



Proste projekty i funkcje w języku Java

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



Z pojęciem funkcji spotkaliśmy się już na lekcjach matematyki. Wiemy również, jaka jest rola funkcji w informatyce. W programach opisują one, jak na podstawie danych wejściowych obliczyć pewną wartość.

Więcej na ten temat znajdziesz w e-materiale [Proste projekty i funkcje](#). W tym e-materiale zapoznamy się z wykorzystaniem funkcji w projekcie w języku Java.

Implementację projektów w pozostałych językach programowania znajdziesz w e-materiałach:

- [Proste projekty i funkcje w języku C++](#),
- [Proste projekty i funkcje w języku Python](#).

Więcej zadań? Przejdź do e-materiału [Proste projekty i funkcje – ćwiczenia](#).

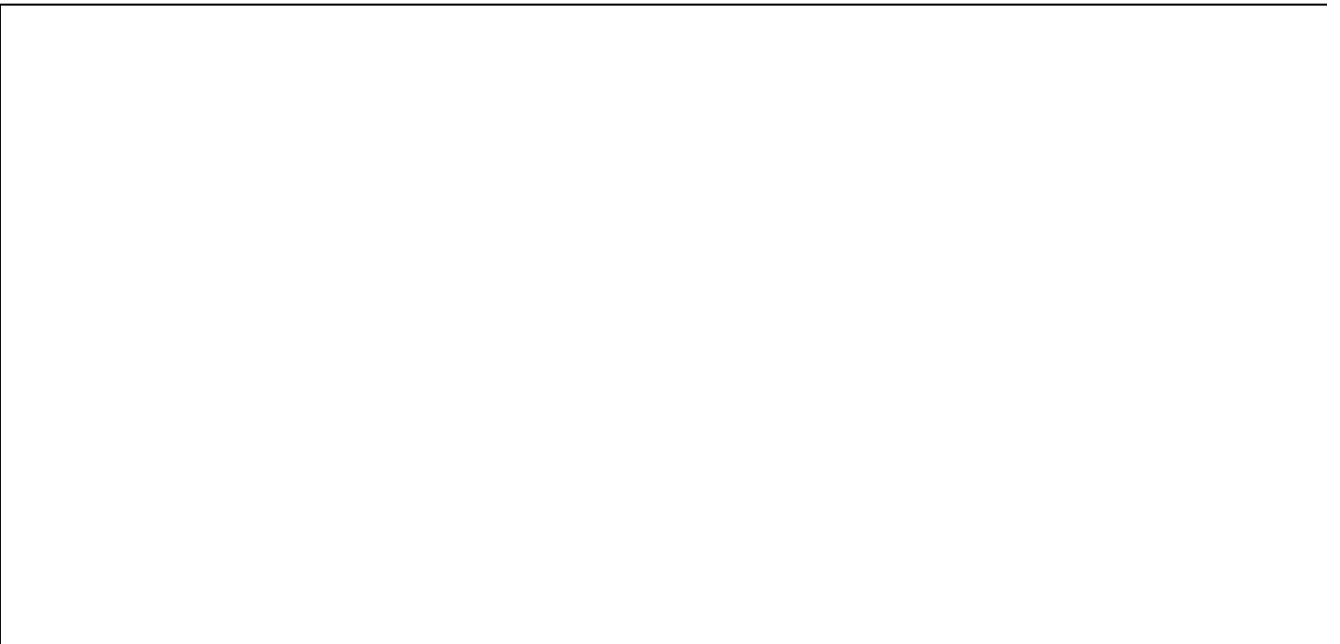
Twoje cele

- Przeanalizujesz informacje dotyczące tworzenia i wywoływania funkcji w języku Java.
- Zapiszesz algorytm rozwiązywania równania kwadratowego, wykorzystując język programowania Java.
- Wykonasz kilka ćwiczeń sprawdzających twoją wiedzę na temat funkcji.

Animacja

Polecenie 1

Zapoznaj się z animacją omawiającą sposób sprawdzania poprawności numeru PESEL. Zapisz specyfikację i pseudokod zaprezentowanego programu.





Film dostępny pod adresem </preview/resource/RhGA8CfWR9MGS>

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY 3.0.

Film nawiązujący do treści proste projekty i funkcje w języku java.

Polecenie 2

Zapoznaj się z animacją omawiającą sposób sprawdzania poprawności numeru PESEL. Zapisz w języku Java zaprezentowany program.



