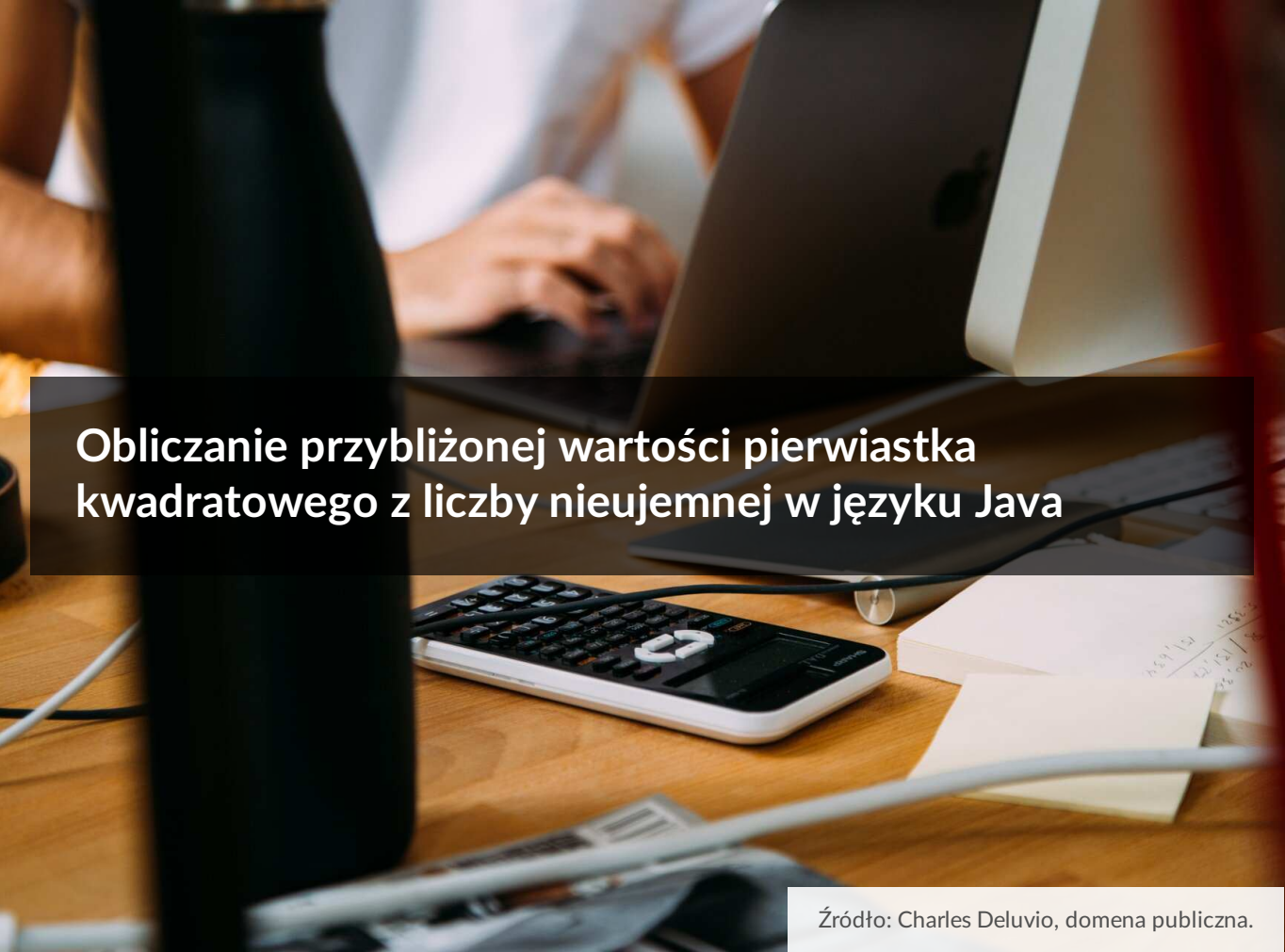




Obliczanie przybliżonej wartości pierwiastka kwadratowego z liczby nieujemnej w języku Java

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



Obliczanie przybliżonej wartości pierwiastka kwadratowego z liczby nieujemnej w języku Java

Źródło: Charles Deluvio, domena publiczna.

Z e-materiału [Obliczanie przybliżonej wartości pierwiastka kwadratowego z liczby nieujemnej](#) znasz algorytm obliczania pierwiastka kwadratowego. W tym e-materiale zaimplementujemy go w języku Java.

Implementacje tego algorytmu w innych językach programowania zostały omówione w e-materiałach:

- [Obliczanie przybliżonej wartości pierwiastka kwadratowego z liczby nieujemnej w języku C++](#),
- [Obliczanie przybliżonej wartości pierwiastka kwadratowego z liczby nieujemnej w języku Python](#).

