

TYPY PROSTE I OPERATORY

Ćwiczenia 2

ZADANIE 1. QUIZ

```
5 public class PrimitivesQuiz {
6     public static void main(String[] args) {
7
8         // what will be printed to the console after this program runs:
9
10        //      System.out.println('a' + 'A');
11
12        //      System.out.println("\");
13
14        //System.out.println("what will be the result of this addition? " + 1 + 2 + 3);
15
16        // System.out.println(1 + 2 + 3 + " and what the result will be here?");
17
18        //System.out.println("what will be the result of this addition? " + (1 + 2 + 3));
19
20        //      boolean boolExample = !(true);
21        //
22        //      System.out.println(boolExample);
23        //
24        //      //byte data type, so 8 bit integer
25        ////      //
26        //      byte by = (byte)135;
27        //      System.out.println(by);
28        //
29        //      byte bySecond = (byte)256;
30        //      System.out.println(bySecond);
31        //
```

```
32        //      //why is there a mistake here?
33        //      int i = 12345678901234567;
34
35        //      why is there a mistake here?
36        //      long lg = 12345678901234567;
37
38
39        //what's going on here
40        //      char c = 0x6d;
41        //      char d = 0155;
42        //      System.out.println(c + " = " + d);
43        //
44        //      double test = 2.0 - 1.9;
45        //      System.out.println(test);
46        //
47        ////      // find the value of a variable
48        //      int a = 1000000000;
49        //      System.out.println("Here the variable has a value: " + a);
50        //
51        //      a += 1000000000;
52        //      System.out.println("Here the variable has a value: " + a);
53        //
54        //      a += 1000000000;
55        //      System.out.println("Here the variable has a value: " + a);
56        //
57        //      a += 1000000000;
58        //      System.out.println("Here the variable has a value: " + a);
```

ZADANIE 1. QUIZ

```
60  /// //what will be printed to the console after this program runs:
61  //   System.out.println(2147483647 + 1);
62  //   System.out.println(2147483647L + 1);
63  //
64  //   int value = 12_346_789;
65  //   System.out.println("Value of a variable is: " + value);
66  //
67  int x1 = 40;
68  int y1 = 10;
69
70  ///   x1 += y1;
71  //     System.out.println("Now x1 has value: " + x1);
72  ///
73  //     x1 += y1;
74  //     System.out.println("Now x1 has value: " + x1);
75
76  int x = 899;
77  short y = 10;
78
79  ///   y = x;
80  //     x = y;
81  //     System.out.println(x);
82
83  Integer integerOne = new Integer(value: 100);
84  Integer integerTwo = new Integer(value: 100);
85
```

```
86  ///   System.out.println(integerOne == integerTwo);
87  //
88  //   System.out.println(integerOne.equals(integerTwo));
89  //
90  //   double a = 0.7;
91  //   double b = 0.9;
92  //   double c = a + 0.1;
93  //   double d = b - 0.1;
94  //
95  //   System.out.println(0.8 == 0.8);
96  //   System.out.println(c == d);
97  //
98  //   System.out.println("c = " + c);
99  //   System.out.println("d = " + d );
100
101  int f = 10;
102  AtomicInteger g = new AtomicInteger(initialValue: 10);
103
104  valueTest(f,g);
105
106  ///   System.out.println("f = " + f);
107  //     System.out.println("g = " + g);
108
109  }
110  @ private static void valueTest(int f, AtomicInteger g) {
111      f = 20;
112      g.set(20);
113  }
114
```

ZADANIE 2. ILE WAŻYSZ NA MARSIE?

Grawitacja Marsa stanowi około 38% ziemskiej. Napisz program, który obliczy Twój ciężar na Marsie.

Założenie:

- wszystkie zmienne zadeklaruj na górze metody main,
- wszystkie zmienne oprócz nazwy zgodnej z zasadami „clean code” powinny zawierać również Twoje nazwisko, np.: *float marsWeightBielecki*;
- początkowe zmienne mają być typu *float*,
- po dokonaniu obliczeń dokonaj przypisania wyniku do nowej zmiennej tym razem typu *double*,
- po wykonaniu przypisania **wypisz na konsole** zmienną *double* ograniczając jej długość do 4 miejsc po przecinku,
- dokonaj rzutowania powyższej zmiennej typu *double* na nową zmienną typu *int*,
- dokonaj rzutowania powyższej zmiennej typu *int* na nową zmienną typu *char*,
- wykonaj jakieś działanie matematyczne na tej zmiennej *char* i przypisz wartość tego działania do nowej zmiennej typu *int*.

