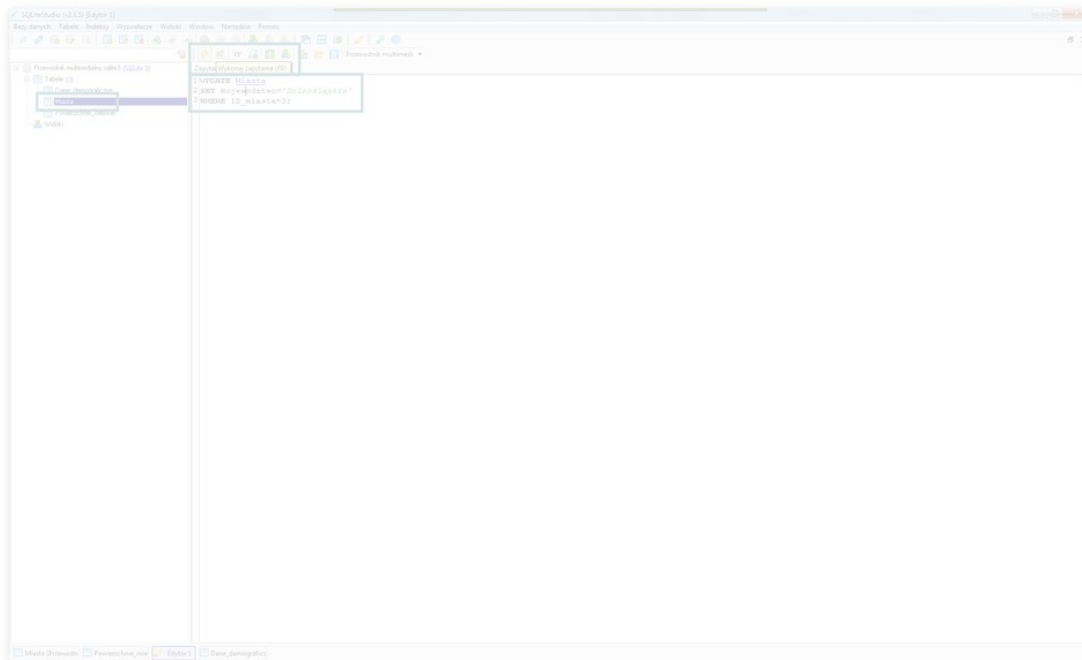
Źródło: Maciej Kacperski <maciej.kacperski@p.lodz.pl>, licencja: CC BY 3.0.

Dodawanie rekordów

Sposób wprowadzania rekordów do tabeli omawiany był w e-podręczniku dla gimnazjum. Rekordy można dodawać w trybie graficznym, przy wykorzystaniu widoku tabeli lub widoku formularza. Alternatywną metodą jest dodawanie rekordów z użyciem polecenia INSERT języka SQL.

Przy wprowadzaniu danych należy pamiętać o właściwym powiązaniu rekordów pomiędzy tabelą nadrzędną *Miasta* oraz tabelami podrzędnymi *Dane demograficzne* i *Powierzchnie miejskie*. Kolejne wprowadzone rekordy do tabeli *Miasta* mają wartość pola *ID_miasta* zwiększaną (inkrementowaną) o jeden. Ponadto, w tabeli *Miasta* wpisanych jest tylko sześć rekordów, natomiast w tabeli *Dane statystyczne* wpisanych rekordów było wielokrotnie więcej. Wartości występujące w polu *ID_miasta* są unikalne, nie możemy więc dodać nowego rekordu o już istniejącej w tym polu wartości. Po wprowadzeniu danych do tabeli *Dane demograficzne*, w polu klucza obcego *ID_miasta* będą występowały wyłącznie wartości odpowiadające wartościom z pola *ID_miasta* w tabeli *Miasta*.

Zrozumienie sposobu wprowadzania rekordów w bazie relacyjnej ułatwi pokaz slajdów zamieszczony poniżej



Zasób interaktywny dostępny pod adresem <https://zpe.gov.pl/a/DECUGF4hr>

Źródło: Maciej Kacperski <maciej.kacperski@p.lodz.pl>, licencja: CC BY 3.0.

Wyszukiwanie danych

Wyszukiwanie informacji w relacyjnej bazie danych jest realizowane w nieco odmienny sposób niż w bazie zawierającej jedną tabelę. Dane są zapisywane w wielu tabelach powiązanych ze sobą za pomocą kluczy głównych i kluczy obcych. W zależności od tego, czy wyszukiwane informacje znajdują się w jednej tabeli, czy w wielu tabelach, stosuje się różną składnię zapytań w języku SQL.

Sposób konstruowania zapytań umożliwiający wyszukanie informacji w jednej tabeli był omawiany w gimnazjum w rozdziale dotyczącym baz danych i wyszukiwania informacji w bazie danych. Wyszukiwanie informacji z dwóch tabel wymaga wprowadzenia do zapytania, w klauzuli *WHERE*, kryteriów dotyczących klucza głównego i klucza obcego. Wprowadzone kryteria wyszukiwania muszą powiązać rekordy z tabeli podrzędnej z rekordem z tabeli nadrzędnej.

Oto przykładowe zadanie, które będzie polegało na wyszukaniu informacji o danych demograficznych i powierzchniach miejskich dla Krakowa. Nazwa miasta jest zapisana w tabeli *Miasta*. Dane demograficzne są zapisane w tabeli *Dane demograficzne*. Dane są powiązane między tabelami poprzez pole *ID_miasta*, które w tabeli *Miasta* jest kluczem głównym, a w tabeli *Dane demograficzne* jest kluczem obcym. Wyszukanie w tabeli *Miasta* danych dla Krakowa można zrealizować za pomocą zapytania, które będzie miało w klauzuli

WHERE kryterium wykorzystujące nazwę miasta lub identyfikator zapisany w polu *ID_miasta*.

Zapytanie w języku SQL będzie wyglądało następująco:

```
SELECT Nazwa_miasta,
```

```
Wojewodztwo,
```

```
ID_miasta
```

```
FROM Miasta
```

```
WHERE Nazwa_miasta = 'Kraków';
```

lub z wykorzystaniem pola *ID_miasta*

```
SELECT Nazwa_miasta,
```

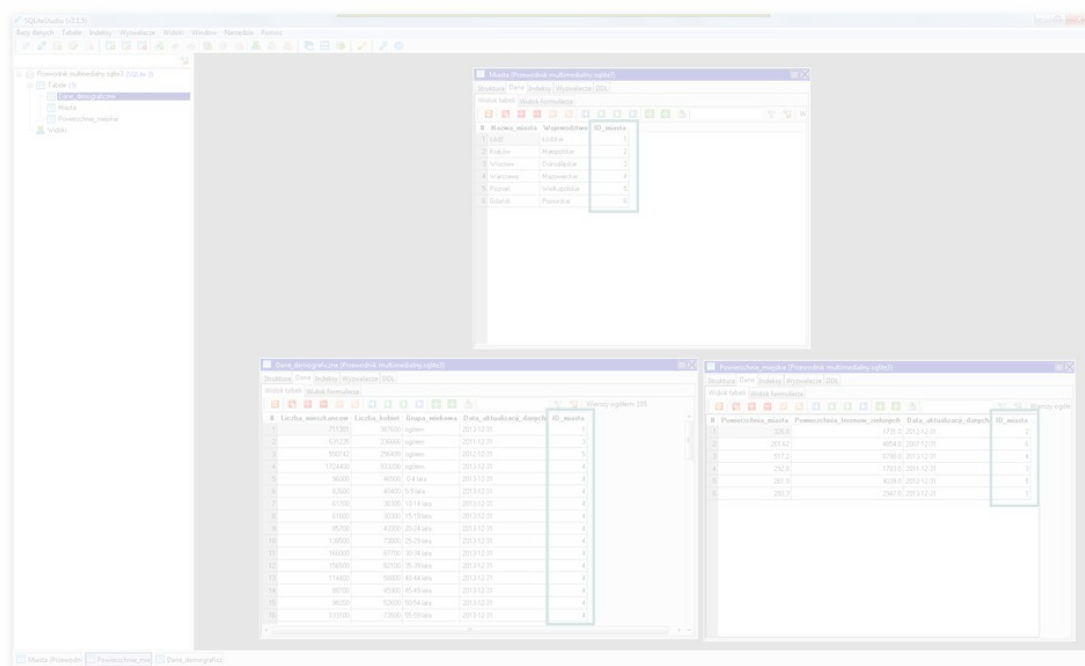
```
Wojewodztwo,
```

```
ID_miasta
```

```
FROM Miasta
```

```
WHERE ID_miasta = 2;
```

Na zamieszczonym niżej pokazie slajdów przedstawiony został proces tworzenia zapytań w relacyjnej bazie danych



Zasób interaktywny dostępny pod adresem <https://zpe.gov.pl/a/DECUGF4hr>

Źródło: Maciej Kacperski <maciej.kacperski@p.lodz.pl>, licencja: CC BY 3.0.

Zapytanie wyszukujące rekordy z tabeli *Dane demograficzne* musi wykorzystywać wartości zapisane w kluczu obcym tej tabeli *ID_miasta* i kod polecenia w języku SQL będzie następujący:

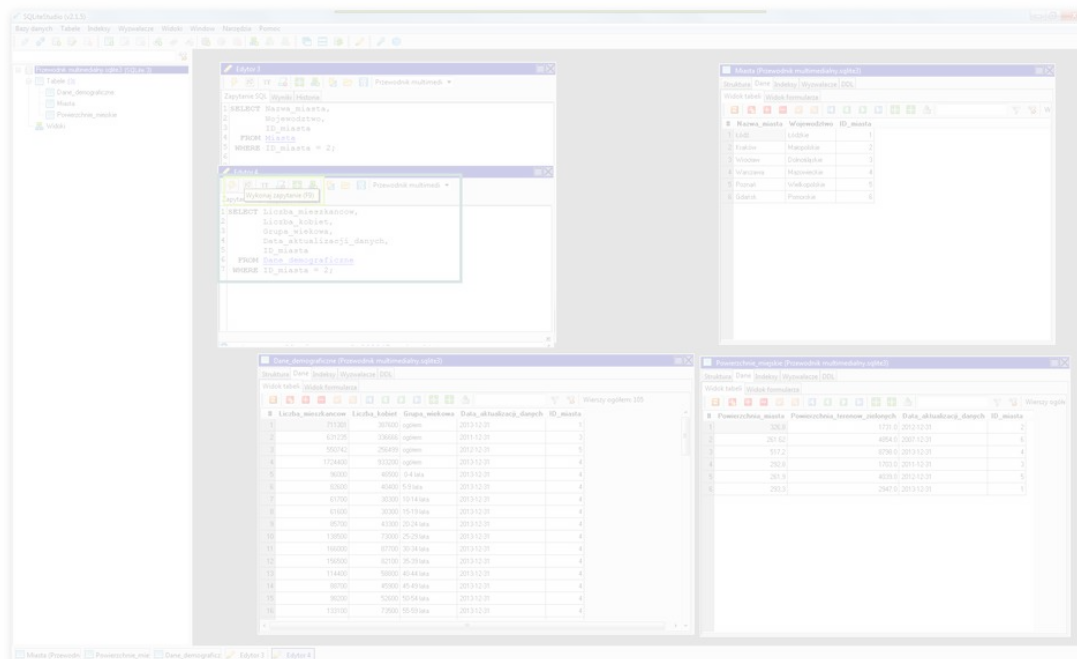
```
SELECT Liczba_mieszkancow,
```

```
Liczba_kobiet,
```

```
Grupa_wiekowa,
```

```
Data_aktualizacji_danych,
```

```
ID_miasta
FROM Dane_demograficzne
WHERE ID_miasta = 2;
```



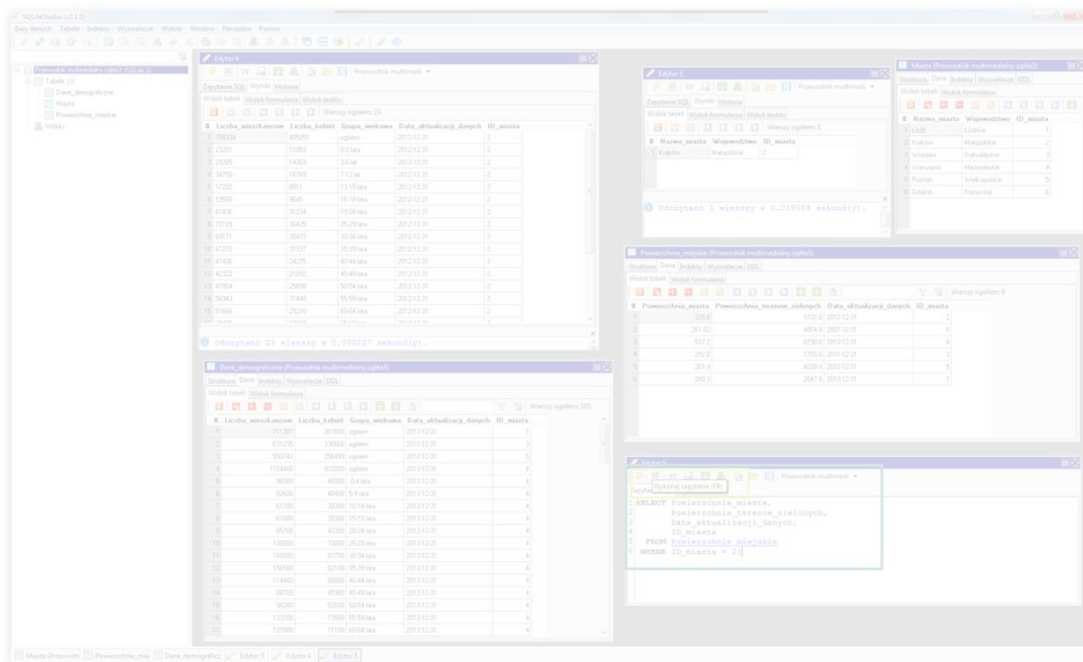
Zasób interaktywny dostępny pod adresem <https://zpe.gov.pl/a/DECUGF4hr>

Źródło: Maciej Kacperski <maciej.kacperski@p.lodz.pl>, licencja: CC BY 3.0.

