



Szyfr Cezara w języku Java

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Prezentacja multimedialna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Poznaliśmy już [szyfr Cezara](#), czyli klasyczny szyfr przesuwający (podstawieniowy). Jego działanie polega na zastąpieniu każdej z liter tekstu jawnego odpowiadającą jej literą, oddaloną o określoną liczbę miejsc w alfabecie.

W tym e-materiale dowiesz się, w jaki sposób można zapisać algorytm szyfru Cezara w języku Java.

Ciekawi cię, jak wyglądają implementacje w innych językach programowania? Możesz się z nimi zapoznać w dwóch pozostałych e-materiałach z tej serii:

- [Implementacja szyfru Cezara w języku C++](#),
- [Szyfr Cezara w języku Python](#).

Więcej zadań? Sięgnij do [Szyfr Cezara – zadania maturalne](#).

Twoje cele

- Wykorzystasz w praktyce wiedzę na temat szyfru Cezara.
- Przeanalizujesz implementacje algorytmu szyfrującego i deszyfrującego wiadomości za pomocą szyfru Cezara.
- Rozwiążesz kilka zadań związanych z szyfrem Cezara.

Przeczytaj

Komputerowa realizacja w języku Java

Zanim spróbujemy napisać własną [implementację algorytmu](#) szyfru Cezara, przypomnijmy kilka ważnych kwestii związanych z językiem Java.

- Aby odwołać się od i -tego znaku w łańcuchu znaków (`String`), należy użyć notacji `tekst.charAt(i)`, gdzie `tekst` oznacza nazwę łańcucha znaków.
- Do sprawdzenia długości łańcucha znaków służy funkcja `length()`. Jest ona wywoływana w następujący sposób: `tekst.length()`, gdzie `tekst` to nazwa łańcucha znaków.
- W języku Java wykorzystywany jest standard kodowania Unicode. W przypadku dokonywania operacji arytmetycznej na znaku wchodzącym w skład łańcucha należy zwrócić uwagę na otrzymaną wartość. Znak zostanie zastąpiony odpowiadającym mu kodem. Przykładowo, wykonując działanie `'A' * 1`, otrzymamy wartość 65, oznaczającą, że kod litery A wynosi właśnie 65.
- Aby zamienić cały łańcuch znaków na taki, który składa się wyłącznie z wielkich liter, użyjemy funkcji `tekst.toUpperCase()`. Zamianę w przeciwnym kierunku, czyli na małe litery, możemy wykonać funkcją `tekst.toLowerCase()`, gdzie `tekst` to nazwa łańcucha znaków.

Polecenie 1

Napisz w języku Java program szyfrujący przy użyciu szyfru Cezara łańcuch znaków wyraz składający się z wielkich liter alfabetu łacińskiego. Przetestuj działanie programu dla wiadomości ENIGMA dla klucza równego 3. Pamiętaj, że w szyfrze Cezara nie rozróżnia się małych i wielkich liter. Spacje zapisujemy bez zmian.

Specyfikacja problemu:

Dane:

- wyraz – łańcuch znaków; tekst jawny
- klucz – liczba naturalna; klucz szyfrowania

Wynik:

szyfrogram – łańcuch znaków; zaszyfrowany tekst jawny

Polecenie 2

Porównaj swoje rozwiązanie z zaproponowanym w prezentacji.

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

Słownik

implementacja algorytmu

komputerowa realizacja algorytmu przy użyciu języka programowania; często mianem implementowania określa się sam proces tworzenia programu

klucz szyfrujący

wartość używana do zamiany tekstu jawnego na zaszyfrowany; w przypadku szyfru Cezara jest to liczba określająca, o ile miejsc w alfabecie oddalone są litery tekstu jawnego od odpowiadających im znaków tekstu zaszyfrowanego

szyfr Cezara

szyfr podstawieniowy, w którym każdy znak tekstu jawnego jest zastępowany przez inny znak, oddalony od niego o stałą liczbę pozycji (tzw. klucz)

Prezentacja multimedialna

Polecenie 1

Napisz w języku Java program deszyfrujący łańcuch znaków wyraz składający się z liter alfabetu łacińskiego (małych i wielkich) przy użyciu szyfru Cezara. Przetestuj jego działanie dla wiadomości aBcDeFgHiJkLmNoPqRsTuVwXyZ oraz klucza równego -5.

Twój program powinien poprawnie zinterpretować znaki spacji (wstawić ją w tekście jawnym tam, gdzie pojawiła się w tekście tajnym). Program powinien szyfrować poprawnie łańcuchy znaków składające się z liter alfabetu i ew. spacji.

Pamiętaj, że klucze są liczbami całkowitymi, mogą zatem być ujemne. W sytuacji, kiedy klucz jest liczbą ujemną, program powinien dodawać do niej liczbę 26 (tyle jest liter w alfabecie łacińskim) tak długo, dopóki wartość klucza nie będzie liczbą dodatnią.

Specyfikacja problemu:

Dane:

- wyraz – łańcuch znaków; zaszyfrowany tekst; tekst składający się z liter i spacji
- klucz – liczba całkowita; klucz szyfrowania




Wynik:

- jawny – odszyfrowany tekst

Polecenie 2

Porównaj swoje rozwiązanie z zaproponowanym w prezentacji.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Napisz program, który poda kod zmiennej `litera` w systemie Unicode. Przetestuj działanie programu dla zmiennej `litera` o wartości `G`.

Specyfikacja problemu:

Dane:

- `litera` – znak, którego kod w systemie Unicode chcemy uzyskać

Wynik:

- `szyfrogram` – liczba całkowita, kod znaku `litera` w systemie Unicode

Ćwiczenie 2



Napisz program, który poda, jakim kluczem zostało zaszyfrowane słowo za pomocą szyfru Cezara.