Przeczytaj

Rozwiązywanie równania kwadratowego – implementacja w języku Java

Specyfikacja problemu:

Dane:

- a – liczba rzeczywista; różna od zera; współczynnik przy x^2
- b – liczba rzeczywista; współczynnik przy x
- c – liczba rzeczywista; wyraz wolny

Wynik:

Program, na standardowym wyjściu, wypisuje odpowiedni komunikat:

- Równanie nie ma rozwiązań,
- Równanie ma jedno rozwiązanie $x_1 = X_1$, gdzie w miejscu X_1 powinna znajdować się wyznaczona wartość miejsca zerowego,
- Równanie ma dwa rozwiązania: $x_1 = X_1$ $x_2 = X_2$, gdzie w miejscu X_1 , X_2 powinny znajdować się wyznaczone wartości miejsc zerowych.

Zaimplementujmy stabilny algorytm rozwiązywania równania kwadratowego. Pierwszym krokiem jest zadeklarowanie [funkcji statycznej](#) `obliczRownanie()`, która zwróci łańcuch znaków wynik. Będzie on wyświetlał obliczone wartości pierwiastków równania kwadratowego lub informację o braku rozwiązań.

```
1 static String obliczRownanie(double a, double b, double c) {  
2     String wynik = "";  
3  
4     return wynik;  
5 }
```

Tworzymy blok instrukcji warunkowej, która obsłuży przypadki: $\Delta > 0$, $\Delta = 0$, oraz $\Delta < 0$.

Pierwszym przypadkiem, którym się zajmiemy, jest $\Delta < 0$. W takiej sytuacji zwracamy komunikat, że równanie kwadratowe nie ma rozwiązań.

```
1 static String obliczRownanie(double a, double b, double c) {  
2     String wynik = "";
```

```

3      double delta = b * b - 4 * a * c;
4
5      if (delta < 0) {
6          wynik = "Rownanie nie ma rozwiazan";
7      } else if (delta == 0) {
8
9      }
10
11     return wynik;
12 }

```

Kolejny rozpatrywany przypadek, to sytuacja, kiedy $\Delta = 0$. W takiej sytuacji równanie ma jedno rozwiązanie.

```

1 static String obliczRownanie(double a, double b, double c) {
2     String wynik = "";
3
4     double delta = b * b - 4 * a * c;
5
6     if (delta < 0) {
7         wynik = "Rownanie nie ma rozwiazan";
8     } else if (delta == 0) {
9         double x1 = 0;
10        x1 = -b / (2 * a);
11        System.out.println("Rownanie ma jedno rozwiazanie x1 = "
12    }
13    return wynik;
14 }

```

Następnym przypadkiem jest sytuacja, gdy wyróżnik trójkianu kwadratowego jest większy niż 0 (blok else). Deklarujemy zmienne, które będą przechowywały oba pierwiastki: double x1 oraz double x2.

```

1 static String obliczRownanie(double a, double b, double c) {
2     String wynik = "";
3
4     double delta = b * b - 4 * a * c;
5
6     if (delta < 0) {
7         wynik = "Rownanie nie ma rozwiazan";

```

```

8      } else if (delta == 0) {
9          double x1 = 0;
10         x1 = -b / (2 * a);
11         System.out.println("Rownanie ma jedno rozwiazanie x1 = "
12
13     } else {
14         double x1 = 0;
15         double x2 = 0;
16
17     }
18
19     return wynik;
20 }

```

Rozpoczynamy implementację kodu wyliczającego wartości pierwiastków na podstawie wzorów Viète'a. Tworzymy instrukcje warunkową. Sprawdzamy w niej wartość zmiennej b. Jeżeli jest ona mniejsza od zera, wówczas pod zmienną x2 zapisujemy wartość obliczoną ze wzoru: $\frac{-b + \sqrt{\Delta}}{2a}$. Z kolei drugi pierwiastek, x1, z wyrażenia: $\frac{c}{ax_2}$.

```

1  static String obliczRownanie(double a, double b, double c) {
2      String wynik = "";
3
4      double delta = b * b - 4 * a * c;
5
6      if (delta < 0) {
7          wynik = "Rownanie nie ma rozwiazan";
8      } else if (delta == 0) {
9          double x1 = 0;
10         x1 = -b / (2 * a);
11         System.out.println("Rownanie ma jedno rozwiazanie x1 = "
12
13     } else {
14         double x1 = 0;
15         double x2 = 0;
16
17         if(b < 0){
18             x2 = (-b + Math.sqrt(delta)) / (2 * a);
19             x1 = c / (a * x2);
20         }
21