Zapytanie wyszukujące rekordy z tabeli *Powierzchnie miejskie* musi wykorzystywać wartości zapisane w kluczu obcym tej tabeli czyli polu *ID_miasta*. Kod polecenia w języku SQL będzie następujący.

```
SELECT Powierzchnia_miasta,
       Powierzchnia_terenow_zielonych,
       Data_aktualizacji_danych,
       ID_miasta
FROM Powierzchnie_miejskie
WHERE ID_miasta = 2;
```

Realizacja przedstawionych zapytań SQL dla trzech tabel bazy **Przewodnik multimedialny** zwróci wyniki w postaci wyszukanych rekordów w trzech oddzielnych widokach. Tak jak zostało to przedstawione w poniższym pokazie slajdów.



Zasób interaktywny dostępny pod adresem <https://zpe.gov.pl/a/DECUGF4hr>

Źródło: Maciej Kacperski <maciej.kacperski@p.lodz.pl>, licencja: CC BY 3.0.

Wykorzystanie tak uzyskanych danych nie jest wygodne, zmusza do ciągłej kontroli pól klucza *ID_miasta* w celu połączenia danych z różnych tabel. Znacznie lepszym rozwiązaniem jest otrzymanie wyniku zapytania SQL w jednym widoku. Język SQL oferuje możliwość konstruowania zapytań, w których można odwołać się do pól występujących w różnych tabelach. Składnia polecenia języka SQL musi uwzględniać nazwę tabeli przy wprowadzaniu w treści zapytania nazwy pola tej tabeli. Składnia instrukcji jest następująca:

```
SELECT table_name.column1, table_name.column2, ... table_name.columnN
FROM table_name
WHERE [condition]
```

W instrukcji występują już znane elementy określające nazwę tabeli oraz warunki wyszukiwania. Inny jest zapis nazwy pola, który poprzedzony jest nazwą tabeli i separatorem w postaci kropki. Ten sposób zapisu jest powszechnie wykorzystywaniu w adresach internetowych.

Postać zapytania umożliwiającego wybór Krakowa z tabeli miasta będzie miał następującą postać:

```
SELECT Miasta.Nazwa_miasta,
       Miasta.Wojewodztwo,
       Miasta.ID_miasta
FROM Miasta
WHERE Miasta.ID_miasta = 2;
```

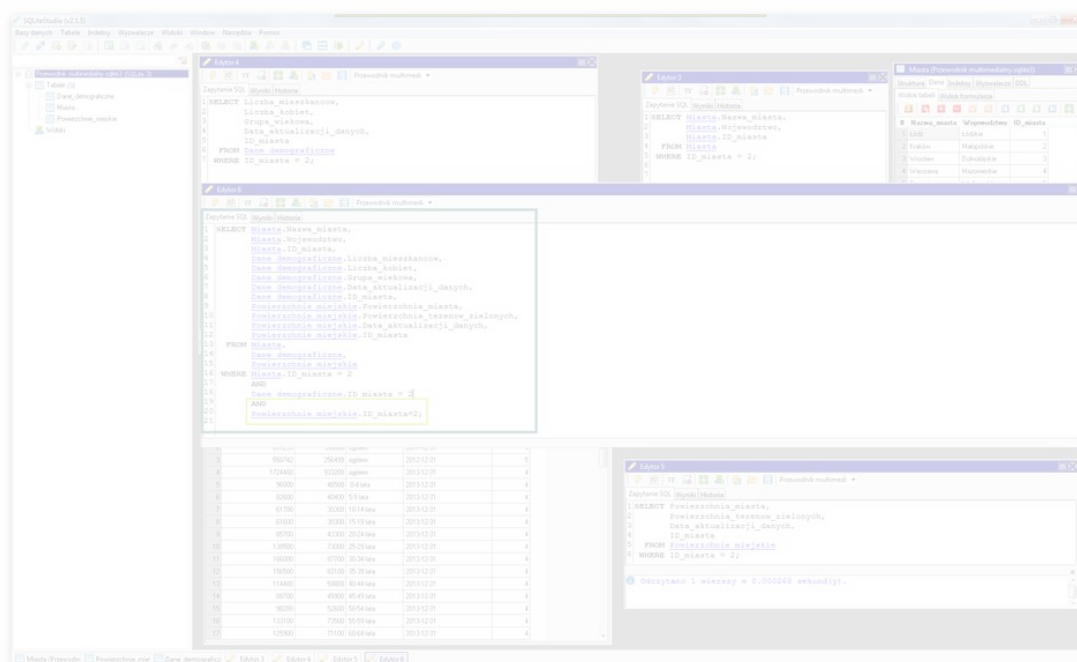
W zapytaniu, które będzie pobierało dane z trzech tabel, należy uwzględnić nazwy tabel w klauzuli *FROM*. Trzeba wymienić nazwy wszystkich tabel, z których są wybierane pola, rozdzielając je przecinkiem. W warunku, który występuje w klauzuli *WHERE*, należy

uwzględnić wszystkie nazwy pól występujące w poprzednich zapytaniach i połączyć je w jedno zdanie logiczne z operatorem *AND*.

Przykład zapytania SQL przedstawiony jest poniżej

```
SELECT Miasta.Nazwa_miasta,  
Miasta.Wojewodztwo,  
Miasta.ID_miasta,  
Dane_demograficzne.Liczba_mieszkanow,  
Dane_demograficzne.Liczba_kobiet,  
Dane_demograficzne.Grupa_wiekowa,  
Dane_demograficzne.Data_aktualizacji_danych,  
Dane_demograficzne.ID_miasta,  
Powierzchnie_miejskie.Powierzchnia_miasta,  
Powierzchnie_miejskie.Powierzchnia_terenow_zielonych,  
Powierzchnie_miejskie.Data_aktualizacji_danych,  
Powierzchnie_miejskie.ID_miasta  
FROM Miasta,  
Dane_demograficzne,  
Powierzchnie_miejskie  
WHERE Miasta.ID_miasta = 2  
AND  
Dane_demograficzne.ID_miasta = 2  
AND  
Powierzchnie_miejskie.ID_miasta = 2;
```

Wynikiem działania takiego zapytania jest widok zawierający scalone dane z trzech tabel. Proces konstruowania zapytania SQL został przedstawiony poniżej:



Zasób interaktywny dostępny pod adresem <https://zpe.gov.pl/a/DECUGF4hr>

W wyniku zapytania wyświetlane są również pola klucza głównego i pola kluczy obcych *Miasta.ID_miasta*, *Dane_demograficzne.ID_miasta*, *Powierzchnie_miejskie.ID_miasta* pola te w tym widoku nie są przydatne dla użytkownika, ponieważ informacja z trzech tabel została już scalona. Można usunąć te pola z klauzuli *SELECT*. Ostateczna postać zapytania będzie następująca:

```
SELECT Miasta.Nazwa_miasta,  
       Miasta.Wojewodztwo,  
       Miasta.ID_miasta,  
       Dane_demograficzne.Liczba_mieszkancow,  
       Dane_demograficzne.Liczba_kobiet,  
       Dane_demograficzne.Grupa_wiekowa,  
       Dane_demograficzne.Data_aktualizacji_danych,  
       Dane_demograficzne.ID_miasta,  
       Powierzchnie_miejskie.Powierzchnia_miasta,  
       Powierzchnie_miejskie.Powierzchnia_terenow_zielonych,  
       Powierzchnie_miejskie.Data_aktualizacji_danych,  
       Powierzchnie_miejskie.ID_miasta  
FROM Miasta,  
     Dane_demograficzne,  
     Powierzchnie_miejskie  
WHERE Miasta.ID_miasta = 2  
AND  
     Dane_demograficzne.ID_miasta = 2  
AND  
     Powierzchnie_miejskie.ID_miasta = 2
```

Strona internetowa projektu (Internet)

Wstęp

Projekt jest już niemalże skończony. Ostatnim etapem będzie opublikowanie efektów pracy w Internecie. W tym celu należy przygotować prostą stronę internetową prezentującą wykonaną aplikację. Projekt strony internetowej powinien być spójny z wyglądem aplikacji. Potrzebny będzie również dostęp do konta pocztowego, aby zapewnić możliwość komunikacji z użytkownikami stworzonej aplikacji.

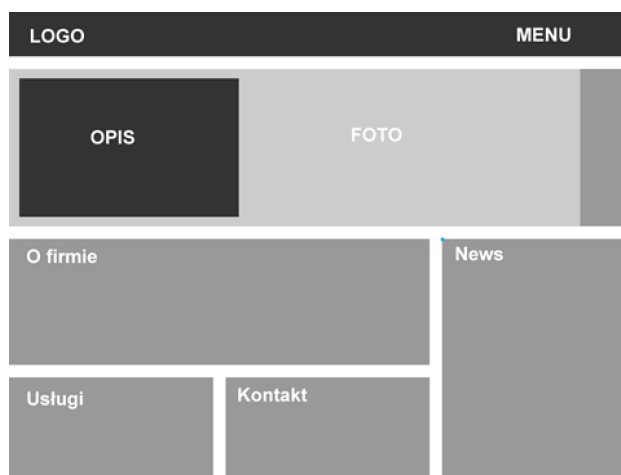
Wybór technologii tworzenia serwisów WWW

Należy się zastanowić, jakie informacje i dane dotyczące realizowanego projektu powinno się umieścić na stronie internetowej. Wybór technologii, w jakiej będzie tworzona strona, zależy od rodzaju treści i sposobu ich prezentacji. Najprostszą możliwością jest zapisanie utworzonej wcześniej prezentacji multimedialnej w formacie html.

W ramach realizowanego projektu można utworzyć stronę promującą opracowaną aplikację w technologii HTML5 z zastosowaniem kaskadowych arkuszy stylów. Przed przejściem do implementacji, należy zaplanować układ strony.

Projekt strony

Układ poszczególnych elementów (tzw. layout) strony internetowej w dużej mierze zależy od funkcji jaką będzie pełniła projektowana strona. Jeśli ma ona zaprezentować zrzuty ekranu ze stworzonej gry komputerowej lub fotografie z multimedialnego przewodnika geograficznego, konieczne będzie zaplanowanie dostatecznej przestrzeni lub nawet osobnej podstrony na galerię. W przypadku symulatora lotów kosmicznych najważniejsze mogą być zebrane dane statystyczne osadzone na stronie w postaci tabelki.



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Tworzenie strony z wykorzystaniem: HTML 5 i CSS3

Stronę internetową można napisać wykorzystując zwykły notatnik. Edytory tekstu mają możliwość zapisu pliku w formacie html. Dzięki tej opcji można zaprojektować stronę internetową bez znajomości html, jednak bardzo pomocne jest wykorzystanie podstawowych poleceń i tagów języka html.

W Internecie można znaleźć wiele darmowych edytorów html do projektowania stron WWW. Bardzo pomocną opcją jest możliwość kolorowania składni oraz sprawdzanie poprawności kodu. Rozwiązania te pomagają sprawnie edytować tworzoną stronę. Tworząc stronę w języku polskim, ważne jest by edytor miał możliwość kodowania polskich znaków w standardzie ISO i UTF. Znaczniki html, w większości składają się ze znaczników otwierających i zamykających, dlatego bardzo pomocną opcją jest ich domykanie. W sieci można znaleźć także edytory wizualne oraz kreatory szkieletu strony, które w sposób automatyczny są w stanie wygenerować szkielet

projektowanej witryny

W tabeli zestawiono nazwy popularnych edytorów dla różnych systemów operacyjnych

Nazwa	System operacyjny
Adobe Dreamweaver	Windows
Ager Web Edytor	Windows
Bluefish	Windows, Linux, Mac OS X
CoreEditor	Windows
EdHTML	Windows
HateML Pro 2	Windows
kED	Windows
KompoZer	Windows
Microsoft Expression Web	Windows
Pajączek	Windows
nvu	Windows
PSPad	Windows
Web Edit	Windows
Notepad++	Windows
NetBeans	Windows
Kate	Linux
Quanta Plus	Linux
gedit	Linux
Geany	Linux
Smultron	Mac OS X
Taco HTML Edit	Mac OS X
Fraise	Mac OS X

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Jednym z polecanych edytorów jest Bluefish, gdyż stanowi wolne oprogramowanie. Dzięki wieloplatformowości, pozwoli na sprawną naukę edytora przez użytkowników różnych systemów operacyjnych. Program posiada szereg przydatnych funkcji, takich jak kolorowanie składni HTML, PHP, C, Javy, JavaScriptu, JSP, SQL, XML, Pythona, Perla, CSS, ColdFusion, Pascala, R i Octave/Matlab, automatyczne zamykanie znaczników HTML i XML. Ma wbudowane funkcje kreatora stron (np. automatyczne generowanie galerii obrazów). Adres z którego można go pobrać:

<http://www.bennewitz.com/bluefish/stable/source/>

Aby móc obejrzeć zaprojektowaną stronę WWW należy ją otworzyć w przeglądarce. Przeglądarki internetowe do niedawna komunikowały się z serwerem za pomocą protokołu HTTP, choć obecnie najpopularniejszym protokołem staje się HTTPS.

