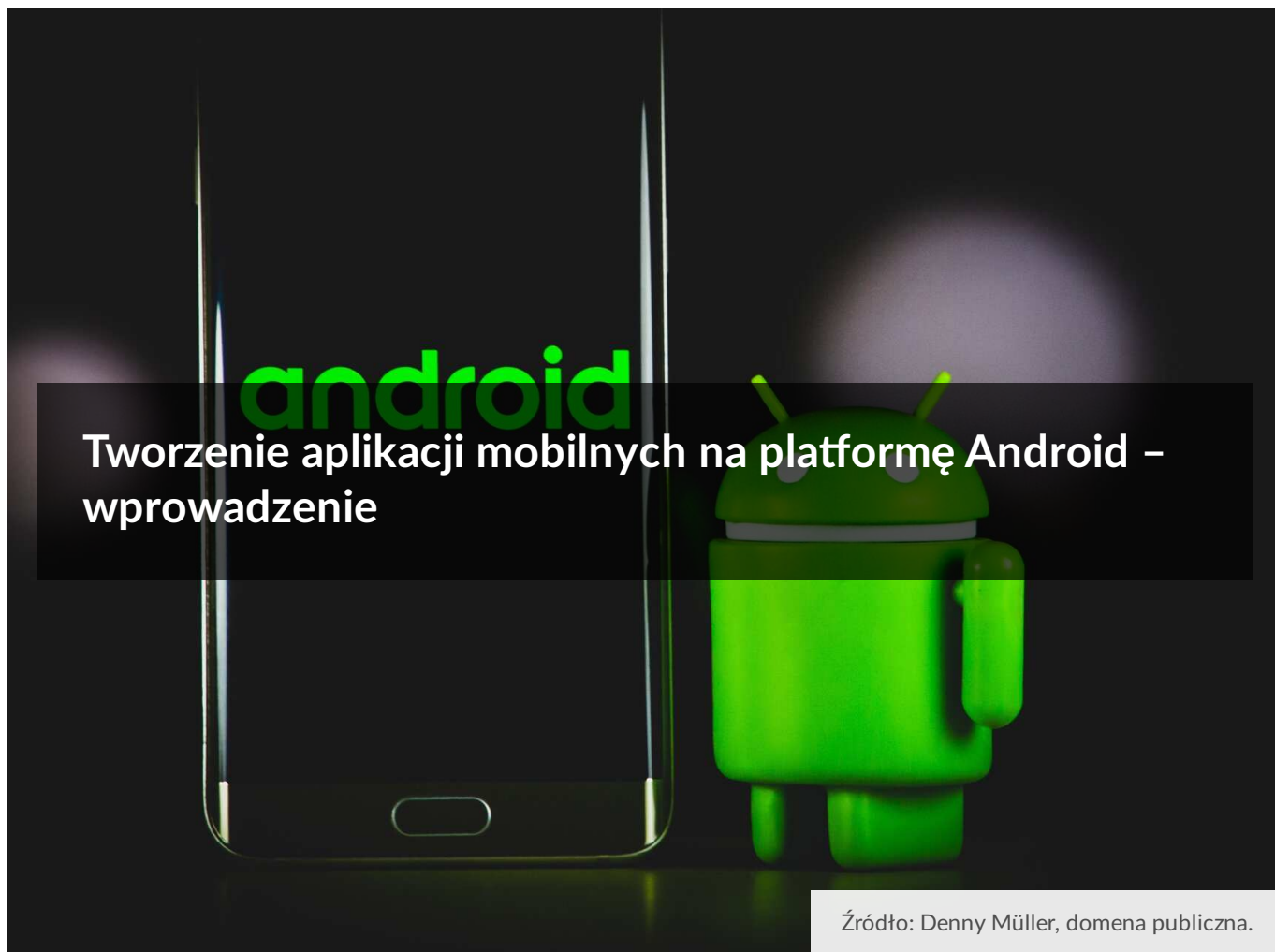




Tworzenie aplikacji mobilnych na platformę Android – wprowadzenie

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Symulacja interaktywna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Zastanów się, jakie są najpopularniejsze systemy operacyjne na urządzeniach mobilnych. Zapewne wymienisz wśród nich **Android**. Jeśli korzystasz z tego systemu, wiesz, że w sklepie z aplikacjami można znaleźć nie tylko te stworzone przez wielkie firmy, ale i przez pojedyncze osoby. Ty też jesteś w stanie opracować własną aplikację, która może znaleźć się w takim sklepie.

W tym e-materiale dowiesz się, jak to zrobić. Najpierw jednak zapoznasz się z istotą działania aplikacji mobilnych, najważniejszymi pojęciami oraz narzędziami służącymi do stworzenia takiego projektu. Na koniec wyjaśnimy, w jaki sposób zacząć pracę nad tworzeniem własnej aplikacji mobilnej dla systemu operacyjnego Android.

Twoje cele

- Przeanalizujesz najważniejsze definicje elementów aplikacji mobilnej.
- Zapoznasz się z różnymi narzędziami służącymi do stworzenia aplikacji mobilnej.
- Przedstawisz proces tworzenia aplikacji mobilnych.

Przeczytaj

W listopadzie 2007 roku opublikowano pierwszą wersję Android SDK. W jej skład weszły następujące narzędzie programistyczne:

- debbuger (program służący do dynamicznej analizy innych programów, w celu znalezienia i określenia zawartości ich błędów, tzw. bugów),
- biblioteki (plik dostarczający dane, podprogramy oraz typy danych, z których można korzystać na poziomie kodu źródłowego programu, dzięki bibliotekom wielokrotnie używamy tego samego kodu),