Więcej zadań? Przejdź do: [Obliczanie przybliżonej wartości pierwiastka kwadratowego z liczby nieujemnej – zadania maturalne](#).

Twoje cele

- Prześledzisz założenia algorytmu Newtona-Raphsona.
- Przeanalizujesz implementację algorytmu Newtona-Raphsona.
- Zapiszesz algorytm w języku Java obliczania przybliżonej wartości pierwiastka kwadratowego.

Implementacja metody Newtona-Raphsona w języku Java

Metoda Newtona-Raphsona służy do wyznaczania przybliżonej wartości pierwiastka kwadratowego. Geometryczna interpretacja algorytmu jest następująca: tworzymy kolejne prostokąty, coraz bardziej zbliżone do poszukiwanej figury. Załóżmy, że pierwiastkujemy pewną liczbę c , a pierwszy wielokąt wyglądać będzie następująco:

$$b = \frac{c}{a}$$
$$P = c$$
$$a = \frac{c}{2}$$

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

Zapiszmy funkcję odpowiedzialną za proces wyszukiwania docelowej długości boku a kwadratu. Pamiętajmy, że jedną z niezbędnych zmiennych będzie również precyzja, której rolę zajmiemy się w następnym kroku. Na razie w funkcji znajdą się jedynie założenia początkowe, dotyczące długości boków pierwszego prostokąta.

```
1 public class Pierwiastkowanie {  
2     public static double pierwiastkowanie(double pierwiastkowany,  
3         double a = pierwiastkowany / 2;  
4         double b = pierwiastkowany / a;  
5  
6     }  
7 }
```

W kolejnym kroku musimy stworzyć pętlę odpowiedzialną za wyliczanie kolejnych długości boków prostokąta. Operację tę przeprowadzamy do momentu, gdy wartość bezwzględna różnicy pomiędzy długościami obu boków będzie mniejsza niż zadana precyzja. Im mniejszą wartość zmiennej precyzja ustalimy, tym wynik będzie dokładniejszy. Długości boków prostokątów wyliczamy w następujący sposób:

$$a = \frac{(a + b)}{2}$$

$$b = \frac{c}{a}$$

gdzie a i b oznaczają długości boków prostokąta, a c jego pole powierzchni.

Mając już niezbędne informacje, możemy przejść do implementacji funkcji. Powtarzanie przedstawionych operacji, dopóki nie spełnimy wspomnianego warunku odnośnie do precyzji i różnicy boków, oznacza konieczność zastosowania pętli. Dodatkowo w warunku zastosujemy wartość bezwzględną. Użyjemy funkcji `abs` z biblioteki `Math`.

```

1 import java.util.Scanner;
2
3 public class Pierwiastkowanie {
4     public static double pierwiastkowanie(double pierwiastkowany,
5         double a = pierwiastkowany / 2;
6         double b = pierwiastkowany / a;
7
8         while (Math.abs(a - b) > precyzja) {
9             a = (a + b) / 2;
10            b = pierwiastkowany / a;
11        }
12        return a;
13    }

```

Teraz wystarczy już tylko pobrać od użytkownika pierwiastkowaną liczbę oraz żądaną precyzję. Po wywołaniu funkcji otrzymamy wypisany przybliżony wynik pierwiastkowania kwadratowego.

```

1 public static void main(String[] args) {
2     Scanner scanner = new Scanner(System.in);
3     double c, p;
4
5     System.out.println("Wprowadz liczbę do pierwiastkowania");
6     c = scanner.nextDouble();
7
8     System.out.println("Wprowadz precyzję");
9     p = scanner.nextDouble();
10
11    System.out.println("Wynik pierwiastkowania");
12    System.out.println(pierwiastkowanie(c, p));

```

```
13 }
```

Cały program prezentuje się następująco:

```
1 import java.util.Scanner;
2
3 public class Pierwiastkowanie {
4     public static double pierwiastkowanie(double pierwiastkowany,
5         double a = pierwiastkowany / 2;
6         double b = pierwiastkowany / a;
7
8         while (Math.abs(a - b) > precyzja) {
9             a = (a + b) / 2;
10            b = pierwiastkowany / a;
11        }
12        return a;
13    }
14
15    public static void main(String[] args) {
16        Scanner scanner = new Scanner(System.in);
17        double c, p;
18
19        System.out.println("Wprowadz liczbe do pierwiastkowania")
20        c = scanner.nextDouble();
21
22        System.out.println("Wprowadz precyzje");
23        p = scanner.nextDouble();
24
25        System.out.println("Wynik pierwiastkowania");
26        System.out.println(pierwiastkowanie(c, p));
27    }
28 }
```

W ten sposób zapisaliśmy implementację metody Newtona-Raphsona dla pierwiastków kwadratowych w języku Java.