Najpopularniejsze przeglądarki na rynku światowym (dane z grudnia 2015 według serwisu www.w3schools.co) to:

- Google Chrome – 66,5%
- Internet Explorer – 6,9%
- Mozilla Firefox – 20%
- Safari – 3,6%
- Opera – 1,4%

Większość przeglądarek (w najnowszych wersjach) wspiera HTML5 i CSS3.

HTML to język do oznaczania składni (markup), który pozwala klasyfikować zawartość dokumentu z rozszerzeniem .html lub .htm.

Pojedynczy element HTML nosi nazwę Taga lub znacznika. Każdy znacznik może zawierać skojarzone z nim atrybuty.

Konstrukcja HTML5 różni nieco od konstrukcji strony w HTML czy XHTML. Każdy dokument będzie się zaczynał od wpisu określającego rodzaj języka. Jest to informacja dla przeglądarki w jaki sposób interpretować występujące na stronie znaczniki. Dla HTML5 znacznik otwierający to

```
<!DOCTYPE html>.
```

Następnym elementem wspólnym dla wszystkich języków html jest znacznik

```
<html></html>.
```

Wewnątrz tego znacznika tworzona będzie treść projektowanego dokumentu czyli właściwa zawartość strony WWW.

W strukturze strony można wyróżnić dwie zasadnicze części. Części te zdefiniowane są odpowiednio w znacznikach `<head></head>` oraz `<body></body>`. Pierwszy znacznik odpowiada za określenie podstawowych właściwości strony (tytuł, strona kodowa, informacje o autorze, słowa kluczowe itp.).

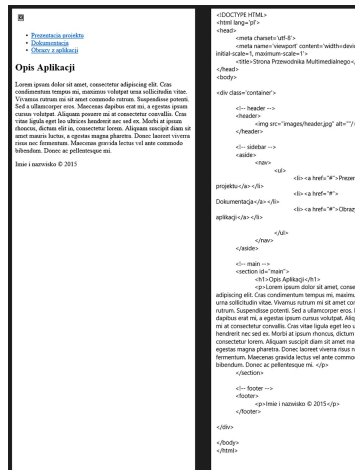
Druga część zamknięta w znacznikach `<body></body>`, jest to ciało strony czyli to co zostanie wyświetlone w oknie przeglądarki.

Podstawowy szkielet strony HTML5 jest następujący:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8" />
5 <title>To jest przykładowa strona</title>
6 </head>
7 <body>
8 A tutaj treść strony
9 </body>
10 </html>
```

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Poniżej znajduje się przykładowy szkielet ze zdefiniowanym układem strony i podstawową treścią.



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Na projektowanej stronie internetowej powinny znaleźć się informacje na temat stworzonej aplikacji. Na stronie głównej można umieścić logo projektu i podstawowe informacje na temat funkcjonalności i charakterystyki autorskiej aplikacji. W menu nawigacyjnym powinny znajdować się odnośniki do podstron zawierających dokumentację projektu oraz podstron zawierających prezentację zrzutów ekranowych z aplikacji.

Podstrony, z którymi będą się łączyć hiperłącza projektowanej strony wykonane zostaną z dokumentów powstałych podczas realizacji projektu, to znaczy:

- prezentacja będzie wcześniej stworzoną prezentacją zapisaną w formacie html,
- dokumentacja będzie linkiem do pliku dokumentacji zapisanego w formacie pdf,
- obrazy to podstrona stworzona w html, zawierająca zdjęcia z aplikacji.

Szkielet strony jest już gotowy, więc można zająć się jej stylem z wykorzystaniem CSS, aby nabrała ona odpowiedniego wyglądu. Ponieważ szerokość została ustalona na 1050px, tak więc kod CSS będzie wyglądał następująco:

```
* {padding:0;margin:0;}
```

```
html {width:100%;height:100%;}
```

```
body {font-family:Arial, Helvetica, Sans-serif;font-size:14px;line-height:1.5;font-weight:400;color:#4
```

```
.container {margin:0 auto;}
```

```
header {margin:25px 0;}
```

```
header img {width:100%;}
```

```
aside {float:left;margin:0 0 25px;}
```

```
aside nav ul {list-style:none;}
```

```
aside nav ul li {margin-bottom:1px;}
```

```
aside nav ul li a {display:block;text-decoration:none;color:#000;background:#ff0040;padding:8px 20px;}
```

