



Reprezentacja tekstu w komputerze

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Gra edukacyjna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Reprezentacja tekstu w komputerze

Źródło: Siora Photography, domena publiczna.

W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

To, co widzimy na monitorze komputera czy na ekranie smartfonu, stanowi wynik końcowy wielu procesów i operacji. Każdy ciąg znaków, zdjęcie czy film są zakodowane według określonych reguł oraz konwencji. To ciąg bitów i bajtów, czyli zer i jedynek.

W tym e-materiale dowiesz się więcej na temat sposobów reprezentowania informacji w komputerach.

Twoje cele

- Wyjaśnisz, w jaki sposób kodowany jest tekst w komputerze.
- Przeanalizujesz metody kodowania obrazów i animacji.
- Rozwiążesz ćwiczenia dotyczące sposobów reprezentowania informacji w komputerach.

Przeczytaj

Systemy kodowania znaków

Istnieje wiele systemów kodowania znaków. Omówimy te najbardziej znane i najczęściej wykorzystywane – **ASCII**, **ISO**, **Unicode**.

ASCII

ASCII, z angielskiego *American Standard Code for Information Interchange*, to jeden z najbardziej znanych systemów kodowania znaków. W podstawowej wersji umożliwia on zakodowanie 128 znaków. System ten polega na tym, że znakom przypisuje się liczby z zakresu od 0 do 127, a w rozszerzonej wersji z zakresu od 0 do 255.

Podstawowa wersja

Przypisania znaków do liczb:

- od 0 do 32 – liczbom tym przypisane są tzw. **znaki sterujące**, czyli m.in. *Escape*, *Backspace*, *Shift* itd.;
- od 33 do 47 – zakres ten obejmuje znaki specjalne, w ich skład wchodzi m.in. *!*, *#*, *%*, *\$*;
- od 48 do 57 – liczbom z tego zakresu przypisane są cyfry (jako znaki) – 0, 1, 2, 3, 4, 5, 6, 7, 8, 9;
- od 65 do 90 – tym liczbom przypisane są **wielkie** litery alfabetu łacińskiego;
- od 97 do 122 – obejmuje **małe** litery alfabetu łacińskiego.

Poniższa tabela obrazuje kodowania znaków z zakresu od 32 do 127.

32: SPC	33: !	34: "	35: #	36: \$
37: %	38: &	39: '	40: (	41:)
42: *	43: +	44: ,	45: -	46: .
47: /	48: 0	49: 1	50: 2	51: 3
52: 4	53: 5	54: 6	55: 7	56: 8
57: 9	58: :	59: ;	60: <	61: =
62: >	63: ?	64: @	65: A	66: B
67: C	68: D	69: E	70: F	71: G
72: H	73: I	74: J	75: K	76: L
77: M	78: N	79: O	80: P	81: Q
82: R	83: S	84: T	85: U	86: V
87: W	88: X	89: Y	90: Z	91: [
92: \	93:]	94: ^	95: _	96: `
97: a	98: b	99: c	100: d	101: e
102: f	103: g	104: h	105: i	106: j
107: k	108: l	109: m	110: n	111: o
112: p	113: q	114: r	115: s	116: t
117: u	118: v	119: w	120: x	121: y
122: z	123: {	124:	125: }	126: ~
127: DEL				

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

Rozszerzona wersja

Z czasem kodowanie zaledwie 128 znaków przestało wystarczać – system ten nie obejmował chociażby kodowania znaków narodowych, tj. „Ą”, „Ę”... W związku z tym powstały nowe systemy kodowania znaków, które w pewnej części oparte były na istniejącym już ASCII. Były to systemy kodowania ISO-8859.

ISO-8859

W porównaniu do ASCII nie jest to 7-bitowy, a 8-bitowy system kodowania znaków. Dodatkowy bit pozwala na zakodowanie 128 znaków więcej.

Część wspólna systemu kodowania ISO i ASCII to pierwsze 128 znaków – przypisania od 0 do 127 – w ISO-8859 są one identyczne jak ASCII.