Słownik

iteracja

(łac. *iteratio* – powtarzanie) wielokrotne wykonywanie tej samej operacji

metoda Newtona-Raphsona

algorytm wyznaczania wartości pierwiastka funkcji, bazujący na wyliczaniu kolejnych przybliżeń, coraz bliższych rzeczywistej wartości pierwiastka; może być wykorzystywany między innymi do wyznaczania wartości pierwiastka kwadratowego

parametr

element składni w określonym języku programowania umożliwiający komunikację pomiędzy podprogramem wywołanym a programem wywołującym; parametry określa się wraz z deklaracją określonego podprogramu w jego nagłówku

warunek stopu

warunek, po którego spełnieniu następuje zakończenie realizacji algorytmu

Film samouczek

Polecenie 1

Zaimplementuj algorytm, którego celem będzie policzenie przybliżonej wartości pierwiastka kwadratowego wybranej liczby przy wykorzystaniu algorytmu Newtona-Raphsona. Program przetestuj dla liczby 8.

Specyfikacja problemu:

Dane:

- a – liczba całkowita
- $\epsilon = 0.01$

Wynik:

- x – przybliżona wartość pierwiastka kwadratowego liczby a

Twoje zadania

1. Oblicz przybliżoną wartość pierwiastka kwadratowego, wykorzystując algorytm Newtona-Raphsona.

```
1 public class Zadanie {
2     public static void main(String args[]) {
3         int a = 8;
4         double epsilon = 0.1;
5         double pierwiastek = 0.0;
6         double x = a;
7
8         // tutaj dopisz kod
9
10        System.out.println("Pierwiastek z " + a + " wynosi: "
11        + pierwiastek);
12    }
13 }
```

Polecenie 2

Porównaj swoje rozwiązanie z przedstawionym w filmie.



Obliczanie wartości pierwiastka kwadratowego

Implementacja algorytmu obliczania wartości pierwiastka kwadratowego w języku Java



Film dostępny pod adresem </preview/resource/RKEXcTtyX2Rhg>

Film przedstawiający obliczanie wartości pierwiastka kwadratowego w języku Java.

Plik o rozmiarze 405.00 B w języku polskim

Polecenie 3

Porównaj implementację zaprezentowaną w filmie z tą w sekcji "Przeczytaj". Jakie różnice dostrzegasz?

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Napisz program, który metodą Newtona-Raphsona obliczy wartość pierwiastka kwadratowego liczby c przy ustalonej precyzji p . Oblicz, ile iteracji wykona algorytm, a następnie podaj wynik pierwiastkowania z dokładnością o wartości 0,01 (bez zaokrąglania). Wypisz wyniki oddzielone pojedynczym znakiem odstępu. Swój program przetestuj dla następujących danych:

- $c = 35$
- $p = 0.0001$

Specyfikacja problemu:

Dane:

- c – liczba podpierwiastkowa; liczba rzeczywista
- p – precyzja, z jaką algorytm powinien wyznaczyć wartość pierwiastka; liczba rzeczywista

Wynik:

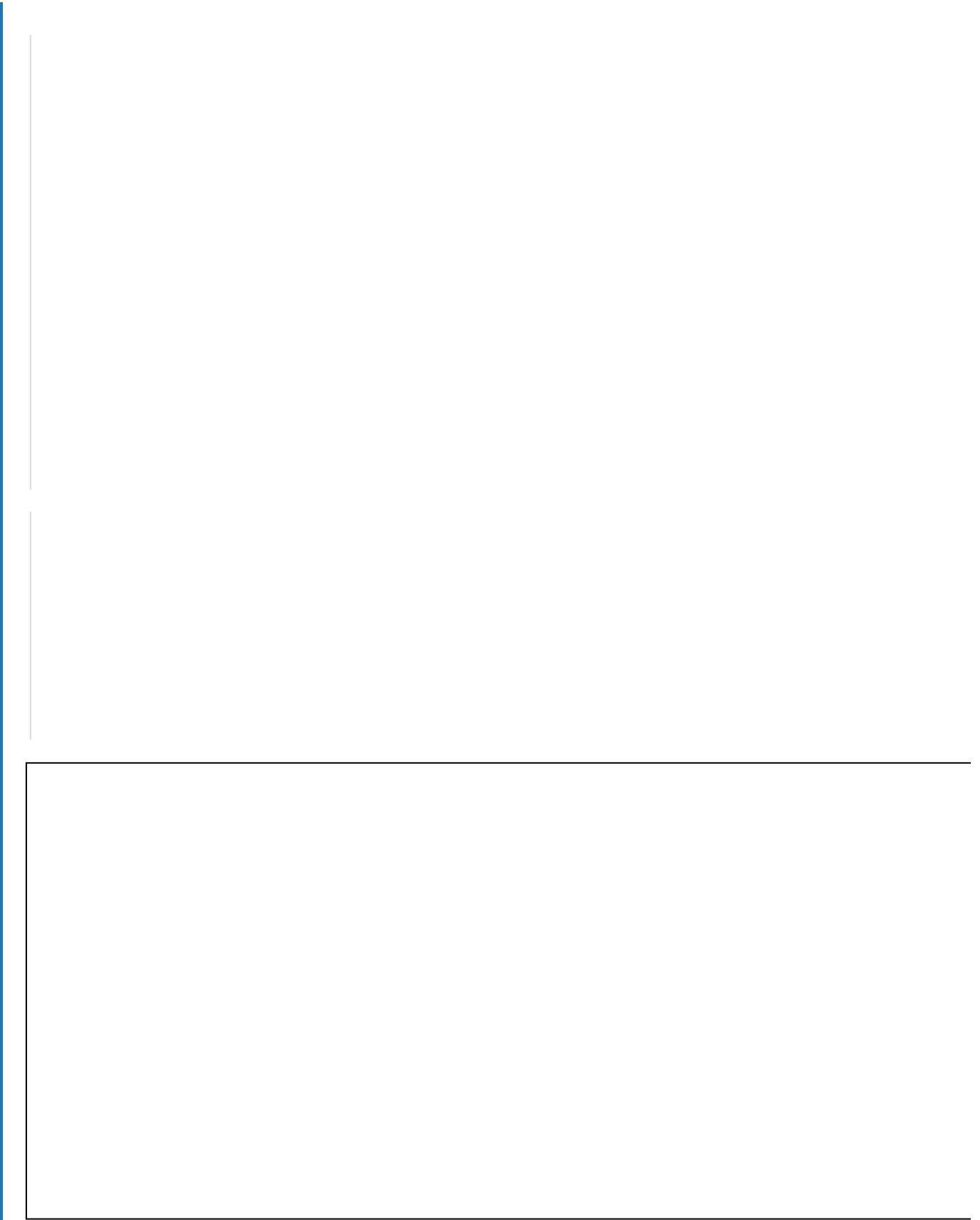
- `iteracje` – liczba iteracji wykonana przez algorytm; liczba naturalna
- `wynik` – wynik pierwiastkowania z dokładnością do 0,01 bez zaokrąglania; liczba rzeczywista

Przykładowe wyjście:

```
1 Liczba iteracji = 5
2 Pierwiastek kwadratowy z liczby 35.0 = 5.91
```

Twoje zadania

1. Program ma obliczyć wartość pierwiastka kwadratowego liczby c przy ustalonej precyzji p i wypisać liczbę wykonanych iteracji oraz wynik pierwiastkowania z dokładnością do 0,01 bez zaokrąglania.



Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Obliczanie przybliżonej wartości pierwiastka kwadratowego z liczby nieujemnej w języku Java

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.

Treści nauczania – wymagania szczegółowe

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):

- Prześledzisz założenia algorytmu Newtona-Raphsona.

- Przeanalizujesz implementację algorytmu Newtona-Raphsona.
- Zapiszesz algorytm w języku Java obliczania przybliżonej wartości pierwiastka kwadratowego.

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;
- oprogramowanie dla języka Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji).

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Obliczanie przybliżonej wartości pierwiastka kwadratowego z liczby nieujemnej w języku Java”. Uczniowie zapoznają się z treściami w sekcji „Przeczytaj” w kontekście programowania.

Faza wstępna:

1. Nauczyciel inicjuje rozmowę wprowadzającą w temat lekcji. Przedstawia cele zajęć oraz kryteria sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel prosi wybranego ucznia lub uczniów o przedstawienie sytuacji problemowej związanej z tematem lekcji.

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 „Film samouczek”. Uczniowie wspólnie zapoznają się z treścią zawartego w niej multimediu. Zapisują ewentualne problemy i pytania. Po czym następuje dyskusja, w trakcie której nauczyciel wyjaśnia niezrozumiałe treści.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują indywidualnie ćwiczenie nr 1, a następnie porównują swoje odpowiedzi z kolegą lub koleżanką.

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. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności, omawia ewentualne problemy podczas rozwiązywania ćwiczeń z programowania w języku Java.

Praca domowa:

1. Uczniowie wykonują program przedstawiony w filmie w sekcji „Film samouczek”.

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.