```
aside nav ul li a:hover{background:#212121;color:#FFF;}
```

```
#main {float:right;margin:0 0 25px;}
```

```
footer {clear:both;color:#FFF;background:#FF0000;padding:15px 0 15px 30px;margin:0 0 25px;}
```



Źródło: Piotr Napieralski, licencja: CC BY 3.0.

Szablon można rozbudować o kolejne podstrony.

Publikacja i administracja strony WWW

Po utworzeniu stron internetowych, ważnym zadaniem jest pokazanie ich internautom. W celu udostępnienia strony internetowej publicznie, można skorzystać z darmowych usług hostingu. Takie możliwości daje np. portal „cba.pl”. Aby z nich skorzystać, wystarczy zarejestrować się na stronie www.cba.pl, podając potrzebne dane.

W miejscu nazwy domeny należy podać nazwę, która będzie w adresie strony. To może być nick, w przypadku strony domowej, lub fragment opisujący zawartość np. „pasmagorskie”. Domena może składać się z małych liter, myślników i cyfr:

Wybierz nazwę domeny, jedną z naszych (*cba.pl*, *c0.pl*, *y0.pl* oraz tylko dla planu **Pro**: *j.pl*) lub Twoją własną (o ile już ją posiadasz), bez *www*.

pasmagorskie	
pasmagorskie.cba.pl	Free Pro
pasmagorskie.c0.pl	Free Pro
pasmagorskie.y0.pl	Free Pro
pasmagorskie.j.pl	Free Pro
pasmagorskie.pl	domena od 12.90 PLN Free Pro
pasmagorskie.com.pl	domena od 9.90 PLN Free Pro
pasmagorskie.net.pl	domena od 9.90 PLN Free Pro

Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

W celu utworzenia darmowego konta należy wybrać opcję „Free” spośród bezpłatnych domen. Następnie podać nazwę użytkownika, hasło do konta i adres e-mail. Dodatkowo można wprowadzić szczegóły odnośnie stworzonej strony, takie jak kategoria i tytuł, aby łatwiej była ją znaleźć w wyszukiwarce internetowej.

The screenshot shows a web browser window with the URL www.cba.pl/panel/pl/register/domain/pasmagorskie.cba.pl/plan/free. The form is titled "Dane konta" and includes fields for: Nazwa konta (filled with "blaszczyka654"), Hasło (filled with "*****"), Powtórz hasło (filled with "*****"), Siła hasła (a progress bar), Adres email (filled with "blaszczyka654@outlook.com"), and Powtórz adres email (filled with "blaszczyka654@outlook.com"). Below this is a section "Dane użytkownika" with a sub-section "Wpisz w katalogu" containing fields for: Kategoria (a dropdown menu with "Natura" selected), Tytuł (filled with "Europejskie pasma górskie"), and Opis (an empty text area). At the bottom right of the form is a button labeled "Utwórz konto". The footer of the page reads "Copyright © 2005 - 2015 CBA.pl".

Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Dostęp do serwera przechowującego pliki strony u większości usługodawców jest możliwy za pomocą protokołu transferu plików - FTP (File Transfer Protocol). Wygodnym sposobem na przesłanie danych jest wykorzystanie, na przykład, programu FileZilla, dostępnego dla różnych systemów operacyjnych. W głównym oknie programu można wprowadzić adres serwera i dane logowania, które będą opisane w szczegółach lub pomocy na portalu usługodawcy.

Jakie są parametry konta FTP/menadżera plików?

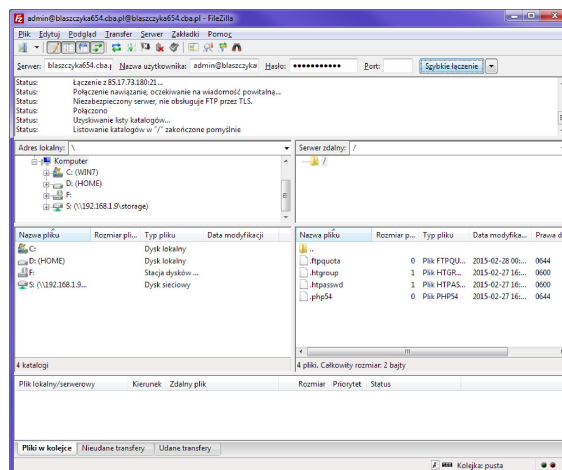
serwer: nazwatwojegokonta.cba.pl (np. user.cba.pl)

użytkownik: admin@login_do_panelu (np. admin@user.cba.pl)

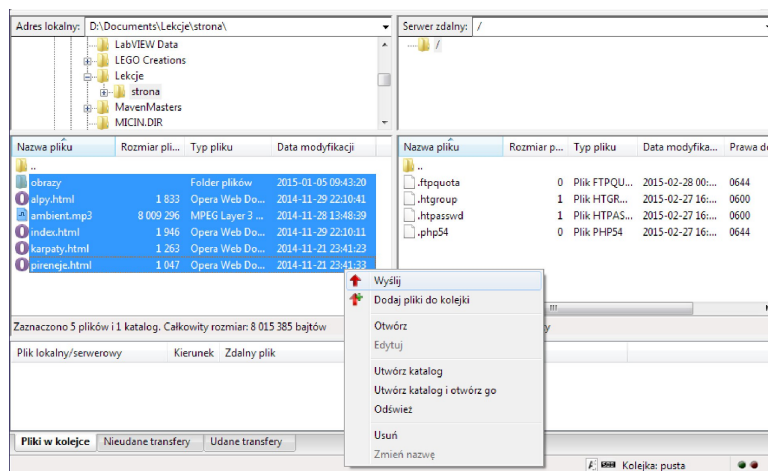
hasło: Takie jak podałeś przy tworzeniu konta

Nazwa użytkownika zawiera @, nie jest to błąd. Czyli w okienko Login w kliencie FTP lub file-menedżerze wpisujemy admin@user.cba.pl.

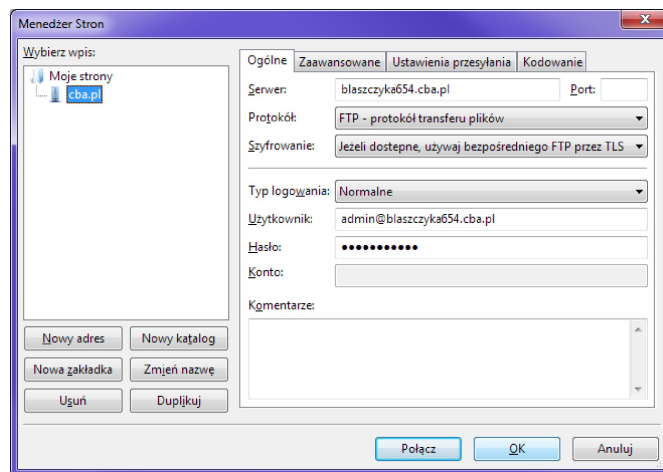
Po pomyślnym połączeniu, po prawej stronie programu pojawiają się pliki przechowywane na serwerze:



Po pierwszym połączeniu może być widoczna przykładowa zawartość strony oraz pliki konfiguracyjne. Jeżeli wśród elementów na serwerze znajduje się folder „public_html”, to w nim należy umieszczać pliki strony. W innym wypadku można je kopiować bezpośrednio do katalogu głównego, jednak zawsze warto się upewnić, sprawdzając w pomocy na portalu usługodawcy. W celu przesłania plików należy znaleźć folder ze stroną na swoim komputerze, nawigując w lewej części okna programu do miejsca zawierającego plik „index.html”. Następnie trzeba zaznaczyć wszystkie elementy i przeciągnąć do okna po prawej stronie, lub klikając na nie prawym przyciskiem myszy wybrać „Wyslij”:

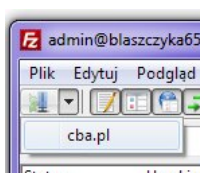


Postęp widoczny będzie na dole. Przesyłanie może potrwać dłużej, w zależności od rozmiaru plików. Po skończeniu przesyłania, należy wpisać w przeglądarce adres strony, w celu sprawdzenia czy działa ona poprawnie. Głównym plikiem strony jest „index.html”. Jest on wczytywany po wpisaniu bezpośredniego adresu w przeglądarce. Przykładowo: <http://www.pasmagorskie.cba.pl/> jest odpowiednikiem: <http://www.pasmagorskie.cba.pl/index.html> Dotyczy to również podkatalogów na stronie: <http://www.pasmagorskie.cba.pl/podkatalog/> odpowiada adresowi: <http://www.pasmagorskie.cba.pl/podkatalog/index.html> Jeżeli wszystko odbyło się bez problemów, można zapisać połączenie do serwera FTP w pamięci programu FileZilla. Z menu „Plik” należy wybrać „Kopiuj aktualne połączenie do Menedżera Stron”, podać jego nazwę i zatwierdzić przyciskiem OK.



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Później można skorzystać z zapisanego połączenia, wybierając je z rozwijanego menu obok pierwszego przycisku na górze.

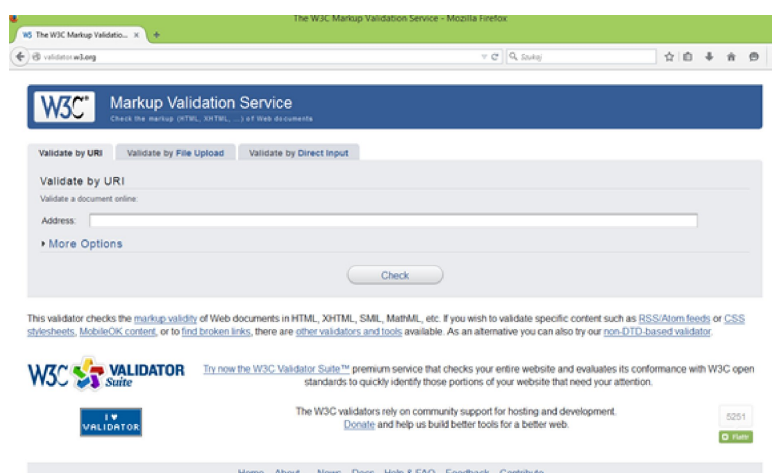


Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

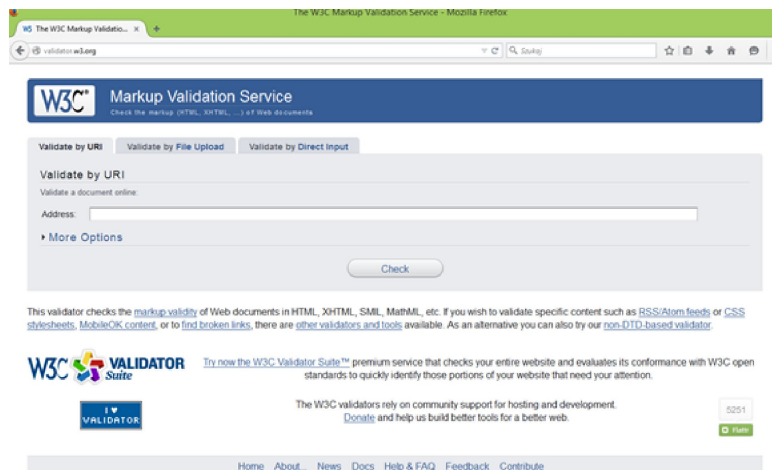
Walidacja stron internetowych

Przed opublikowaniem strony w sieci należy upewnić się, że jest ona poprawnie skonstruowana. Pomimo dobrego wyglądu, może zawierać błędy znajdujące się w jej kodzie. Aby nie publikować wadliwych dokumentów (ładują się dłużej), należy sprawdzić poprawność kodu strony pod kątem jego zgodności ze specyfikacją języka. Do tego celu służą specjalne aplikacje, tzw. walidatory, które dodatkowo zwracają uwagę na brakujące lub źle użyte znaczniki, błędne ich zagnieżdżanie oraz na złe użycie stylów.

Jedną z nich jest aplikacja udostępniona przez organizację W3C (World Wide Web Consortium) na stronie pod adresem <http://validator.w3.org>

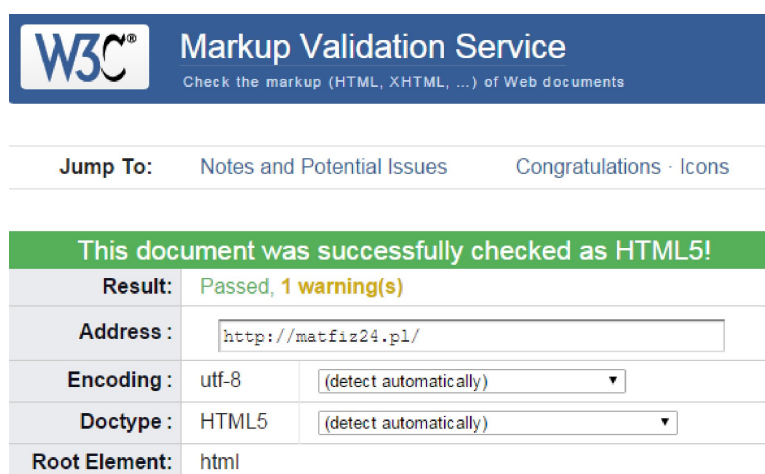


Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

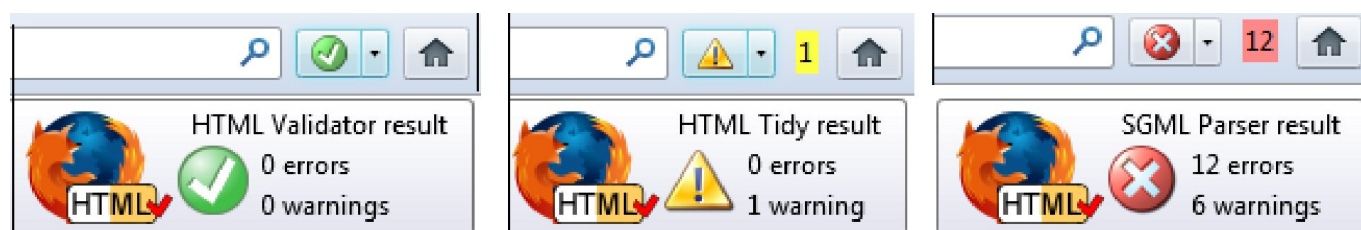


Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Umożliwia ona sprawdzenie kodu strony poprzez podanie adresu strony internetowej, załadowanie pliku HTML strony lub wklejenie bezpośrednie jej kodu HTML. Po wybraniu jednej z powyższych możliwości, po kliknięciu „Check” zostanie ukazany wynik.



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Zielone koło oznacza poprawność kodu HTML, pomarańczowy trójkąt, że na stronie znajdują się drobne zastrzeżenia, a czerwone koło, że strona posiada poważne błędy. Za dopuszczalny można uznać wynik z maksimum 3 błędami.

Będąc pewnym, że strona jest poprawnie zbudowana, można śmiało udostępnić ją w sieci internetowej.

Zarządzanie kontem pocztowym projektu

Korzystając z projektowego konta pocztowego (np. konta Microsoft) należy pamiętać o omówionych w poprzednich klasach zasadach netykiety, czyli swoistego kodeksu zachowania obowiązującego wszystkich użytkowników sieci.

Obecnie, netykieta uznana za formę kodeksu Internautów, pełni rolę pośrednią między zwyczajem a prawem.



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Treści netykiet znajdują się od niedawna w Kodeksie Cywilnym i dotyczą: tajemnicy korespondencji, plagiatu, używania niewłaściwego języka. Z kolei w polskim Kodeksie Karnym (k.k.) z 1997 roku znajdujemy je w treściach dotyczących:

- ochrony prawa własności (wszystkie zasoby sieciowe znajdujące się na serwerach),
- tajemnicy korespondencji znajdujących się w skrzynkach poczty elektronicznej.

W przypadku prawa karnego, praktyka dowodzi, iż popełniane przestępstwa można rozpatrywać w oparciu o ustawy takie jak: Ustawa o świadczeniu usług drogą elektroniczną czy Ustawa o prawie autorskim.

Innymi regulacjami dotyczącymi funkcjonowania Internetu są regulaminy tworzone przez firmy udostępniające bezpłatne korzystanie z kont e-mail i www, a także regulaminy kawiarenek internetowych.

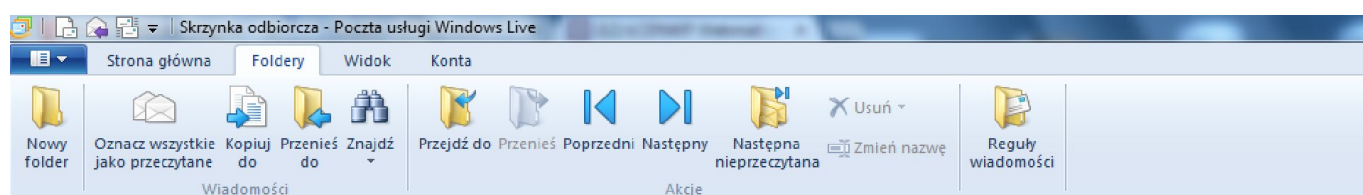
Jak widać, netykieta i prawo wzajemnie się uzupełniają. Netykieta może być więc uznana za część norm życia społecznego, ale rzadko spotykamy odwoływanie się do jej zasad w orzecznictwie sądowym. W polskim prawie nie ma ich spisanych w formie jednolitego „prawa internetowego”, ale kwestie ewentualnych przestępstw są uwzględniane we wspomnianych kodeksach (karnym, cywilnym) oraz w prawie pracy, prawie podatkowym i prawie prasowym.

Pamiętając o przestrzeganiu zasad netykiety można przystąpić do wykorzystywania serwisów pocztowych i klientów poczty elektronicznej do usprawnienia codziennej pracy. W gimnazjum wykorzystywany był program pocztowy, tzw. klient, **Windows Live Mail**. Pozwala on odbierać pocztę z wielu kont e-mail równocześnie. Uruchomienie programu powoduje połączenie go z serwerem poczty przychodzącej i odebranie poczty. Odebrane listy zapisywane są w folderach: **Szybki podgląd**, **Nieprzeczytane wiadomości** oraz w folderze **Skrzynka odbiorcza**.

Dobrze jest jednak znać i korzystać z przedstawionych poniżej, bardziej zaawansowanych możliwości programu, przydatnych podczas pracy nad realizowanym projektem.

- Zarządzanie informacjami na koncie pocztowym poprzez tworzenie układu folderów

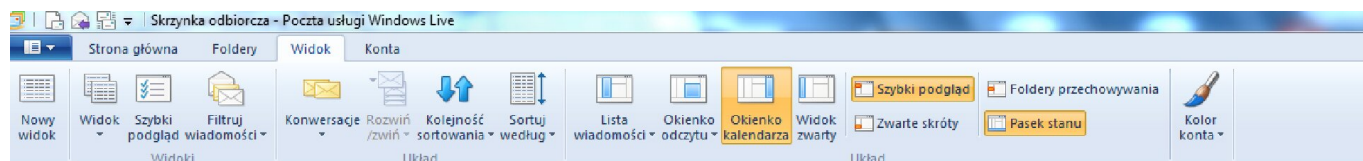