- emulator (program, który po uruchomieniu w danym systemie komputerowym duplikuje funkcje innego systemu komputerowego: pierwszy system to *host* – gospodarz, a drugi, emulowany przez pierwszy, nazywany jest *guest* – gość),
- dokumentacja (ogół dokumentów stworzony dla określonego programu komputerowego, dokumentację dzielimy na techniczną i dokumentację użytkownika),
- tutoriale (samouczki ukazujące krok po kroku, jak nauczyć się obsługi różnych urządzeń, programowania czy tworzenia wybranych plików; tutoriale są interaktywne, dokładniejsze niż instrukcje, bo nastawione na naukę na przykładzie),
- FAQ (ang. *frequently asked questions* – zbiór krótkich porad i odpowiedzi na najczęściej pojawiające się pytania związane z obsługą systemu itp.).



Jedno z pierwszych urządzeń wyposażonych w system Android

Źródło: Kai Hendry, dostępny w internecie:

commons.wikimedia.org [dostęp 15.08.2022], licencja: CC BY 2.0.

Programiści, instalując Android SDK, mają do uwzględnienia określone parametry sprzętowe, ale mogą też korzystać z narzędzi konsolowych w celu tworzenia i debugowania aplikacji na Androida. Pojawiły się zarzuty, że system nie działa poprawnie, brakuje w nim opcji oficjalnego raportowania błędów, infrastruktura jakości nie jest spójna, a dostęp do materiałów technicznych – utrudniony. Pomimo tych uwag, tuż po upowszechnieniu Androida, do użytku trafiała prosta gra o nazwie *Wąż* (ang. *Snake*). Gracz ma za zadanie zjeść jak najwięcej (co sprawi, że wąż będzie coraz dłuższy), a przy tym nie uderzyć w ścianę ani w siebie samego.



Logo Androida w latach 2007-2014

Źródło: Android, dostępny w internecie: android.com [dostęp 15.08.2022], domena publiczna.



Logo Androida w latach 2014-2015

Źródło: Android, dostępny w internecie: android.com [dostęp 15.08.2022], domena publiczna.