```

```

22     }
23
24     return wynik;
25 }

```

Jeżeli zmienna b przechowuje wartość większą bądź równą 0, wówczas do zmiennej x1 zapisujemy wartość wyliczoną ze wzoru: $\frac{-b - \sqrt{\Delta}}{2a}$. Wartość zmiennej x2 liczymy za pomocą wyrażenia: $\frac{c}{ax_1}$. Do zmiennej wynik zapisujemy informacje o rozwiązaniu.

```

1 static String obliczRownanie(double a, double b, double c) {
2     String wynik = "";
3
4     double delta = b * b - 4 * a * c;
5
6     if (delta < 0) {
7         wynik = "Rownanie nie ma rozwiazan";
8     } else if (delta == 0) {
9         double x1 = 0;
10        x1 = -b / (2 * a);
11        System.out.println("Rownanie ma jedno rozwiazanie x1 = "
12
13    } else {
14        double x1 = 0;
15        double x2 = 0;
16
17        if(b < 0){
18            x2 = (-b + Math.sqrt(delta)) / (2 * a);
19            x1 = c / (a * x2);
20
21        } else {
22            x1 = (-b - Math.sqrt(delta)) / (2 * a);
23            x2 = c / (a * x1);
24        }
25
26        wynik = "Rownanie ma dwa rozwiazania: \nx1 = " + x1 + " \
27    }
28
29    return wynik;
30 }

```

Należy jeszcze pobrać od użytkownika wartości współczynników funkcji kwadratowej (a, b oraz c). Zrobimy to, korzystając z klasy `Scanner` i funkcji `nextDouble()`, ponieważ współczynniki a, b i c są właśnie typu `double`.

Ważne!

Ponieważ w programie użyta została zewnętrzna klasa `Scanner`, należy pamiętać o dodaniu instrukcji:

```
1 import java.util.Scanner;
```

```
1 public static void main(String[] args) {  
2  
3     double a, b, c;  
4     Scanner scanner = new Scanner(System.in);  
5  
6     System.out.println("Podaj wartosc wspolczynnika a:");  
7     a = scanner.nextDouble();  
8     System.out.println("Podaj wartosc wspolczynnika b:");  
9     b = scanner.nextDouble();  
10    System.out.println("Podaj wartosc wspolczynnika c:");  
11    c = scanner.nextDouble();  
12  
13    System.out.println(obliczRownanie(a, b, c));  
14  
15 }
```

Program jest gotowy. Oto cały kod:

```
1 import java.util.Scanner;  
2  
3 public class RownanieKwadratowe {  
4  
5     static String obliczRownanie(double a, double b, double c) {  
6         String wynik = "";  
7  
8         double delta = b * b - 4 * a * c;  
9  
10        if (delta < 0) {  
11            wynik = "Rownanie nie ma rozwiazan";  
12        } else if (delta == 0) {
```

```

13         double x1 = 0;
14         x1 = -b / (2 * a);
15         System.out.println("Rownanie ma jedno rozwiazanie x1
16
17     } else {
18         double x1 = 0;
19         double x2 = 0;
20
21         if(b < 0){
22             x2 = (-b + Math.sqrt(delta)) / (2 * a);
23             x1 = c / (a * x2);
24
25         } else {
26             x1 = (-b - Math.sqrt(delta)) / (2 * a);
27             x2 = c / (a * x1);
28         }
29
30         wynik = "Rownanie ma dwa rozwiazania: \nx1 = " + x1 +
31     }
32
33     return wynik;
34 }
35
36 public static void main(String[] args) {
37
38     double a, b, c;
39     Scanner scanner = new Scanner(System.in);
40
41     System.out.println("Podaj wartosc wspolczynnika a:");
42     a = scanner.nextDouble();
43     System.out.println("Podaj wartosc wspolczynnika b:");
44     b = scanner.nextDouble();
45     System.out.println("Podaj wartosc wspolczynnika c:");
46     c = scanner.nextDouble();
47
48     System.out.println(obliczRownanie(a, b, c));
49
50 }
51 }

```

Słownik

algorytm numerycznie stabilny

algorytm, który gwarantuje rozwiązania (dla dowolnego zadania rozważnej klasy)
z błędem tego samego rzędu, co optymalny poziom błędu rozwiązania danego zadania

funkcja statyczna

funkcja, do użycia której nie jest wymagane tworzenie obiektu

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Uzupełnij kod programu tak, aby obliczał rozwiązanie równania kwadratowego w postaci $ax^2 + bx + c = 0$. Przetestuj jego działanie dla następujących danych:

- $a = -20$
- $b = 7$
- $c = 8$

Specyfikacja problemu:

Dane:

- a – liczba rzeczywista; współczynnik przy x^2
- b – liczba rzeczywista; współczynnik przy x
- c – liczba rzeczywista; wyraz wolny

Wynik:

Program, na standardowym wyjściu, wypisuje odpowiedni komunikat:

- Nie jest to równanie kwadratowe,
- Równanie nie ma rozwiązania,
- Równanie ma jedno rozwiązanie $x_1 = X_1$, gdzie w miejscu X_1 powinna znajdować się wyznaczona wartość miejsca zerowego,
- Równanie ma dwa rozwiązania: $x_1 = X_1$ $x_2 = X_2$, gdzie w miejscu X_1 , X_2 powinny znajdować się wyznaczone wartości miejsc zerowych.