Drugiej połowie, czyli liczbom od 128 do 255 przypisane są znaki narodowe. To jednak również nie wystarczało, ponieważ każdy język ma swoje unikalne znaki. Dodatkowe 128 możliwości kodowań nadal nie pozwalało na zakodowanie wszystkich potrzebnych znaków. W związku z tym powstało wiele norm ISO, które obejmowały różne grupy językowe. Oto kilka z nich:

- **ISO-8859-1** – znaki alfabetu łacińskiego (Europa Zachodnia),
- **ISO-8859-2** – znaki alfabetu łacińskiego (Europa Środkowa i Wschodnia),
- **ISO-8859-5** – znaki rosyjskie (cyrylica),
- **ISO-8859-7** – znaki alfabetu greckiego.

Standard **ISO-8859-2** obejmuje polskie znaki.

Największą wadą takiego rozwiązania był fakt, że jeżeli chcielibyśmy użyć (np. podczas pisania dokumentu) zarówno znaków polskich (norma ISO-8859-2) jak i znaków rosyjskiej cyrylicy (norma ISO-8859-5) to nie było to możliwe lub nie działało poprawnie. W związku z tym powstał kolejny system kodowania, najbardziej obszerny – **Unicode**.

Unicode

System ten wychodzi naprzeciw powyższym problemom – daje możliwość kodowania wszystkich znaków narodowych. Jeżeli piszemy dokument w standardzie Unicode możemy użyć praktycznie dowolnego znaku.

Ważne!

Warto wspomnieć, że pierwsze 128 kodowań pokrywa się z tabelą ASCII.

Aktualną wersją Unicode (lipiec 2021) jest **Unicode 13.0** i ma on zdefiniowane 143859 różnych kodowań znaków. Niesie to ze sobą pewną wadę tego systemu. Dotychczas omówione systemy kodowań – ASCII i ISO-8859 – umożliwiały kolejno kodowanie 128 i 256 różnych znaków. W Unicode możliwe jest kodowanie tak wielu znaków, ponieważ wykorzystywane jest więcej pamięci. Niektóre znaki są kodowane nawet za pomocą 4 bajtów.

Istnieją różne metody kodowania Unicode, najczęściej używanymi są:

- UTF-8,
- UTF-16,
- UTF-32.

Różnią się one m.in. liczbą używanych bitów (z powyżej wymienionych metod, liczby znajdujące się na końcu nazwy, oznaczają liczbę wykorzystywanych bitów).

Przykład 1

Imię **Mateusz** przekonwertowane na kod UTF-8 przedstawiony za pomocą szesnastkowego systemu liczbowego prezentować będzie się następująco:

4D 61 74 65 75 73 7A 0A

Konwertując dodatkowo system szesnastkowy na binarny otrzymamy następujący ciąg bitów:

100110101100001011101000110010101110101011100110111101000001010

Kodowanie animacji

Najpopularniejszym formatem, w jakim zapisywane są animacje jest format GIF (*ang. Graphics Interchange Format*). Umożliwia on zapis wielu obrazków w jednym pliku. Zapętlona sekwencja zapisanych obrazków tworzy **animację**. Oto przykład animacji w formacie GIF:

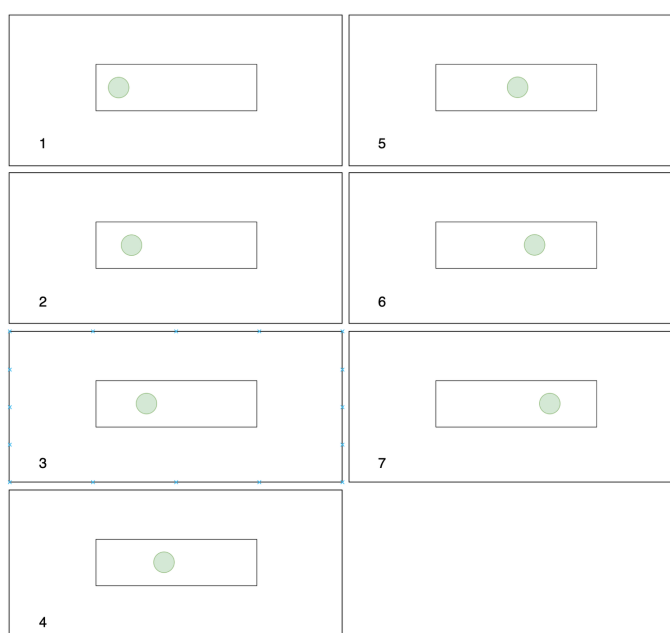


Film dostępny pod adresem </preview/resource/R1aHOEK98DGf1>

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

Film nawiązujący do treści materiału

Składa się ona z siedmiu kolejnych obrazów:



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

Jeżeli obrazków, które składają się finalnie na animację w formacie GIF, byłoby więcej, to zielona kulka poruszałaby się płynniej i bez „skoków” zauważalnych w obecnej postaci.