Logo Androida w latach 2015-2019

Źródło: Android, dostępny w internecie: android.com [dostęp 15.08.2022], domena publiczna.



Logo Androida od 2019 roku

Źródło: Android, dostępny w internecie: android.com [dostęp 15.08.2022], domena publiczna.

Projektowanie aplikacji mobilnych

Co trzeba wziąć pod uwagę, tworząc aplikacje mobilne dla systemu operacyjnego Android? Należy pamiętać, że współpracuje on z urządzeniami mobilnymi, których zasoby są ograniczone, np. przez wydajność procesora lub rozmiar pamięci RAM.

Dodatkowo projektowanie aplikacji działających na urządzeniach z systemem operacyjnym Android wymaga analizy funkcjonalnej proponowanego rozwiązania oraz uwzględnienia specyfikacji warunków, w jakich uruchamiana jest ta aplikacja. Jednym z ważniejszych aspektów jest też sposób komunikacji aplikacji z różnymi podzespołami, w jakie wyposażone jest urządzenie, np. aparat, żyroskop czy GPS.

Przede wszystkim jednak efektywne projektowanie aplikacji, zgodne z założeniami systemu operacyjnego Android, wymaga poznania podstawowych pojęć.

Podstawowe komponenty aplikacji mobilnej

Funkcjonowanie aplikacji mobilnej obejmuje kilka podstawowych elementów składowych. Poznajmy ich definicje.

Kontekst (obiekt typu Context)

Kontekst jest środowiskiem, w którym uruchamiana jest zaprojektowana aplikacja mobilna. Za jego pośrednictwem można zarządzać aplikacją i uzyskiwać dostęp do parametrów aplikacji oraz urządzenia, na którym została uruchomiona.

Aktywność (obiekt typu Activity)

Aktywności to podstawowe elementy, z których tworzone są zaplanowane funkcjonalności aplikacji mobilnej. Oznacza to, że wywoływanie funkcji aplikacji mobilnej odbywa się przez uruchomienie aktywności. W danym zachowaniu można wywoływać kolejne aktywności, dzięki czemu uzyskujemy możliwość realizacji nawet bardzo złożonych scenariuszy. Podstawową właściwością aktywności jest krótki czas jej wykonania oraz kontakt z użytkownikiem aplikacji lub z innymi aplikacjami.

Intencja (obiekt typu Intent)

Intencje stanowią mechanizm obsługi zdarzeń występujących w aplikacji mobilnej (komunikaty asynchroniczne). Z wykorzystaniem mechanizmu intencji możliwe jest uruchomienie aktywności (także w innej aplikacji) lub serwisu w momencie wystąpienia określonego zdarzenia. Mechanizm intencji służy także do komunikacji między różnymi aplikacjami uruchomionymi na urządzeniu mobilnym oraz do obsługi zdarzeń pochodzących z innych aplikacji (np. połączenie przychodzące) i wyposażenia urządzenia (np. niski stan baterii, wykonano zdjęcie).

Usługa/Serwis (obiekt typu Service)

Usługi można traktować jako specyficzne zastosowanie aktywności. Zawierają one implementację funkcji, nie wykorzystują interfejsu komunikacji z użytkownikiem oraz wymagają długiego czasu działania. Może to wynikać z czasochłonnych algorytmów obliczeniowych lub z potrzeby dostępu do baz danych (w tym też tych zewnętrznych) i konieczności wysłania lub pobrania dużej ilości danych. Cechą charakterystyczną usługi

jest działanie „w tle”. W postaci usługi często implementuje się funkcje działające cyklicznie.

Dostawcy treści (obiekt typu **ContentProvider**)

Dostawcy treści zapewniają dostęp do swoich zasobów innym aplikacjom zainstalowanym na urządzeniu. Udostępniane dane mogą być przechowywane przez dostawcę w bazie danych SQLite lub w innej trwałej postaci. Dostawcy treści mogą też być związani komponentami sprzętowymi zainstalowanymi w urządzeniu, takimi jak GPS.