Każde z powyższych działań wypisz na konsole, dodając przy tym jakiś tekst tłumaczący co zostało zrobione.

ZADANIE 3. PARZYSTA CZY NIE PARZYSTA?

Napisz program który generuje losową liczbę z przedziału od 1 do 100 (możesz użyć metody `Random()` z klasy `Math`, a w następnym kroku sprawdza (dzięki modulo) czy jest to liczba parzysta czy nieparzysta.

Każde z powyższych działań wypisz na konsole, dodając przy tym jakiś tekst tłumaczący co zostało zrobione.

ZADANIE 4. ILE POJEMNIKÓW NA KLOCKI POTRZEBUJE?

Wybierz liczbę nieparzystą z przedziału od 50 do 100, i zapisz ją jako zmienną `int` mówiącą nam ile klocków Lego posiadamy (np. `amountOfBricks`), następnie wybierz liczbę parzystą z przedziału od 5 do 10 stwierdzającą ile klocków Lego mieści się w jednym pojemniku (np.: `containerCapacity`) i zapisz ją również jako zmienną `int`.

Napisz program który obliczy ile mamy pełnych pojemników, a ile pojemników w ogóle i ile klocków jest w pojemniku który nie został wypełniony w całości (użyj do tego operatora `%` czyli modulo).

Każde z powyższych działań wypisz na konsoli, dodając przy tym jakiś tekst tłumaczący co zostało zrobione.

ZADANIE 5. JAK NIE OBLICZAĆ WARTOŚCI WYMAGAJĄCYCH PRECYZJI W JĘZYKU JAVA

Przy użyciu typów double zaimplementuj poniższe:

Założmy że jakiś produkt kosztuje 9.99 netto, oblicz jego wartość brutto (zakładamy VAT 23%). Następnie pomnóż ją przez 10000 (czyli np. sprzedaliśmy 10000 szt tego produktu), i oblicz tę wartość bez VAT.

Zaimplementuj powyższe działania używając klasy Big Decimal. Wypisz na konsole wszystkie wyliczane wartości, porównaj ich wartości, jakie wnioski?

ZADANIE 5. JAK NIE OBLICZAĆ WARTOŚCI WYMAGAJĄCYCH PRECYZJI W JĘZYKU JAVA – SUGESTIE Z CHAT GPT

New chat

What's the best way to store a floating-point value which is the price of a product in Java?

In Java, the best way to store a floating-point value that represents the price of a product is to use the `double` data type.

The `double` data type is a 64-bit floating-point value that can represent decimal numbers with a high degree of precision. It is capable of representing a wide range of values, including very small values (e.g., 4.9E-324) and very large values (e.g., 1.7976931348623157E308).

To store the price of a product using a `double` variable, you can declare a variable like this:

```
java
double price = 19.99;
```

This will create a variable named `price` and assign it the value of `19.99`. You can then use this variable in your program to perform calculations or display the price to the user.

Note that when working with floating-point values, it's important to be aware of the limitations of floating-point arithmetic, such as rounding errors and imprecise representations of certain values. In cases where exact decimal representation is needed, you may want to consider using a different data type, such as `BigDecimal`.

Regenerate response

ChatGPT Feb 13 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.

New chat

Cool! Now show me how to store this value using the Big Decimal class.

Sure! Here's an example of how to store a price using the `BigDecimal` class in Java:

```
java
import java.math.BigDecimal;

public class Product {
    private BigDecimal price;

    public Product(BigDecimal price) {
        this.price = price;
    }

    public BigDecimal getPrice() {
        return price;
    }

    public void setPrice(BigDecimal price) {
        this.price = price;
    }
}
```

In this example, we've created a `Product` class with a private `BigDecimal` field named `price`. We've also provided a constructor and getter/setter methods for accessing the `price` field.

Regenerate response

To create a `BigDecimal` object to store the price, you can use one of the many constructors.

ChatGPT Feb 13 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.

DZIĘKUJĘ

Więcej na:

www.vistula.edu.pl



Akademia Finansów i Biznesu Vistula
ul. Stokłosy 3
02-787 Warszawa
(obok stacji metro Stokłosy)