Z biegiem czasu liczba e-mali zwiększa się. Porządkując je, można niektóre z nich przenieść do utworzonego folderu o samodzielnie wybranej nazwie. Wystarczy na karcie poleceń **Foldery** kliknąć ikonę **Nowy folder**. Można również przenieść wybrane listy do innego folderu korzystając z ikon **Kopijuj do** i **Przenieś do** widocznych na karcie **Foldery**.



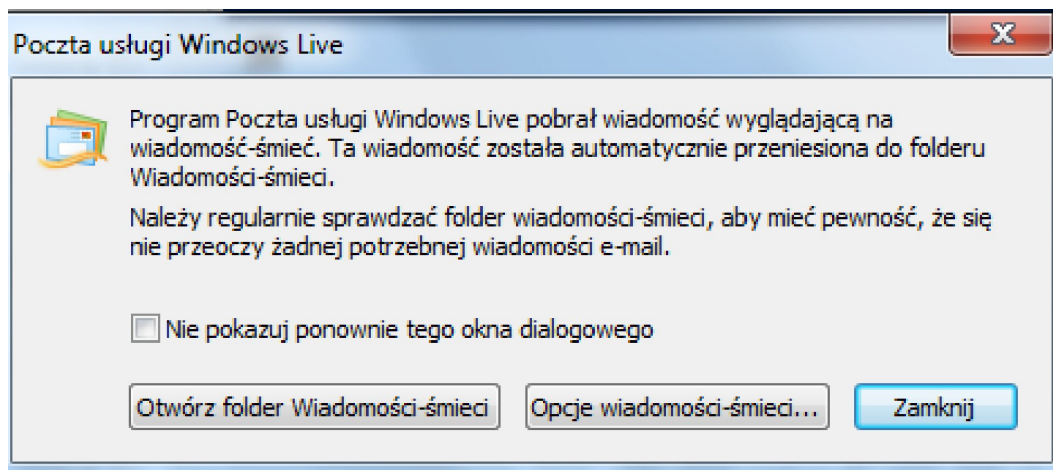
Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

- Filtrowanie korespondencji

Program jest wyposażony w filtr nazywany filtrem spamu. Przegląda on treść nadesłanych wiadomości i przenosi podejrzone do folderu wiadomości-śmieci.

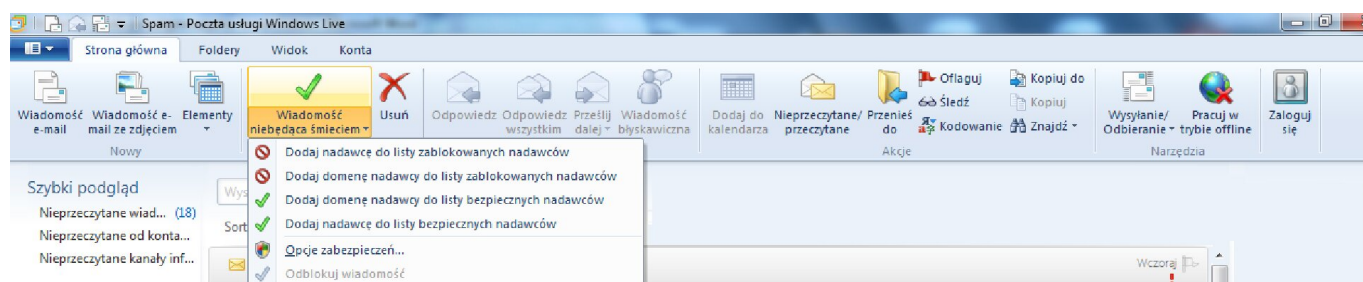


Źródło: Sławomir Szaruga, licencja: CC BY 3.0.



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

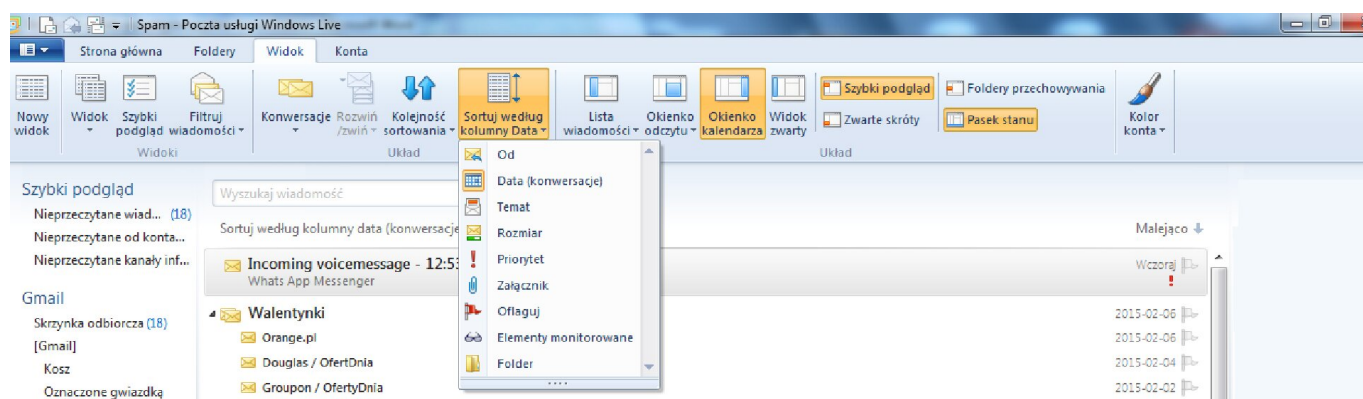
W dowolnym czasie można je przejrzeć i te naprawdę podejrzane usunąć definitywnie.



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

- Sortowanie wiadomości

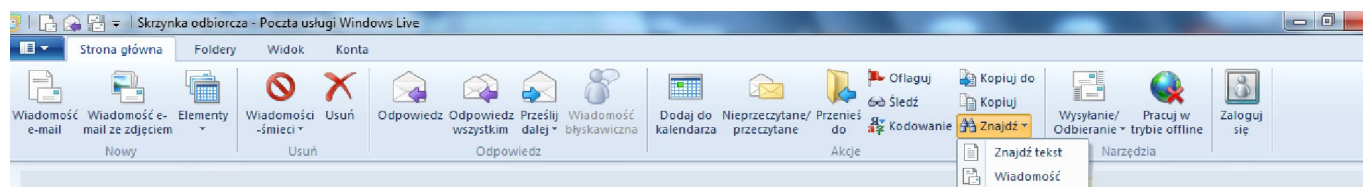
Układ wiadomości w skrzynce odbiorczej można zmienić dokonując czynności sortowania. W tym celu należy zaznaczyć wybraną kolumnę i klikając kartę poleceń **Widok** wybrać **Kolejność sortowania**, rosnąco lub malejąco. Podobny efekt otrzyma się dwukrotnie klikając nagłówek kolumny, np. *Temat*, *Data* lub *Rozmiar*. Zmianie ulegnie sposób sortowania z rosnącego na malejący i odwrotnie. Można również dokonać wyboru kolumny korzystając z polecenia **Sortuj według kolumny ...**

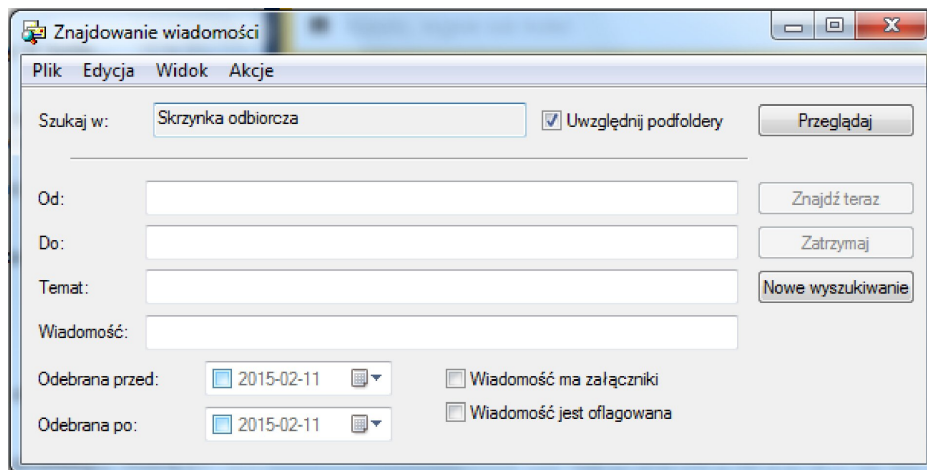


Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

- Znajdowanie wiadomości

Można przeszukać skrzynkę odbiorczą w poszukiwaniu konkretnego listu. W tym celu wystarczy kliknąć ikonę **Znajdowanie wiadomości** i w ten sposób uzyskać możliwość odnalezienia e-maila według nadawcy, tematu i treści wiadomości.

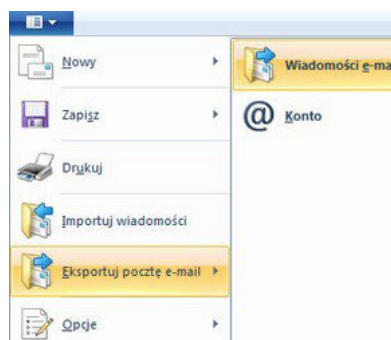




Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

- Importowanie i eksportowanie wiadomości

Przy wymianie komputera, wiadomości znajdujące się w skrzynce odbiorczej każdy chce przenieść do programu pocztowego w nowym komputerze. Należy więc wykonać operację wykonywania kopii zapasowych. Przed jej rozpoczęciem trzeba utworzyć folder w wybranej lokalizacji. Po jego utworzeniu można przystąpić do wyeksportowania wiadomości. W tym celu w menu **Plik**, trzeba kliknąć polecenie **Eksportuj pocztę e-mail**, a następnie wybrać pozycję **Wiadomości e-mail**.



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Najlepiej jest wybrać format eksportu w postaci „**Poczta usługi Windows Live firmy Microsoft**”.

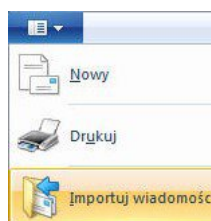
Następnie należy wskazać folder, w którym mają być zapisane wiadomości i ponownie kliknij **Dalej**.

Kolejnym krokiem jest zaznaczenie elementów, które mają być wyeksportowane, po zaznaczeniu odpowiednich folderów należy kliknąć przycisk **Dalej**.

Jeśli wszystko przebiegło pomyślnie, powinien pojawić się komunikat o powodzeniu operacji eksportu. Wystarczy teraz zakończyć operację odpowiednim poleceniem.

Jeśli chcemy otworzyć pocztę w nowej lokalizacji należy dokonać importu. Trzeba w tym celu wybrać z menu **Plik** polecenie **Importuj wiadomości**.

Następnie należy zaznaczyć format pliku w którym zarchiwizowane zostały wiadomości i kliknąć polecenie **Dalej**. Należy zlokalizować folder w którym znajduje się zapisane archiwum za pomocą polecenia **Przeglądaj**, i znowu kliknąć **Dalej**. Jeśli operacja przebiegła pomyślnie, można będzie podejrzeć zaimportowane wiadomości.



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

- Książka kontaktów

Przy prowadzeniu rozbudowanej korespondencji przydatna jest **Lista kontaktów**, która ułatwia adresowanie korespondencji. Wystarczy pisać nowy list klikając DO, DW lub UDW a następnie wybrać kontakt, jako adresata wiadomości. Dostęp do niej można uzyskać, klikając w ikonę **Kontakty**. Książkę kontaktów można aktualizować ręcznie lub automatycznie. W tym drugim przypadku automatycznie dodany do listy zostanie adres pocztowy, na który trzykrotnie została wysłana odpowiedź.

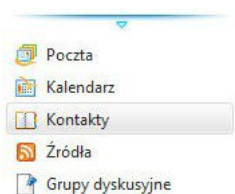
Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Można w książce kontaktów utworzyć tzw. listy dystrybucyjne, czyli grupy kontaktów. Uzyskuje się wówczas możliwość wysyłania jednocześnie jednego listu do grupy odbiorców. Listę dystrybucyjną można utworzyć klikając ikonę **Kategoria** karty poleceń **Strona główna kontaktów**. W otwartym oknie należy wpisać jej nazwę i wybrać kontakty, które powinny się w niej znaleźć. Po zakończeniu pracy nad kompletowaniem adresów trzeba kliknąć **Zapisz**, a nowa lista zostanie utworzona.

Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

- Importowanie i eksportowanie kontaktów

Równie cenne jak wiadomości są informacje znajdujące się w książce kontaktów. Można je wyeksportować, klikając pozycję **Poczta usługi Windows Live**, a w niej **Kontakty**.



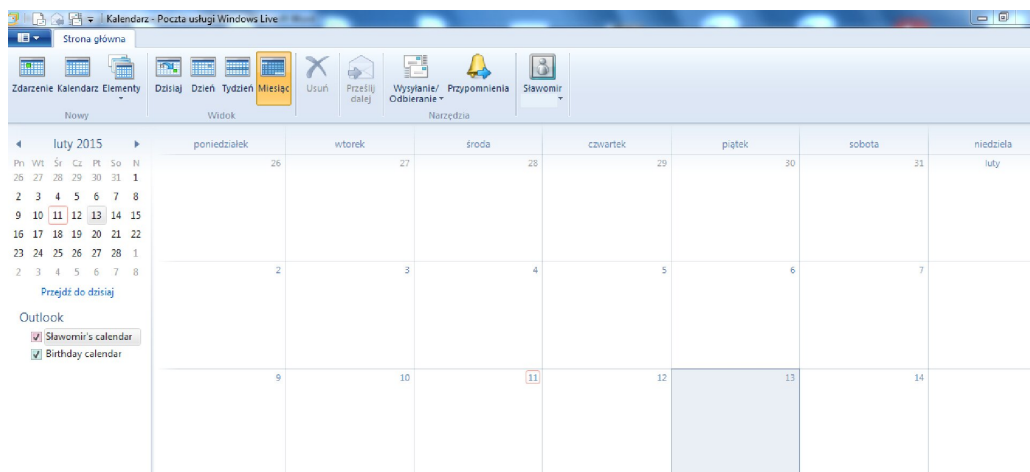
Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Na ukazanej wstążce należy kliknąć przycisk **Eksportuj**, w menu rozwijanym zaznaczyć format pliku (.vcf dla jednego lub dwóch kontaktów, .csv przy większej liczbie), wskazać miejsce zapisu kontaktów korzystając z przycisku **Przeglądaj**. Następnie należy nazwać powstały plik i kliknąć **Zapisz**. Po kliknięciu przycisku **Dalej**, trzeba zaznaczyć pola, które chce się wyeksportować i kliknąć **Zakończ**. Aby zaimportować kontakty trzeba również skorzystać z pozycji **Kontakty** poczty Windows Live Mail z tym, że teraz ze wstążki należy wybrać przycisk **Importuj**. Następnie zaznaczyć format wyeksportowanych kontaktów i korzystając z przycisku **Przeglądaj** znaleźć folder

zawierający kontakty. Podobnie jak poprzednio trzeba kliknąć **Dalej** i zaznaczyć pola do zaimportowania. Na koniec należy kliknąć **Zakończ**.

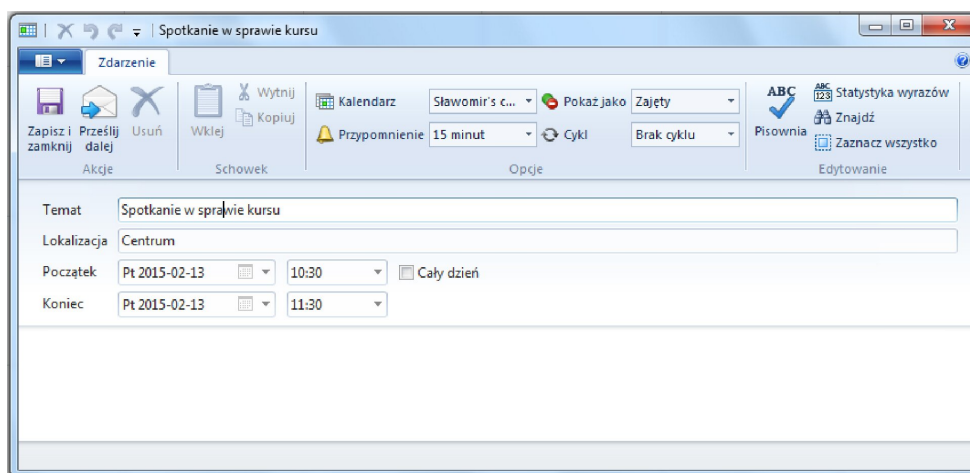
- Kalendarz usługi Windows Live Mail

Ikona Kalendarza (lewy dolny róg programu) umożliwi korzystania z kalendarza w układzie dziennym, tygodniowym lub miesięcznym.



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

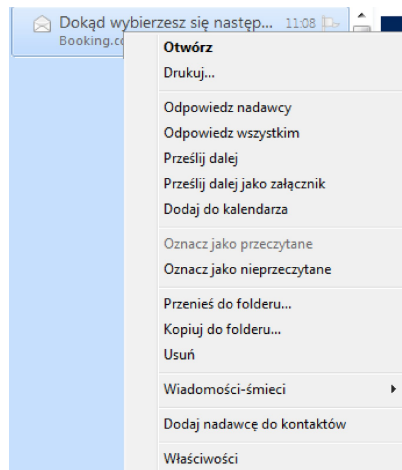
Kliknięcie w datę lub godzinę umożliwia edycję informacji o zdarzeniu. Podwójne kliknięcie otwiera okno w którym można wpisać dodatkowe informacje np. początek i koniec imprezy. Każdy wpis można usunąć poleceniem **Usuń** z menu podręcznego.



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Ważne!

Umieszczając wskaźnik myszy na dowolnym e-mailu znajdującym się w skrzynce odbiorczej i wybierając z menu podręcznego polecenie Dodaj do kalendarza możesz dodać go do kalendarza.



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Bezpieczeństwo w Internecie

Bezpieczeństwo w internecie było wielokrotnie omawiane w poprzednich klasach, jednak ze względu na wagę problemu, przypominamy kilka podstawowych zasad.

Internet umożliwia zdobywanie informacji, dokonywanie zakupów, operacji finansowych oraz współpracę z innymi jego użytkownikami nad tworzeniem nowych treści. Korzystając z niego można się jednak natknąć na wiele niebezpieczeństw. By się przed nimi uchronić należy pamiętać o zachowaniu ostrożności. Poniżej przedstawione porady pozwolą poruszać się bezpiecznie w sieci internetowej, podczas korzystania z usług internetowych oraz administrowania stroną WWW.

Logowanie się i wylogowywanie

Powszechnie stosowanym sposobem zabezpieczania dostępu do strony internetowej jest logowanie, które pozwala na autoryzację użytkownika. Wcześniej administrator powinien w procesie rejestracji przydzielić każdemu użytkownikowi niepowtarzalny w serwisie login oraz hasło. Bezpieczeństwo w sieci zwiększa korzystanie z bezpiecznych stron internetowych oraz wylogowanie się z nich po zakończeniu pracy.

Na przykład w przeglądarce IE połączenia z bezpieczną witryną sygnalizowane są zielonym kolorem paska adresu.



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Reputację strony można również uzyskać korzystając z poznanego w gimnazjum bezpłatnego programu WOT (Web of Trust) służącego do określenia witryn internetowych pod względem zaufania, wiarygodności i bezpieczeństwa dzieci.



Ważne!

Zawsze, po zakończeniu lub przerwaniu na jakiś czas pracy z komputerem stacjonarnym, laptopem lub tabletem, należy zablokować ekran. Sprzęt powinien być tak skonfigurowany, aby w chwili przejścia w tryb uśpienia blokował się automatycznie, a wznowienie

pracy było możliwe po wprowadzeniu stosownego hasła.

Zabezpieczanie haseł

Korzysta się z nich w rozwiązaniach zapewniających bezpieczeństwo kart bankomatowych, kont bankowych, telefonów, haseł komputerowych. Jednak często hasła i kody są ustawione w dość przewidywalny sposób. Najczęściej są to informacje takie jak data urodzenia, czy imię dziecka. Czasami jedno hasło i stosuje się w wielu miejscach, np. do karty bankomatowej, telefonu czy laptopa. Ktoś kto pozna to hasło ma wówczas dostęp do wszystkich chronionych nim kont. Z reguły strony internetowe banków, czy sklepów internetowych, dają swoim użytkownikom możliwość zmiany hasła i należy z tej możliwości korzystać. Nie powinny to być jednak hasła, które są bardzo popularne i w związku z tym łatwe do odgadnięcia. Powinny być odpowiednio długie min. osiem znaków, a idealne to hasło około czternasto znakowe, zawierające wielkie i małe litery oraz symbole.

Najpopularniejsze hasła bezpieczeństwa na świecie w 2014 roku