Przykładowe wyjście:

```
1 Rownanie ma dwa rozwiazania:  
2 x1 = 0.8312202374203345  
3 x2 = -0.4812202374203344
```

Twoje zadania

1. Program rozwiązuje równanie kwadratowe o zadanych współczynnikach a, b, c oraz wypisuje odpowiedni komunikat.





Pan Krzysztof jest numizmatykiem. Niedawno na lokalnej giełdzie staroci udało mu się zakupić paczkę starych banknotów. Pana Krzysztofa interesują banknoty, których siedmiocyfrowy numer seryjny jest palindromem, to znaczy czytany od lewej strony jest identyczny jak czytany od strony prawej, np. 3245423. Napisz funkcję wskazującą, które z banknotów mają numer seryjny będący palindromem. Numery seryjne banknotów przechowywane są w n -elementowej tablicy `numery`, a każdy numer seryjny ma siedem znaków. Pan Krzysztof ponumerował banknoty od 0. Działanie programu przetestuj dla następujących danych:

- $n = 19$
- `numery = ["1233332", "1112111", "3132204",
"3412589", "9014521", "5213645", "1234567",
"9234821", "1423405", "1234961", "5429245",
"8730378", "1234543", "6345234", "9910199",
"6655321", "1122334", "9948328", "7786877"]`

Specyfikacja problemu:

Dane:

- n – liczba naturalna dodatnia; rozmiar tablicy `numery`
- `numery` – tablica zawierająca n napisów, każdy rozmiaru 7

Wynik:

Program powinien wyświetlać indeksy z tablicy `numery`, na których znajdują się numery seryjne będące palindromami. Indeksy powinny być oddzielone spacją.

Przykładowe wyjście:

```
1 1 10 11 14 18
```

1. Program powinien wypisać numery indeksów numerów seryjnych z n-elementowej tablicy numery, które są palindromami.

Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Proste projekty i funkcje w języku Java

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres podstawowy

Podstawa programowa:

Cele kształcenia – wymagania ogólne

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.

Treści nauczania – wymagania szczegółowe

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres podstawowy. Uczeń:

- 4) porównuje działanie różnych algorytmów dla wybranego problemu, analizuje algorytmy na podstawie ich gotowych implementacji;
- 5) sprawdza poprawność działania algorytmów dla przykładowych danych.

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

Zakres podstawowy. Uczeń:

- 1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2);

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

1) zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy poznane na wcześniejszych etapach oraz algorytmy:

g) obliczania przybliżonej wartości pierwiastka kwadratowego,

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 informacje dotyczące tworzenia i wywoływania funkcji w języku Java.
- Zapiszesz algorytm rozwiązywania równania kwadratowego, wykorzystując język programowania Java.
- Wykonasz kilka ćwiczeń sprawdzających twoją wiedzę na temat funkcji.

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: „Proste projekty i funkcje w języku Java”. Uczniowie zapoznają się z treściami w sekcji „Animacja”.

Faza wstępna:

1. Nauczyciel wprowadza uczniów szczegółowo w temat lekcji i jej cele. Może posłużyć się wyświetloną na tablicy zawartością sekcji „Wprowadzenie”.

Faza realizacyjna:

1. **Praca z multimediami.** Uczniowie rozwiązują polecenie 1 w parach. Na forum klasy dyskutują na temat ewentualnych trudności.
2. **Praca z tekstem.** Uczniowie zapoznają się z sekcją „Przeczytaj”. Nauczyciel odpowiada na ewentualne pytania, objaśnia niezrozumiałe zagadnienia.
3. **Ćwiczenie umiejętności.** Prowadzący zapowiada uczniom, że będą rozwiązywać ćwiczenie nr 1 z sekcji „Sprawdź się”. Uczniowie wykonują je w parach. Po ustalonym czasie następuje porównanie napisanych kodów podczas wspólnego omówienia rozwiązań.

Faza podsumowująca:

1. Nauczyciel wyświetla na tablicy temat lekcji i cele zawarte w sekcji „Wprowadzenie”. W kontekście ich realizacji podsumowuje przebieg zajęć, a także wskazuje mocne i słabe strony pracy uczniów.
2. Na koniec zajęć nauczyciel prosi uczniów o rozwinięcie zdania: „Na dzisiejszych zajęciach nauczyłam/łem się jak...”.

Praca domowa:

1. Uczniowie rozwiązują problem 2 z sekcji „Animacja”.
2. Uczniowie wykonują ćwiczenie 2 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 na lekcji jako podsumowanie i utrwalenie wiedzy uczniów.