Odbiorcy komunikatów (obiekt typu **BroadcastReceiver**)

Odbiorcy komunikatów umożliwiają komunikację z innymi aplikacjami dzięki otrzymywaniu konkretnych komunikatów wysyłanych przez inne aplikacje zainstalowane na urządzeniu.

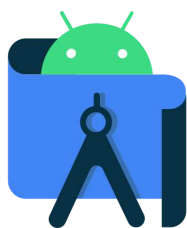
Przyjrzyjmy się dwóm przykładowym implementacjom własnych komunikatów:

- Pierwsza aplikacja jest nadawcą komunikatów. Rozesłanie komunikatu możliwe jest z wykorzystaniem przycisku lub zadania cyklicznego. Do rozsyłania komunikatów wykorzystana została metoda `sendBroadcast(intent)`.
- Druga aplikacja, obok głównej aktywności, posiada klasę definiującą odbiorcę komunikatów: `MojOdbiorcaKomunikatow`, [dziedziczącą](#) po klasie `BroadcastReceiver`. Klasa `MojOdbiorcaKomunikatow` musi być zadeklarowana w pliku `AndroidManifest.xml`, z podaniem rodzaju komunikatu, na jaki ma reagować (**filtr intencji**). W przykładzie zdefiniowano własny typ komunikatu:

```
1 com.example.komunikatynadawca.MOJ_KOMUNIKAT.
```

Przygotowanie środowiska programistycznego

android studio



Źródło: [™]/_@Google LLC, dostępny w internecie:
developer.android.com [dostęp 15.08.2022], licencja: CC BY
2.5.

Poznaliśmy już podstawowe pojęcia związane z projektowaniem aplikacji mobilnych. Teraz przeanalizujemy krok po kroku, jak prawidłowo **zainstalować** Android Studio.

1. Przechodzimy na stronę internetową programu Android Developers.
2. Pobieramy wersję instalacyjną odpowiednią dla używanego systemu operacyjnego. W tym celu możemy wybrać opcję `Download options`.
3. Postępujemy zgodnie z informacjami wyświetlanymi w trakcie instalacji (dobrze jest wybrać obszar instalacji Android Studio na partycji dysku, która ma większą

pojemność).

4. Usuwamy plik instalacyjny, gdyż nie będzie już potrzebny.

Teraz możemy po raz pierwszy **uruchomić** Android Studio.

1. Na ekranie Welcome klikamy opcję Next .
2. Następnie mamy do wyboru opcje Standard i Custom. W większości przypadków wystarczy opcja Standard, jednak dobrze jest wybrać opcję Custom.
3. Następnie wybieramy miejsce instalacji **Java Development Kit (JDK)**.
Wskazujemy partycję, na której zainstalowaliśmy Android Studio.
4. W kolejnym kroku wybieramy sposób wyświetlania programu Android Studio – mamy do wyboru jasny (*light*) oraz ciemny (*dark*).
5. Następnie program prosi o wybór Android **SDK** i wskazanie miejsca instalacji – podobnie jak podczas instalacji Android Studio dobrze jest wybrać partycję z większą pojemnością. Wybieramy: Android SDK oraz API 30 i Performance Intel Haxm, dzięki któremu emulatorzy Androida potrzebne do testowania programów będą uruchamiać się szybciej i sprawnie odpowiadać na nasze działania.
6. Następnie wyznaczamy ilość pamięci RAM, jaką jesteśmy w stanie przeznaczyć na emulator Androida. Sugerowana wartość to recommended, ale jeśli mamy taką możliwość, możemy wybrać większą wartość.
7. Po przejściu opisanych kroków rozpoczyna się proces instalowania wymaganych komponentów do poprawnego działania Android Studio. Może to potrwać kilka minut.
8. Jeśli instalacja wymaganych komponentów zakończy się pomyślnie, pojawi się okno z opcjami do wyboru. Jest wśród nich możliwość stworzenia nowego projektu itp.