Działanie programu przetestuj dla odszyfrowanego słowa INforMATYkA oraz zaszyfrowanego QVnwzUIBGsI.

Specyfikacja problemu:

Dane:

- `zaszyfrowanyWyraz` – ciąg znaków; słowo zaszyfrowane szyfrem Cezara
- `odszyfrowanyWyraz` – ciąg znaków; słowo odszyfrowane szyfrem Cezara

Wynik:

- `klucz` – liczba naturalna; wartość klucza użyta do zaszyfrowania
`słowa odszyfrowanyWyraz`

Ćwiczenie 3



Napisz program, który zaszyfruje ciąg zdanieDoZaszyfrowania, używając szyfru, który przesuwa każdą literę o kwadrat jej pozycji w alfabecie łacińskim. Zachowaj wielkość liter. Ciąg składa się jedynie ze znaków alfabetu łacińskiego. Spacje nie są przesuwane.

Działanie programu przetestuj dla ciągu znaków IgnORANtIA IURiS noCET.

Przykład 1

Jeżeli litera jest na k -miejscu w alfabecie, to zaszyfrowana litera będzie stała na miejscu w alfabecie przesuniętym o $(k \cdot k) \bmod 26$ (ponieważ właśnie tyle liter zawiera alfabet łaciński). Np. dla litery A (małej i wielkiej) przesunięcie wyniesie 1, zatem uzyskamy B, dla K przesunięcie wyniesie 17 – uzyskamy również literę B!

Uwaga: jeżeli przesunięcie wyniesie 0, kodowana litera się nie zmienia.

Specyfikacja problemu:

Dane:

- zdanieDoZaszyfrowania – ciąg znaków; tekst jawny

Wynik:

- zaszyfrowaneZdanie – ciąg znaków zaszyfrowany szyfrem, który każdą literę w ciągu znaków zdanieDoZaszyfrowania przesuwa o kwadrat jej pozycji w alfabecie łacińskim

Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Szyfr Cezara w języku Java

Grupa docelowa:

Szkoła ponadpodstawowa, liceum ogólnokształcące, technikum, zakres podstawowy i 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 podstawowy. Uczeń:

2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy:

b) na tekstach: porównywania tekstów, wyszukiwania wzorca w tekście metodą naiwną, szyfrowania tekstu metodą Cezara i przestawieniową,

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):

- Wykorzystasz w praktyce wiedzę na temat szyfru Cezara.
- Przeanalizujesz implementacje algorytmu szyfrującego i deszyfrującego wiadomości za pomocą szyfru Cezara.
- Rozwiążesz kilka zadań związanych z szyfrem Cezara.

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-materiałach;
- tablica interaktywna/tablica, pisak/kreda;
- oprogramowanie dla języka Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji).

Przebieg lekcji

Przed lekcją:

1. Uczniowie powtarzają najważniejsze informacje dotyczące szyfru Cezara.
2. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Szyfr Cezara w języku Java”. Uczniowie zapoznają się z treściami w sekcji „Przeczytaj” w kontekście programowania.

Faza wstępna:

1. Wyświetlenie przez nauczyciela tematu i celów zajęć, przejście do wspólnego ustalenia kryteriów sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Uczniowie tworzą pytania dotyczące tematu zajęć, na które odpowiedzą w trakcie lekcji.

Faza realizacyjna:

1. Chętna lub wybrana osoba referuje najważniejsze informacje dotyczące szyfru Cezara. W razie potrzeby nauczyciel uzupełnia jej wypowiedź.

2. **Praca z multimediami.** Nauczyciel wyświetla zawartość sekcji „Prezentacja multimedialna”. Uczniowie wspólnie zapoznają się z jego treścią.
3. Nauczyciel dzieli klasę na czteroosobowe grupy. Każda z grup dzieli się na dwie dwuosobowe podgrupy. Pierwsza z podgrup przygotowuje wiadomość do zaszyfrowania i program, który zaszyfruje ją szyfrem Cezara. Szyfruje wiadomość i wraz z kluczem przekazuje drugiej podgrupie. Pierwszym zadaniem drugiej podgrupy jest przygotowanie programu, który odszyfruje wiadomość. Następnie do odszyfrowanej wiadomości należy dodać nowy fragment, a całość zaszyfrować szyfrem płótkowym. Druga podgrupa przekazuje pierwszej zaszyfrowaną wiadomość oraz klucz. Zadaniem pierwszej podgrupy jest odszyfrowanie ostatecznej wiadomości. Podgrupy nie mogą ze sobą rozmawiać. By utrudnić zadanie, klucz może być ukryty pod postacią największego wspólnego dzielnika dwóch liczb etc.
4. Drużyna, która pierwsza wykona zadanie, wygrywa.
5. **Ćwiczenie umiejętności.** Uczniowie wykonują ćwiczenie 1–2 z sekcji „Sprawdź się”.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat lekcji zawarty w sekcji „Wprowadzenie” i inicjuje krótką rozmowę na temat zrealizowanych celów (czego uczniowie się nauczyli).
2. Na koniec zajęć z programowania w Javie nauczyciel prosi uczniów o rozwinięcie zdania: „Na dzisiejszych zajęciach nauczyłam/łem się jak...”.

Praca domowa:

1. Uczniowie wykonują ćwiczenie 3 z sekcji „Sprawdź się”.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka Java SE 8 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Eclipse 4.4 (lub nowszej wersji).

Wskazówki metodyczne:

- Treści w sekcji „Przeczytaj” można wykorzystać jako podsumowanie i utrwalenie wiedzy uczniów.