Kodowanie dźwięku

Wyróżnia się dwie główne metody kodowania dźwięków:

- analogowe,
- cyfrowe.

Za pomocą pierwszej metody, analogowej, możemy zakodować dźwięk na takich nośnikach jak taśmy, kasety czy płyty winylowe. Do odtworzenia dźwięku potrzebny jest wówczas specjalny odtwarzacz, który mechanicznie odczyta treść zapisaną na nośniku.

W drugiej metodzie, cyfrowej, dźwięk zapisywany jest za pomocą sygnałów, których wartości pobierane są co określony odstęp czasu. Proces ten nazywany jest **próbkowaniem**.

Liczba bitów, na której zapiszemy dźwięk zależy będzie od trzech głównych czynników: częstotliwości próbkowania (Hz), rozmiaru próbki (b) oraz czasu próbkowania (s).

$$\text{Liczba bitów } [b] = \text{czas } [s] \cdot \text{częstotliwość próbkowania } [Hz] \cdot \text{rozmiar próbki } [b]$$

Przykład 2

Założmy że kodujemy cyfrowo pewien dźwięk trwający 10 sekund. Częstotliwość próbkowania to 50Hz, natomiast rozmiar próbki to 32 bity. W takiej sytuacji nasz dźwięk zapiszemy na następującej liczbie bitów:

$$10s \cdot 50Hz \cdot 32b = 16000b$$

Kodowanie cyfrowe zapewnia znacznie lepszą jakość dźwięku oraz lepszą możliwość manipulowania dźwiękiem.

Dźwięk zakodowany cyfrowo najczęściej zapisywany jest na płytach CD oraz na dyskach.

Słownik

alfabet łaciński

najbardziej popularny na świecie system znaków; składa się z 26 liter – A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z

bajt

wykorzystywana w dziedzinie informatyki, jednostka pamięci złożona z 8 bitów

Gra edukacyjna

Polecenie 1

Przeanalizuj poniższą prezentację omawiającą sposoby reprezentowania obrazów.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PGxPlpBPk>

1

W tej prezentacji poznasz sposoby kodowania obrazów. Głównymi sposobami są: **grafika rastrowa** oraz **grafika wektorowa**. Oba sposoby kodują informacje za pomocą bitów.

2

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PGxPlpBPk>

Obraz, zakodowany za pomocą grafiki rastrowej, to zbiór wielu punktów o ustalonym konkretnym położeniu (umiejscowionych bardzo blisko siebie), przyjmujących określone kolory. Punkty te nazywa się **pikselami**.

3

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PGxPlpBPk>

Liczba pikseli jest zależna od obszaru zwanego **bitmapą**. Jest to mapa bitów, czyli obszar który określa, ile maksymalnie może się znaleźć pikseli w jego szerokości i wysokości. Bitmapa

o rozmiarze 320x100 będzie polem o szerokości 320 pikseli i długości 100 pikseli. W sumie taki obszar będzie mógł pomieścić maksymalnie 32000 pikseli.

Bitmapa nie dość, że definiuje liczbę i położenie, to także przechowuje informacje o kolorach poszczególnych pikseli.

Im większa jest bitmapa, tym lepsza jest jakość zakodowanej grafiki.

4



Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PGxPlpBPk>

Jeżeli chcielibyśmy utworzyć przedstawiony powyżej kwadrat, zbudowany z 16 pikseli o różnych kolorach, wówczas wykorzystana bitmapa powinna posiadać rozmiar 4x4.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PGxPlpBPk>

5

Kolor każdego z pikseli definiowany jest przez jeden z modeli przestrzeni barw – RGB. Nazwa pochodzi od trzech podstawowych kolorów, na których model ten się opiera – Red (czerwony), Green (zielony) oraz Blue (niebieski).

Kolory inne niż podstawowe uzyskuje się poprzez połączenie, mieszankę kolorów podstawowych z określoną ich intensywnością. W przypadku **24-bitowej** wersji RGB, intensywność określa się wartością od 0 do 255. Istnieją także inne wersje, np. 32-bitowa wersja RGBA, gdzie ostatnie 8 bitów ("A") odpowiada za przezroczystość koloru.