Lp.	Hasło	Lp.	Hasło
1.	123456	11.	1234567
2.	password	12.	monkey
3.	12345	13.	letmein
4.	12345678	14.	abc123
5.	qwerty	15.	111111
6.	123456789	16.	mustang
7.	1234	17.	access
8.	baseball	18.	shadow
9.	dragon	19.	master
10.	football	20.	michael

Źródło: Piotr Napieralski, licencja: CC BY 3.0.

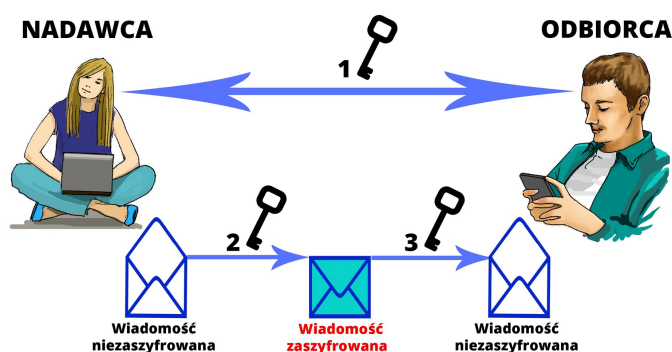
Uwaga!

Przy wpisywaniu hasła bezpieczniej jest używać klawiatury ekranowej, ponieważ istnieją programy tzw. keyloggers, które rejestrują klawisze naciskane przez użytkownika.

Połączenia szyfrowane

Podczas korzystania z Internetu przekazywane informacje przechodzą przez wiele komputerów i są narażone na przeczytanie przez osoby do tego nieupoważnione. Aby chronić przesyłane w sieci dane, wykorzystywane są metody kryptografii. Kryptografia jest dziedziną obejmująca zagadnienia związane z utajnieniem (zaszyfrowaniem) danych w przesyłanych wiadomościach i zabezpieczaniem przed niepożądanym do nich dostępem. Rozszyfrowanie wiadomości będzie możliwe tylko przez osobę znającą specjalny klucz rozszyfrowujący. Dla osób nie znających klucza rozszyfrowującego wiadomość będzie wyłącznie niezrozumiałym ciągiem znaków.

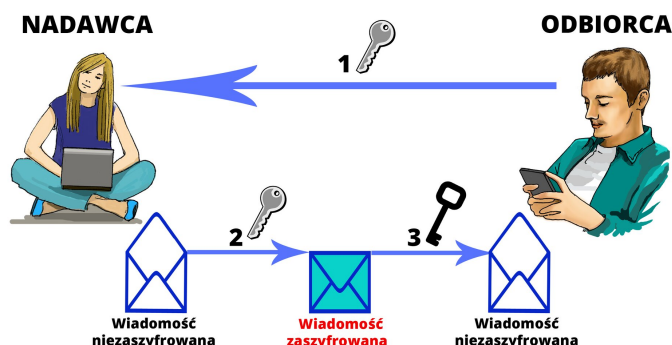
Prawie codziennie korzystając z Internetu wykorzystuje się rozmaite kody i szyfry, a także algorytmy autoryzacji. Wszystkie wcześniej stosowane szyfry miały charakter symetryczny czyli taki, w którym nadawca i odbiorca używali tego samego klucza, którego znajomość była niezbędna do zaszyfrowania jak i odszyfrowania wiadomości. Czyli szyfrowanie symetryczne jest skuteczne o ile klucz nie wpadnie w niepowołane ręce.



Źródło: Ewa Dobrzyńska, licencja: CC BY 3.0.

Szyfry symetryczne są również i obecnie wykorzystywane do przesyłania wiadomości przez Internet. Do tego rodzaju szyfrów należy, opracowany przez firmę Netscape Communications, protokół kryptograficzny SSL (Secure Socken Layer). SSL jest najczęściej kojarzony z protokołem HTTP (HTTPS) i jest obsługiwany przez większość przeglądarek internetowych. W kryptografii istotną rolę odgrywa również długość wykorzystywanych kluczy, czyli w tym przypadku liczby składających się nań bitów. Klucz długi jest trudniejszy do złamania. Współcześnie używane klucze liczą 128-bitów.

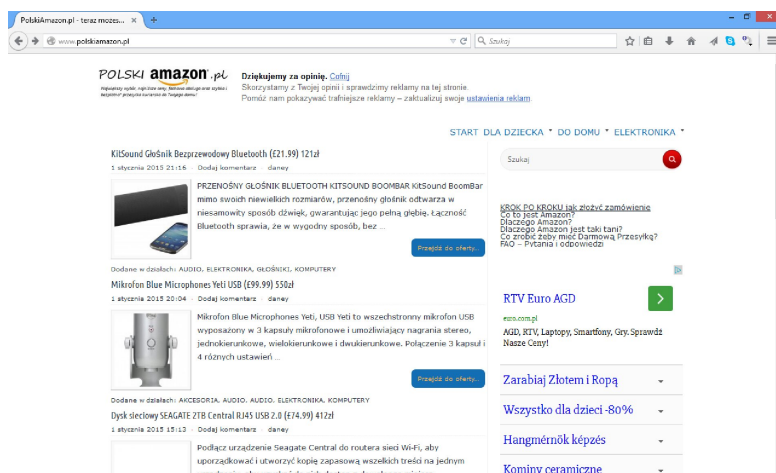
Od końca lat siedemdziesiątych XX wieku stosowana jest kryptografia asymetryczna oparta na dwóch kluczach: publicznym, służącym do szyfrowania i rozprowadzanym wśród użytkowników oraz kluczu tajnym, prywatnym – służącym do odczytywania wiadomości. Klucz który stosowany do szyfrowania wiadomości nie daje możliwości jej odszyfrowania. Szyfrowanie bazujące na kluczu asymetrycznym zapewnia wysoki poziom bezpieczeństwa. Znacznie wyższy niż w metodzie stosującej jeden, wspólny klucz.



Źródło: Ewa Dobrzyńska, licencja: CC BY 3.0.

Nadawca szyfruje wiadomość kluczem publicznym otrzymanym od Odbiorcy, który następnie otrzymuje zaszyfrowaną wiadomość i rozszyfrowuje ją kluczem prywatnym.

Protokół SSL używany jest przede wszystkim przez witryny sklepów internetowych, zapewniając możliwość dokonania zakupów bez obawy o przechwycenie poufnych danych (np. numerów kart kredytowych.). Przed uzyskaniem możliwości zakupu należy wypełnić formularz zgłoszeniowy, w którym będzie zawarta prośba o podanie adresu e-mail i domowego. Adres poczty elektronicznej będzie potrzebny do potwierdzenia zamiaru dokonania zakupu, a zakupiony towar zostanie przesłany na adres domowy za pośrednictwem tradycyjnej poczty. Zakupy można dokonywać w wielu sklepach, a nawet w największym sklepie internetowym na świecie (własność amerykańskiego przedsiębiorstwa Amazon), działającym w naszym kraju pod adresem www.polskiamazon.pl.

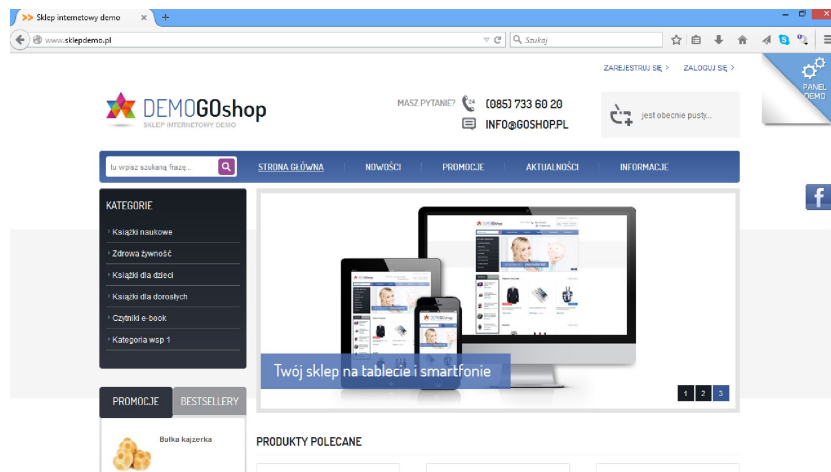


Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Dokonując zakupów poprzez Internet należy pamiętać o:

- korzystaniu z tego typu serwisów sklepowych ze swojego domowego sprzętu,
- zachowaniu w tajemnicy haseł dostępu i danych osobowych,
- wyborze sposobu płatności, np. gotówką przy odbiorze lub z posiadanego przez ciebie rachunku bankowego,

Sposób dokonywania zakupu w sklepie internetowym można prześledzić na demonstracyjnej stronie WWW: <http://www.sklepdemo.pl/>



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Niestety przy zastosowaniu szyfrowania asymetrycznego istnieje możliwość zastąpienia klucza publicznego kluczem nie należącym do danej firmy. Wysyłając wiadomości warto sprawdzić, czy klucz publiczny jest kluczem prawdziwym. Dlatego wprowadza się tutaj cyfrowe certyfikaty wydawane przez instytucję zaufania publicznego. Przeglądarki internetowe są przygotowane do obsługi cyfrowych certyfikatów, które w protokole SSL zawierają m.in. nazwę domeny, dla której zostały wystawione. Chcąc sprawdzić, czy strona jest zabezpieczona certyfikatem bezpieczeństwa, wystarczy kliknąć podwójnie w ikonę kłódki na pasku adresowym. W ten sposób są chronione strony poważnych instytucji finansowych, takich jak np. banki.

Po przejściu do zakładki **Połączenie** można uzyskać informacje o certyfikacie potwierdzającym, że połączono się z właściwą stroną internetową, na przykład z bankiem, a nie udającą go witryną. Kliknięcie polecenia **Informacje o certyfikacie** pokaże okno umożliwiające sprawdzenie autentyczności i daty ważności certyfikatu.

Uwaga

Uwaga!

Zawsze przed przystąpieniem do sesji należy sprawdzić czy:

- W przeglądarce jest włączona obsługa szyfrowania.
- W pasku adresowym przeglądarki adres internetowy banku jest poprzedzony skrótem „https”. Litera „s” oznacza wspomniane wcześniej szyfrowanie protokołu komunikacji a zatem wprowadzony przez ciebie login i hasło nie dadzą się odczytać przez inne osoby.
- Pasek adresowy jest zabarwiony na zielono.
- W polu adresu jest kłódka sygnalizująca, że strona jest zabezpieczona certyfikatem bezpieczeństwa.
- Data logowania jest datą faktycznego korzystania z usługi w e-banku.

Sposób przeprowadzania operacji bankowych można prześledzić na demonstracyjnych wersjach stron banków internetowych, np. banku ipko <http://demo.ipko.pl/html/login.html>

- Sprawdzanie autentyczności dokumentów przesyłanych drogą elektroniczną

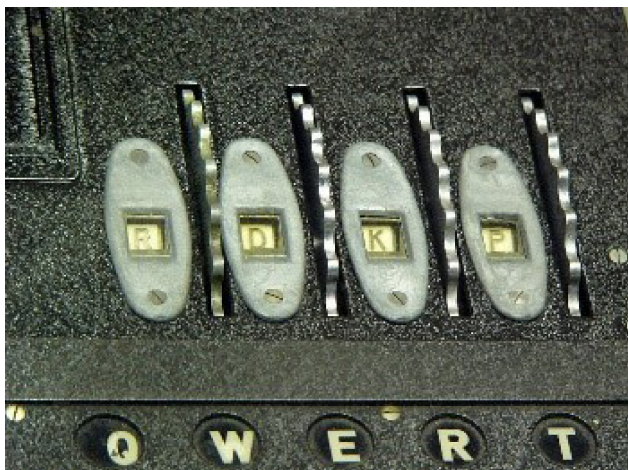
Coraz częściej pojawia się z możliwością przekazywania dokumentów drogą elektroniczną. Istnieje wiele sposobów, dzięki którym można stwierdzić, że dana osoba jest autorem przekazywanego dokumentu. Taką osobę może np. charakteryzować podpis elektroniczny oraz informacja biologiczna.

- Podpis cyfrowy (z ang. DSA - Digital Signature Algorithm) to dane w postaci elektronicznej, pozwalające na identyfikację osoby składającej podpis na dokumentach elektronicznych. Są one przedstawione w postaci ciągu cyfr powstającego w wyniku wykorzystania technik kryptograficznych utajniających dane poprzez ich szyfrowanie. Dzięki nim możemy przekształcić normalny tekst w taki sposób, iż stanie się on niezrozumiały dla nieupoważnionego odbiorcy. Osoba otrzymująca wiadomość tego typu jest w stanie przekształcić ją do pierwotnej czytelnej postaci. W podpisach cyfrowych stosuje się kryptografię asymetryczną.

Kolejność czynności przy tworzeniu dokumentu z podpisem cyfrowym jest następująca:

- nadawca pisze wiadomość,
- nadawca wytwarza wiadomość,
- nadawca podpisuje wiadomość za pomocą algorytmu podpisu cyfrowego (używając swojego prywatnego klucza),
- nadawca łączy wiadomość i podpis w celu uzyskania nowej, podpisanej wiadomości,
- nadawca tworzy losowy klucz szyfrujący do zaszyfrowania podpisanej wiadomości,
- nadawca szyfruje wiadomość za pomocą klucza i algorytmu konwencjonalnego,
- nadawca pobiera jawny klucz odbiorcy,
- nadawca szyfruje pobranym kluczem wygenerowany przez siebie losowy klucz szyfrujący,
- nadawca łączy zaszyfrowaną wiadomość z zaszyfrowanym kluczem,

- nadawca wysyła zabezpieczoną wiadomość.



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

- U podstaw identyfikacji biometrycznej leży fakt, że pewne cechy ciała ludzkiego są przypisane do konkretnej osoby i są niezmiennie. Stosowane są następujące metody identyfikacji biometrycznej:
 - skanowanie linii papilarnych,



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

- skanowanie tęczówki,
- identyfikacja głosu,
- rozpoznawanie twarzy.

Wyłudzenie informacji

By chronić się przed wyłudzeniem informacji poprzez Internet dobrze jest poznać najpopularniejsze z tych sposobów. Należą do nich phishing i pharming.

- Phishing polega na przekonaniu użytkownika do podania danych osobowych poprzez fałszywą stronę.
- Pharming obejmuje modyfikację wpisów DNS, co powoduje przekierowanie użytkownika do niewłaściwej strony, kiedy odwiedzi pewien adres WWW.

W przypadku phishingu oszuści wysyłają e-maile, które wydają się pochodzić z legalnych stron internetowych np. instytucji bankowych. W treści e-maila może być zawarta np. prośba o podanie nazwy użytkownika i hasła motywowana koniecznością aktualizacji konta. Dysponując tymi informacjami oszust może uzyskać dostęp do większej ilości informacji (numeru telefonu, numeru PESEL czy numeru karty kredytowej) logując się do udostępnionego konta.

Fałszywe e-maile często wyglądają zaskakująco zgodne z prawdą, a fałszywe strony internetowe, mogą wyglądać jak prawdziwe. Po otrzymaniu e-maila z prośbą o zaktualizowanie informacji należy przejść do strony internetowej, wpisując ręcznie adres URL w polu adresu przeglądarki, zamiast klikania w link w wiadomości e-mail. Jeśli po wpisaniu adresu WWW zostanie wyświetlona prośba o aktualizację danych, to e-mail był prawdziwy. W przeciwnym wypadku e-mail był najprawdopodobniej próbą zdobycia informacji przez tzw. phishera.