Słownik

Android Debug Bridge

interfejs do komunikacji z urządzeniem; ADB ułatwia wykonywanie różnych akcji na urządzeniu, takich jak instalowanie i debugowanie aplikacji

Android Package Kit

odmiana formatu JAR; używany jest do dystrybucji i instalacji pakietów na system operacyjny Android

Dalvik Executable

pliki używane są do inicjowania oraz uruchamiania aplikacji dla platformy Android

dziedziczenie

w programowaniu obiektowym mechanizm współdzielenia atrybutów oraz zachowań pomiędzy klasą bazową a jej pochodną; dzięki dziedziczeniu niektórych

zaimplementowanych rozwiązań można z nich wielokrotnie korzystać w aplikacji
Java Development Kit

darmowe oprogramowanie, udostępniające środowisko niezbędne do programowania
w języku Java

Software Development Kit

zestaw narzędzi niezbędny przy tworzeniu aplikacji korzystającej z określonej biblioteki
dla danego systemu, sprzętu itp.

Symulacja interaktywna

Polecenie 1

Pierwsze telefony komórkowe miały bardzo ograniczone możliwości, jeśli chodzi o liczbę wykonywanych operacji – wynikało to z ograniczeń procesorów. Postęp technologiczny sprawił, że dziś telefony służą nie tylko komunikacji, ale i rozrywce. Wpływ na to miało również zmniejszenie się kosztów produkcji tak wydajnych podzespołów, a w szczególności procesorów. Zapoznaj się z symulacją interaktywną ilustrującą, jak zmieniał się koszt (podany w dolarach amerykańskich) jednego gigaflopsa (jeden flops to jedna operacja zmiennoprzecinkowa na sekundę, zatem 1 GFLOPS to 10^9 operacji zmiennoprzecinkowych na sekundę).

Spróbuj uzupełnić symulację o aktualne dane.

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Polecenie 2

Zastanów się, jakie skutki technologiczne i społeczne mogła mieć ta zmiana.

Polecenie 3

Oblicz, o ile procent tańszy jest przybliżony koszt jednego GFLOPSa w roku 2020 w porównaniu do roku 2000.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Uzupełnij zdanie właściwym fragmentem.

Emulator to program, który po uruchomieniu w danym systemie komputerowym duplikuje funkcje innego systemu komputerowego: pierwszy system to - , a drugi, emulowany przez pierwszy, nazywany jest - .

home

duch

gospodarz

host

guest

root

ghost

gość

dom

Ćwiczenie 2



Wskaż, co oznacza skrót JDK.

☐ Java Development Kit

☐ Java Debugging Kit

☐ Java Design Kit

Ćwiczenie 3



Wskaż, co stanowi mechanizm obsługi zdarzeń występujących w aplikacji mobilnej.

☐ serwis

☐ aktywność

☐ intencja

Ćwiczenie 4

Wyjaśnij, czym są flopsy (FLOPS).



Ćwiczenie 5



Wskaż dokończenie zdania.

Komponent aplikacji mobilnej, za którego pośrednictwem można uruchomić aplikację, zarządzać nią i uzyskiwać dostęp do parametrów aplikacji oraz urządzenia, na którym została uruchomiona to...

☐ dostawca treści.

☐ kontekst.

☐ odbiorca komunikatów.

☐ serwis.

Ćwiczenie 6



Wskaż, które komponenty aplikacji mobilnej systemu Android są specjalnymi klasami, umożliwiającymi komunikację z innymi aplikacjami dzięki otrzymywaniu konkretnych komunikatów, wysyłanych przez inne aplikacje zainstalowane na urządzeniu.

aktywności ☐

/ serwisy ☐

/ dostawcy treści ☐

/ odbiorcy komunikatów ☐

Ćwiczenie 7



Wskaż, co przyspieszy działanie emulatora Androida, jeśli mamy komputer z procesorem Intel.