6



Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PGxPlpBPk>

Przykładowo, dzięki mieszance kolorów podstawowych z zadanymi wartościami intensywności: $R = 192$, $G = 128$, $B = 239$ uzyskujemy powyższy kolor.



7

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PGxPlpBPk>

Pojedynczy pixel o kolorze przedstawionym powyżej, zakodowany będzie jako:

W systemie szesnastkowym: 218521

W kodzie binarnym:

001000011000010100100001

8

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PGxPlpBPk>

Innym sposobem kodowania obrazów jest **grafika wektorowa**.

Jest to znacznie lepszy sposób kodowania obrazów, ponieważ, w przeciwieństwie do grafiki rastrowej, zmniejszanie i zwiększanie obrazu nie pogarsza jakości obrazu.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PGxPlpBPk>

9

Każdy obraz zapisany w grafice wektorowej to zbiór podstawowych figur geometrycznych – linii, łuków, wielokątów czy okręgów. Każda z takich figur posiada określone właściwości i parametry jak np. grubość linii, kolor, współrzędne czy promień. To właśnie dzięki zdefiniowaniu tych atrybutów możemy bezproblemowo skalować grafiki wektorowe.

10



Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PGxPlpBPk>

Gdy mamy do czynienia z grafiką rastrową, narysowany na obrazie okrąg będzie tylko zbiorem niezależnych pixeli o określonym położeniu oraz kolorze. W przypadku grafiki wektorowej okrąg to już zdefiniowany obiekt o określonym kształcie, posiadający swój środek i promień. Gdy będziemy chcieli go przeskalować, wówczas nie powiększamy tylko zbioru pixeli, a modyfikujemy parametry konkretnej figury.

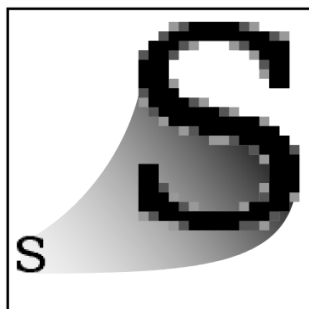
Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PGxPlpBPk>

11

Grafika wektorowa jest zdecydowanie częściej wykorzystywana w programowaniu, ponieważ, z poziomu programu, algorytmu, pozwala na dostęp do konkretnych elementów obrazka.

12



Raster

.jpeg .gif .png



Vector

.svg

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PGxPlpBPk>

Porównanie skalowania grafiki rastrowej i wektorowej zostało przedstawione na obrazku.

13

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PGxPlpBPk>

Obrazy zapisane zarówno w grafice rastrowej i grafice wektorowej mają określone formaty plików. Typowe formaty dla omówionych rodzajów grafik:

Grafika rastrowa:

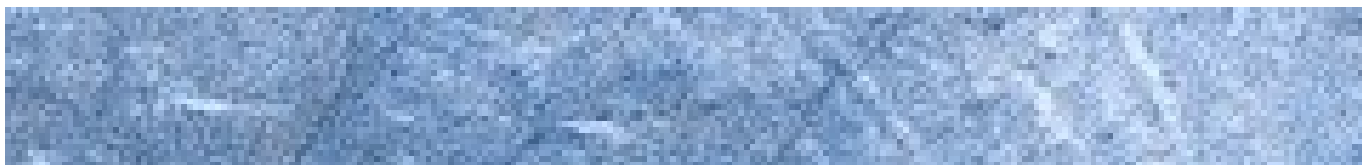
- TIFF (*Tagged Image File Format*)
- BMP (*Windows Bitmap*)
- JPEG (*Joint Photographic Experts Group*)

Grafika wektorowa:

- CDR (*Corel Draw*)
- SVG (*Scalable Vector Graphics*)
- SWF (*Adobe Flash Format*)

Polecenie 2

Weź udział w grze sprawdzającej twoją wiedzę.



Test

Reprezentacja tekstu w komputerze

Poziom trudności:

łatwy

Limit czasu:

5 min

Twój ostatni wynik:

—

Uruchom

Polecenie 3

Zapisz notatkę, w której podsumujesz najważniejsze informacje przedstawione w prezentacji multimedialnej.