Skopiowane logo aby
uporządkować autentyczność



Drogi kliencie,
ze względu na bezpieczeństwo
naszych klientów zmuszeni
jesteśmy do potwierdzenia
danych znajdujących się w
naszej bazie danych.
Prosimy wejść na [naszą stronę](#)
i potwierdzić numer karty
kredytowej oraz związane z nią
dane.

Wikibank

Link do strony
"udającej" stronę
prawdziwego banku

Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Ochrona przed phishingiem

- Nie należy otwierać hiperłączy bezpośrednio z otrzymanego e-maila.
- Prośby o podanie hasła i loginu w mailu należy zignorować i zgłosić odpowiednim osobom (firmie).
- Jeśli strona z logowaniem do instytucji finansowej nie zawiera w adresie nazwy protokołu HTTPS, nie powinno podawać się na niej żadnych danych.

W pharmingu, podobnie jak w przypadku ataków typu „phishing”, użytkownicy wabieni są do fałszywych witryn internetowych. Strony te do złudzenia przypominają strony oryginalne. W przeciwieństwie do phishingu w pharmingu użytkownik przekierowany jest na fałszywą stronę nawet jeśli wpisze prawidłowy i prawdziwy adres strony www.



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Działanie takie powoduje wpisanie przez użytkownika haseł, numerów kart bankowych i innych poufnych danych. Pharming przekierowuje użytkowników do fałszywych stron internetowych, bez wiedzy użytkownika Internetu. Po wpisaniu adresu jest on tłumaczony na adres IP za pośrednictwem serwera DNS. Przeglądarka internetowa łączy się z serwerem tego adresu IP i łączy daną stronę WWW. Wpis DNS dla tej witryny jest często przechowywany na komputerze użytkownika, w cache. W ten sposób komputer nie ma dostępu do serwera DNS i odwiedzając stronę internetową korzysta z tego przechowywanego w cache komputera. Komputer ofiary zostaje najczęściej zainfekowany tak, by lokalne pliki odpowiedzialne za rozpoznawanie nazw DNS zostały zmodyfikowane przez hakera. Można chronić się przed wirusami zmieniającymi pliki na komputerze, instalując program antywirusowy z aktualną bazą wirusów. W przypadku zaobserwowania znaczących różnic w wyglądzie strony internetowej, należy uruchomić ponownie komputer i ponownie połączyć się z tą samą witryną. Jeśli strona nadal będzie budziła podejrzenie, należy skontaktować się z dostawcą usług internetowych. Celem drugiej odmiany ataku typu „pharming” są serwery DNS należące do firmy lub usługodawcy internetowego. Haker musi włamać się do serwera DNS, z którego korzystają pracownicy firmy i przekierowywać połączenia, bez potrzeby włamywania się do ich komputerów. W tym przypadku użytkownik nie może nic zrobić: zapewnienie właściwej ochrony serwera należy do pracowników działu informatycznego firmy, lub usługodawcy internetowego.

Aby upewnić się, czy wyświetlana strona jest prawdziwa można sprawdzić jej certyfikat SSL. Aby się o tym przekonać wystarczy kliknąć symbol kłódki w pasku adresowym przeglądarki. W wyświetlonym oknie pojawi się informacja, kto jest właścicielem strony.

Zabezpieczanie przed niepożądanym i treściami

Można spotykać się również z próbami umieszczania na naszych stronach WWW niepożądanych treści. Istnieje wiele sposobów, dzięki którym można stwierdzić, że ktoś ma prawo wejścia, czyli, że jest autoryzowanym użytkownikiem danego serwisu. Jednym z nich jest system anti-robotowy, np. reCAPTCHA. Standardowe rozwiązania CAPTCHA wykorzystują losowo generowane znaki, które po pewnych zniekształceniach należy przepisać z ukazanego obrazka do pola tekstowego. Dopiero po poprawnym wpisaniu wygenerowanych znaków uzyskuje się dostęp do danego serwisu.



Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Postęp badań nad sieciami neuronowymi spowodował opracowanie systemów praktycznie bezbłędnie rozpoznających tekst. Z tego powodu postanowiono zmienić formę reCAPTCHA na numerację domów, znajdujących się na zdjęciach ulicy Street View. Przykładowy formularz zawiera wycięty numer ze zdjęcia oraz dla dodatkowej weryfikacji numer wygenerowany losowo.

Distorted words:

A "Distorted words" reCAPTCHA interface. It shows a small image of a house number "3435" from a Street View photo. Next to it is a large, randomly generated number "26645249" in a black, slightly distorted font. To the right are three blue square icons: a circular arrow for "Refresh", a speaker for "Listen", and a question mark for "Help". Below these elements is a white text input field with the placeholder text "Type the text above here.".

Źródło: Sławomir Szaruga, licencja: CC BY 3.0.

Taka weryfikacja pojawia się między innymi podczas zakładania poczty elektronicznej, aby chronić się przed nadmiernymi spamerami, rozsyłającymi niechciane e-maile.

Słowniczek

Biblioteka programistyczna

Jest to zestaw podprogramów oraz innych komponentów programistycznych, które mogą być wielokrotnie przywoływane w kodzie pisanego programu. Użycie bibliotek w znacznym stopniu upraszcza i przyspiesza.

Cytowanie piśmiennictwa

Jest to przytaczanie we własnym tekście cudzych zwrotów, powoływanie się na fakty naukowe, informacje zawarte w innych publikacjach. W naukach humanistycznych cytowanie bywa czasem określane terminem „powołania” [Narojczyk K., „Dokument elektroniczny i jego opis bibliograficzny w publikacjach humanistycznych”, Wyd. WM, Olsztyn 2005, strony: 70-74.].

Cytowanie to dosłowne przytoczenie fragmentu czyjejs wypowiedzi (cytat). Możemy też mówić o przytoczeniu w formie parafrazy. Takie cytowanie polega na omówieniu cytowanej wypowiedzi za pomocą własnych słów.

Grafik

Osoba zajmująca się tworzeniem elementów graficznych i projektowaniem wyglądu aplikacji. Osoba odpowiedzialna za grafikę powinna też przygotować obrazy w programie graficznym tak by miały optymalną rozdzielczość dla proponowanej aplikacji.

Gra komputerowa

„Zapisany w dowolnej postaci i na dowolnym nośniku cyfrowym (taśma, dyskietka, płyta, układy elektroniczne itp.) program komputerowy, spełniający funkcję ludyczną poprzez umożliwienie manipulacji generowanymi elektronicznie na ekranie wizyjnym (monitorze, telewizorze, wyświetlaczu ciekłokrystalicznym itp.) obiektami graficznymi lub tekstem, zgodnie z określonymi przez twórców gry regułami. W odróżnieniu od np. programów graficznych, gry służą wyłącznie celom rozrywkowym, a zatem nie spełniają żadnej użytkowej funkcji, umożliwiającej jakąkolwiek formę pracy twórczej”. „Łukasz Sikorski „Magia gier wirtualnych”. Warszawa 1998, s. 13”.

Graficzny interfejs użytkownika

Słowo interfejs, chcąc dokładnie przetłumaczyć na język polski, należałoby określić je jako styk, złącze, złączka. Dosłowne tłumaczenie pozwala nam intuicyjnie domyślić się znaczenia tego słowa w praktyce. Jest to złączka pomiędzy człowiekiem a komputerem, czyli sposób komunikowania się człowieka z komputerem. Interfejs graficzny do tej komunikacji, wykorzystuje obiekty wyświetlane na monitorze w trybie graficznym.

Interfejs działa w dwie strony – komputer komunikuje się z nami przez ekran a my z komputerem wprowadzając dane za pomocą klawiatury i myszy.

Graficzny interfejs użytkownika określa standardowy wygląd aplikacji oraz funkcjonalność obiektów służących za pomost komunikacyjny. Do podstawowych elementów interfejsu można zaliczyć:

1. ikony, oznaczające obiekty i polecenia (np. ikony w systemach Windows czy IOS);
2. okna wyświetlane na całym ekranie lub na jego części ;
3. menu poleceń (za pomocą rozwijanego menu, najczęściej w górze okna aplikacji lub w formie wstążki)

Graficzny interfejs użytkownika jest zwykle związany z konkretnym środowiskiem graficznym i korzysta z jego mechanizmów przy wyświetlaniu informacji.

Harmonogram

Uporządkowany w czasie porządek realizacji zadań w projekcie. Składa się z zadań, zależności między nimi, czasem trwania każdego z nich oraz alokacją potrzebnych zasobów.

IDE - ang. Integrated Development Environment

Zintegrowane środowisko programistyczne (ang. Integrated Development Environment, IDE) jest to środowisko służące do tworzenia oprogramowania. Zintegrowane środowisko programistyczne udostępnia złożoną funkcjonalność obejmującą edycję i kompilowanie kodu źródłowego. Dodatkowo dzięki IDE można w prosty sposób tworzyć elementy programu takie jak okna, ikony czy obsługę zdarzeń systemowych.

Processing rozwijany jest jako projekt Open Source. IDE objęte jest licencją LGPL, dzięki czemu wytworzone oprogramowanie z użyciem tego IDE może być dystrybuowane bez ograniczeń licencyjnych.

Kontrast

Jest to różnica, przeciwieństwo lub przeciwstawienie elementów kompozycji wizualnej. Kontrast pozwala przyciągnąć uwagę odbiorcy dzięki podkreśleniu i zaakcentowaniu wybranych elementów kompozycji. Odpowiednio zastosowany kontrast pozwala na wywołanie emocji, dzięki wytworzeniu hierarchii i wyznaczeniu obszarów dominujących. Ze względu na zestawione różnice rozróżniamy:

- kontrast walorowy,
- kontrast rozmiarowy,
- kontrast ciężkości,
- kontrast kształtu,
- kontrast położenia,
- kontrast teksturowy.

Kontrast jest jednym z podstawowych zabiegów kompozycyjnych pozwalających na wytworzenie interesującego, zrozumiałego i czytelnego komunikatu wizualnego.

Kreator treści

Osoba odpowiedzialna za treści umieszczone w aplikacji. Treści te powinny być wysokie jakościowo, unikalne, poprawne i oryginalne.

Multimedialny przewodnik geograficzny