☐ Intel UHD Graphics

☐ Android SDK

☐ Intel HAXM

Ćwiczenie 8



Przyporządkuj definicje do odpowiednich pojęć.

Android Package Kit

darmowe oprogramowanie,
udostępniające środowisko niezbędne
do programowania w języku Java

Java Development Kit

odmiana formatu JAR, używany jest do
dystrybucji i instalacji pakietów na
system operacyjny Android

Software Development Kit

interfejs do komunikacji z urządzeniem,
ułatwia wykonywanie różnych akcji na
urządzeniu, takich jak instalowanie
i debugowanie aplikacji

Android Debug Bridge

zestaw narzędzi niezbędny przy
tworzeniu aplikacji korzystającej
z określonej biblioteki dla danego
systemu, sprzętu itp.

Davlik Executable

pliki używane są do inicjowania oraz
uruchamiania aplikacji dla platformy
Android

Dla nauczyciela

Autor: Łukasz Łopuszański

Przedmiot: Informatyka

Temat: Tworzenie aplikacji mobilnych na platformę Android – wprowadzenie

Grupa docelowa:

Szkoła ponadpodstawowa, liceum ogólnokształcące, technikum, zakres rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi, w tym: znajomość zasad działania urządzeń cyfrowych i sieci komputerowych oraz wykonywania obliczeń i programów.

Treści nauczania – wymagania szczegółowe

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

1) projektuje rozbudowę i zakup nowego zestawu komputerowego oraz oprogramowania;

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Przeanalizujesz najważniejsze definicje elementów aplikacji mobilnej.
- Zapoznasz się z różnymi narzędziami służącymi do stworzenia aplikacji mobilnej.
- Przedstawisz proces tworzenia aplikacji mobilnych.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Tworzenie aplikacji mobilnych na platformę Android – wprowadzenie”. Uczniowie zapoznają się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Przedstawienie tematu zajęć oraz wspólne z uczniami ustalenie kryteriów sukcesu.
2. Prowadzący prosi uczniów, aby zgłaszali swoje propozycje pytań do tematu. Jedna osoba może zapisywać je na tablicy. Gdy uczniowie wyczerpią swoje pomysły, a pozostały jakieś ważne kwestie do poruszenia, nauczyciel je dopowiada.

Faza realizacyjna:

1. **Praca z tekstem.** Nauczyciel prosi wybraną osobę o podsumowanie najważniejszych informacji z sekcji „Przeczytaj”. Pozostali uczniowie w razie potrzeby uzupełniają nieporuszone kwestie. Jeśli zaistnieją jakieś wątpliwości, nauczyciel wyjaśnia je.
2. **Praca z multimediami.** Nauczyciel wyświetla zawartość sekcji „Symulacja interaktywna”. Uczniowie wspólnie zapoznają się z jej treścią. Zapisują ewentualne problemy i pytania. Po czym następuje dyskusja, w trakcie której nauczyciel wyjaśnia niezrozumiałe treści.
3. **Ćwiczenie umiejętności.** W kolejnym etapie uczniowie indywidualnie wykonują ćwiczenia 1-8 z sekcji „Sprawdź się”. Następnie konsultują swoje rozwiązanie z inną osobą i ustalają jedną wersję odpowiedzi.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat i cele lekcji zawarte w sekcji „Wprowadzenie”. W kontekście ich realizacji następuje omówienie ewentualnych problemów z rozwiązaniem ćwiczeń z sekcji „Sprawdź się”.
2. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności, omawia ewentualne problemy podczas rozwiązania ćwiczeń.

Praca domowa:

1. Wykonaj polecenia z sekcji „Symulacja interaktywna”.

Wskazówki metodyczne:

- Lekcja może być wprowadzeniem do projektu grupowego. Uczniowie mogą zaplanować prostą aplikację, przygotować dla niej dokumentację, diagramy etc.