Polecenie 4

Zaproponuj trzy pytania, które mogłyby pojawić się w grze edukacyjnej.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Zaznacz wszystkie poprawne odpowiedzi. Wskaż typowe formaty dla obrazów zapisanych za pomocą grafiki wektorowej.

☐ SWF

☐ JPG

☐ CDR

☐ SVG

Ćwiczenie 2



Wskaż podstawowy kolor modelu barw RGB.

☐ pomarańczowy

☐ niebieski

☐ żółty

☐ różowy

Ćwiczenie 3



Połącz w pary system kodowania znaków z odpowiednim opisem.

Unicode

najbardziej obszerny system kodowania, obejmuje około 140 tysięcy różnych znaków

ASCII

umożliwia zakodowanie 128 różnych znaków

ISO 8859-2

umożliwia zakodowanie 256 różnych znaków, w tym polskie znaki narodowe

Ćwiczenie 4



Przypisz znaki odpowiednim zakresom liczbowym, zgodnie z tabelą ASCII.

liczby od 48 do 57

!	x	Z	q	5	9	&
#	t	s	2	\$	3	P
B	K	A				

liczby od 65 do 90

liczby od 97 do 122

liczby od 33 do 47

Ćwiczenie 5



Wskaż, jakie znaki (zgodnie z ASCII) przypisane są liczbom z zakresu od 0 do 32.

☐ cyfry

☐ małe litery

☐ polskie znaki

☐ znaki sterujące

Ćwiczenie 6



Wskaż, który ze standardów ISO obejmuje polskie znaki narodowe.

☐ ISO-8859-7

☐ ISO-8859-5

☐ ISO-8859-2

☐ ISO-8859-1

Ćwiczenie 7



Wskaż poprawną odpowiedź.

Czy jeżeli piszemy dokument tekstowy w standardzie ISO-8859-2, możemy w nim użyć zarówno polskich znaków oraz znaków rosyjskiego alfabetu – cyrylicy?

☐ nie

☐ tak

Ćwiczenie 8



Wstaw w tekst odpowiednie cyfry.

ASCII jest -bitowym systemem kodowania znaków, natomiast standard ISO-8859-1 wykorzystuje bitów.

Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Reprezentacja tekstu w komputerze

Grupa docelowa:

Liceum ogólnokształcące i technikum, 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.

Treści nauczania – wymagania szczegółowe

I. Rozumienie, analizowanie i rozwiązywanie problemów.

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

2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych;

5) przedstawia sposoby reprezentowania w komputerze znaków, liczb, wartości logicznych, obrazów, dźwięków, animacji;

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):

- Wyjaśnisz, w jaki sposób kodowany jest tekst w komputerze.
- Przeanalizujesz metody kodowania obrazów i animacji.

- Rozwiążesz ćwiczenia dotyczące sposobów reprezentowania informacji w komputerach.

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: „Reprezentacja tekstu w komputerze”. Nauczyciel prosi uczniów o zapoznanie 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.** Uczniowie przystępują do cichego czytania tekstu e-materiału. Indywidualnie zapoznają się z treścią w sekcji „Przeczytaj”.

2. **Praca z multimediami.** Nauczyciel wyświetla zawartość sekcji „Prezentacja multimedialna”. Uczniowie wspólnie zapoznają się z prezentacją omawiającą sposoby reprezentowania obrazów.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują indywidualnie ćwiczenia nr 1-8. Po wykonaniu każdego z nich następuje omówienie rozwiązania przez nauczyciela.

Faza podsumowująca:

1. Na koniec zajęć nauczyciel raz jeszcze wyświetla na tablicy temat lekcji i cele zawarte w sekcji „Wprowadzenie”. W odniesieniu do ich realizacji dokonuje szczegółowej oceny rozwiązania zastosowanego przez wybranego ucznia.
2. Na koniec zajęć nauczyciel prosi uczniów o rozwinięcie zdania: „Na dzisiejszych zajęciach nauczyłam/łem się jak...”.

Praca domowa:

1. Uczniowie kodują za pomocą tabeli ASCII liczby odpowiadające dacie urodzenia, np. 12.11.06 (12 listopada 2006 roku).

Wskazówki metodyczne:

- Treści w sekcji „Przeczytaj” można wykorzystać jako podsumowanie i utrwalenie wiedzy uczniów.
- W ramach utrwalenia wiadomości uczniowie mogą rozwiązać test (Gra edukacyjna).