Połączenie różnych sposobów przekazywania informacji (w postaci tekstu, dźwięku, grafiki, obrazów ruchomych i nieruchomych) za pośrednictwem komputera lub skomputeryzowanego sprzętu audiowizualnego (odtwarzacza płyt kompaktowych, magnetowidu, elektron. instrumentów muz.); np. wydawnictwa elektroniczne, gry, wideoklipy. „Encyklopedia PWN ”

Praca grupowa

W przypadku realizacji projektu w grupie ważnym staje się tzw. Praca grupowa. Dzięki współczesnym technologiom informatycznym osoby te nie muszą koniecznie znajdować się w jednym pomieszczeniu mogą nawet być rozproszone geograficznie i korzystając z mechanizmów przesyłania i współdzielenia dokumentów w sposób zdalny. W celu przeprowadzenia sprawnej komunikacji pomiędzy członkami grupy wykorzystamy jedno z dostępnych narzędzi do pracy grupowej (szerzej zostanie omówione w rozdziale o chmurze).

Processing

Processing a właściwie Processing.org, to język programowania oraz środowisko programistyczne powstałe w 2001 roku. Jego celem było umożliwienie artystom i twórcom sztuki wizualnej szybkiego prototypowania oraz tworzenia prac w łatwy i sprawny sposób nawet bez wcześniejszej znajomości jakiegokolwiek języka programowania. Processing to tak naprawdę „opakowanie” języka Java, a zwłaszcza metod dotyczących wyświetlania, tworzenia i modyfikowania grafiki dwu- jak i trójwymiarowej. Z założenia ma minimalizować nakład pracy programistycznej, tak by twórca mógł skupić się na aspekcie wizualnym i funkcjonalnym swojej pracy, nie natomiast na elementach technicznych. Obecnie język ten jest dosyć dobrze rozwinięty i świetnie udokumentowany, dzięki czemu coraz więcej osób sięga po niego tworząc projekty komercyjne oraz artystyczne.

Programista

Osoba, która tworzy programy komputerowe w określonym języku programowania.
soba, która tworzy programy komputerowe w określonym języku programowania.

Projektant systemu

Osoba odpowiedzialna za nadzorowanie procesu wdrażania systemu informatycznego, zapewniając zgodność z założeniami. Funkcja ta jest jednym z kluczowych stanowisk, niezbędnych w każdym zespole projektowym.

Styl

Zestaw cech formatowania, takich jak nazwa, rozmiar i kolor czcionki, wyrównanie akapitów i odstępy między akapitami. Niektóre style zawierają nawet obramowania i cieniowanie.

Symulator

Urządzenie do odtwarzania warunków rzeczywistych, w celu szkolenia i treningu pilotów, astronautów, kierowców samochodów, czołgistów, strzelców, operatorów radarowych i innych. „Encyklopedia PWN ”

Webdesigner

Osoba odpowiedzialna za rozplanowanie, zaprojektowanie i wdrożenie strony internetowej.

Wykres Gantt'a

Jest to graf stosowany w zarządzaniu projektami który uwzględnia podział projektu na poszczególne zadania, oraz rozplanowanie ich w czasie.

Wykres taki można zbudować samodzielnie wykorzystując arkusz kalkulacyjny lub skorzystać z gotowych wzorców proponowanych przez szablony programu.

Pojęcia

Cytowanie piśmiennictwa

Cytowanie jest to przytaczanie we własnym tekście cudzych zwrotów, powoływanie się na fakty naukowe, informacje zawarte w innych publikacjach. W naukach humanistycznych cytowanie bywa czasem określane terminem „powołania”[Narojczyk K., „Dokument elektroniczny i jego opis bibliograficzny w publikacjach humanistycznych”, Wyd. WM, Olsztyn 2005, strony: 70-74.].

Cytowanie to dosłowne przytoczenie fragmentu czyjejś wypowiedzi (cytat). Możemy też mówić o przytoczeniu w formie parafrazy. Takie cytowanie polega na omówieniu cytowanej wypowiedzi za pomocą własnych słów.

Graficzny interfejs użytkownika

Słowo interfejs, chcąc dokładnie przetłumaczyć na język polski, należałoby określić je jako styk, złącze, złączka. Dosłowne tłumaczenie pozwala nam intuicyjnie domyślić się znaczenia tego słowa w praktyce. Jest to złączka pomiędzy człowiekiem a komputerem, czyli sposób komunikowania się człowieka z komputerem. Interfejs graficzny do tej komunikacji, wykorzystuje obiekty wyświetlane na monitorze w trybie graficznym. Interfejs działa w dwie strony – komputer komunikuje się z nami przez ekran a my z komputerem wprowadzając dane za pomocą klawiatury i myszy.

Graficzny interfejs użytkownika określa standardowy wygląd aplikacji oraz funkcjonalność obiektów służących za pomost komunikacyjny. Do podstawowych elementów interfejsu można zaliczyć:

1. ikony, oznaczające obiekty i polecenia (np. ikony w systemach Windows czy IOS);
2. okna wyświetlane na całym ekranie lub na jego części ;
3. menu poleceń (za pomocą rozwijanego menu, najczęściej w górze okna aplikacji lub w formie wstążki)

Graficzny interfejs użytkownika jest zwykle związany z konkretnym środowiskiem graficznym i korzysta z jego mechanizmów przy wyświetlaniu informacji.

Gra komputerowa

„zapisany w dowolnej postaci i na dowolnym nośniku cyfrowym (taśma, dyskietka, płyta, układy elektroniczne itp.) program komputerowy, spełniający funkcję ludyczną poprzez umożliwienie manipulacji generowanymi elektronicznie na ekranie wizyjnym (monitorze, telewizorze, wyświetlaczu ciekłokrystalicznym itp.) obiektami graficznymi lub tekstem, zgodnie z określonymi przez twórców gry regułami. W odróżnieniu od np. programów graficznych, gry służą wyłącznie celom rozrywkowym, a zatem nie spełniają

żadnej użytkowej funkcji, umożliwiającej jakąkolwiek formę pracy twórczej” (Łukasz Sikorski „Magia gier wirtualnych”. Warszawa 1998, s. 13).

Harmonogram

Harmonogram to uporządkowany w czasie porządek realizacji zadań w projekcie. Składa się z zadań, zależności między nimi, czasem trwania każdego z nich oraz alokacją potrzebnych zasobów.

IDE - ang. Integrated Development Environment

Zintegrowane środowisko programistyczne (ang. Integrated Development Environment, IDE) jest to środowisko służące do tworzenia oprogramowania. Zintegrowane środowisko programistyczne udostępnia złożoną funkcjonalność obejmującą edycję i kompilowanie kodu źródłowego. Dodatkowo dzięki IDE można w prosty sposób tworzyć elementy programu takie jak okna, ikony czy obsługę zdarzeń systemowych.

Processing rozwijany jest jako projekt Open Source. IDE objęte jest licencją LGPL, dzięki czemu wytworzone oprogramowanie z użyciem tego IDE może być dystrybuowane bez ograniczeń licencyjnych.

Kontrast

Kontrast jest to różnica, przeciwieństwo lub przeciwstawienie elementów kompozycji wizualnej. Kontrast pozwala przyciągnąć uwagę odbiorcy dzięki podkreśleniu i zaakcentowaniu wybranych elementów kompozycji. Odpowiednio zastosowany kontrast pozwala na wywołanie emocji, dzięki wytworzeniu hierarchii i wyznaczeniu obszarów dominujących. Ze względu na zestawione różnice rozróżniamy:

- Kontrast walorowy,
- Kontrast rozmiarowy,
- Kontrast ciężkości,
- Kontrast kształtu,
- Kontrast położenia,
- Kontrast teksturowy.

Kontrast jest jednym z podstawowych zabiegów kompozycyjnych pozwalających na wytworzenie interesującego, zrozumiałego i czytelnego komunikatu wizualnego.

Multimedialny przewodnik geograficzny

MULTIMEDIA [łac.], połączenie różnych sposobów przekazywania informacji (w postaci tekstu, dźwięku, grafiki, obrazów ruchomych i nieruchomych) za pośrednictwem komputera lub skomputeryzowanego sprzętu audiowizualnego (odtwarzacza płyt kompaktowych, magnetowidu, elektron. instrumentów muz.); np. wydawnictwa elektron., gry, wideoklipy.(encyklopedia pwn)

Praca grupowa

W przypadku realizacji projektu w grupie ważnym staje się tzw. Praca grupowa. Praca grupowa polega na pracy wielu osób. Dzięki współczesnym technologiom informatycznym osoby te nie muszą koniecznie znajdować się w jednym pomieszczeniu, mogą nawet być rozproszone geograficznie i korzystając z mechanizmów przesyłania i współdzielenia dokumentów w sposób zdalny.

Bardzo ważna w pracy grupowej jest inicjatywa, obowiązkowość i efektywność wszystkich uczestników grupy.

W celu przeprowadzenia sprawnej komunikacji pomiędzy członkami grupy wykorzystamy jedno z dostępnych narzędzi do pracy grupowej (szerzej zostanie omówione w rozdziale o chmurze).

Processing

Processing a właściwie Processing.org to język programowania oraz środowisko programistyczne powstałe w 2001 roku. Jego celem było umożliwienie artystom i twórcom sztuki wizualnej szybkiego prototypowania oraz tworzenia prac w łatwy i sprawny sposób nawet bez wcześniejszej znajomości jakiegokolwiek języka programowania. Processing to tak naprawdę „opakowanie” języka Java, a zwłaszcza metod dotyczących wyświetlania, tworzenia i modyfikowania grafiki dwu- jak i trójwymiarowej. Z założenia ma minimalizować nakład pracy programistycznej, tak by twórca mógł skupić się na aspekcie wizualnym i funkcjonalnym swojej pracy, nie natomiast na elementach technicznych. Obecnie język ten jest dosyć dobrze rozwinięty i świetnie udokumentowany, dzięki czemu coraz więcej osób sięga po niego tworząc projekty komercyjne oraz artystyczne.

Styl

Styl to zestaw cech formatowania, takich jak nazwa, rozmiar i kolor czcionki, wyrównanie akapitów i odstępy między akapitami. Niektóre style zawierają nawet obramowania i cieniowanie.

Symulator

symulator [łac.], urządzenie do odtwarzania warunków rzeczywistych w celu szkolenia i treningu pilotów, astronautów, kierowców samochodów, czołgistów, strzelców, operatorów radarowych i in.(encyklopedia pwn)

Wykres Gantt'a

Jest to graf stosowany w zarządzaniu projektami który uwzględnia podział projektu na poszczególne zadania, oraz rozplanowanie ich w czasie.

Wykres taki można zbudować samodzielnie wykorzystując arkusz kalkulacyjny lub skorzystać z gotowych wzorców proponowanych przez szablony programu.