JAVA klausimai pakartojimui

Mindaugas Karpinskas
2017

https://docs.oracle.com/javase/tutorial/java/nutsandbolts/QandE/questions_variables.html

Bus Teste!!!

❑ static, non-static

❑ Metodu perkrova

❑ Konstruktoriai, kai vienas kviečia kitą

❑ Masyvai: pvz gauti pirmą, paskutini elementą

❑ Artmetiniai operatoriai +,-,%,++,--...

❑ If (..., boolean laukai

❑ Ciklai

❑ String: substring, length, charAt, equals. Immutable

❑ Ir kt

TODO

- ❏ Next Prime Number – Have the program find prime numbers until the user chooses to stop the asking for the next one.
- ❏ Unit Converter (temp, currency, volume, mass and more) – Converts various units between one another. The user enters the type of unit being entered, the type of unit they want to convert to and then the value. The program will then make the conversion.
- ❏ Credit Card Validator – Takes in a credit card number of a common credit card vendor (Visa, MasterCard, American Express, Discoverer) and validates it to make sure that it is a valid number (look into how credit cards use a checksum).
- ❏ Count Vowels – Enter a string and the program counts the number of vowels in the text. For added complexity have it report a sum of each vowel found. Prints out text without vowels.
- ❏ Count Words in a String – Counts the number of individual words in a string. Words can be separated by any non-letter simbol. Count words only with starts Uppercase.
- ❏ Quote Tracker (market symbols etc) – A program which can go out and check the current value of stocks for a list of symbols entered by the user. The user can set how often the stocks are checked and the program can show green up and red down arrows to show which direction the stock value has moved.
- ❏ Bank Account Manager - Create a class called “Account” which will be an abstract class for three other classes called “CheckingAccount”, “SavingsAccount” and “BusinessAccount”. Manage credits and debits from these accounts through an ATM style program.

The term "instance variable" is another name for ___.

The term "instance variable" is another name for ___.

non-static field

```
public int value = 10;
```

The term "class variable" is another name for ___

The term "class variable" is another name for ____ .

static field

```
public static int GLOBALUS_SKAICIUS = 10;
```

A local variable stores temporary state; it is declared inside a ____.

A local variable stores temporary state; it is declared inside a ____.

method

```
private void padidink(){
    int laikinasKintamasis = 10;
    value += laikinasKintamasis;
}
```

A variable declared within the opening and closing parenthesis of a method is called a ______ .

?

```
public void keisti(int a, int b) {
    // pirmas = a;
    antras = b;
    System.out.println("Pirmas: " + pirmas);
    System.out.println("Antras: " + antras);
}
```

A variable declared within the opening and closing parenthesis of a method is called a ______ .

Parameter or arguments

```
private void padidink(int parametras){
    value += parametras;
}
```

https://docs.oracle.com/javase/tutorial/java/

What are the eight primitive data types supported by the Java programming language?

What are the eight primitive data types supported by the Java programming language?

byte, short, int, long, float, double, boolean, char

Character strings are represented by the class ___ .

Character text are represented by the class ____ .

java.lang.**String**

```
private String tekstas = "Labas, Žemė!";
```

An ____ is a container object that holds a fixed number of values of a single type.

An **array** is a container object that holds a fixed number of values of a single type.

array

```
private int[] masyvas = new int[10];
```

Consider the following code snippet.

```java
public static void main(String[] args) {

    int i = 10;
    int n = i++ % 5;   // 2.

    System.out.println("i = " + i + ", n = " + n);
}
```

1. **What are the values of `i` and `n` after the code is executed?**

2. **What are the final values of `i` and `n` if instead of using the postfix increment operator (`i++`), you use the prefix version (`++i`)?**

Consider the following code snippet.

```java
public static void main(String[] args) {
    int i = 10;
    int n = i++ % 5;
    System.out.println("i = " + i + ", n = " + n);
}
```

1. **Question:** What are the values of `i` and `n` after the code is executed?

   Answer: `i` is 11, and `n` is 0.

2. **Question:** What are the final values of `i` and `n` if instead of using the postfix increment operator (`i++`), you use the prefix version (`++i`))?

   Answer: `i` is 11, and `n` is 1.

Which operator is used to compare two values: = or == ?

= or == ?

In the following program, explain why the value "6" is printed twice in a row

```
public static void main(String[] args) {
    int i = 3;
    i++;
    System.out.println(i);
    ++i;
    System.out.println(i);
    System.out.println(++i);
    System.out.println(i++);
    System.out.println(i);
}
```

The code `System.out.println(++i);` evaluates to 6, because the prefix version of `++` evaluates to the incremented value. The next line, `System.out.println(i++);` evaluates to the current value (6), then increments by one. So "7" doesn't get printed until the next line.

In the following program, explain why the value "6" is printed twice in a row

```java
public static void main(String[] args) {
    int i = 3;
    i++;
    System.out.println(i); // "4"
    ++i;
    System.out.println(i); // "5"
    System.out.println(++i); // "6"
    System.out.println(i++); // "6"
    System.out.println(i); // "7"
}
```

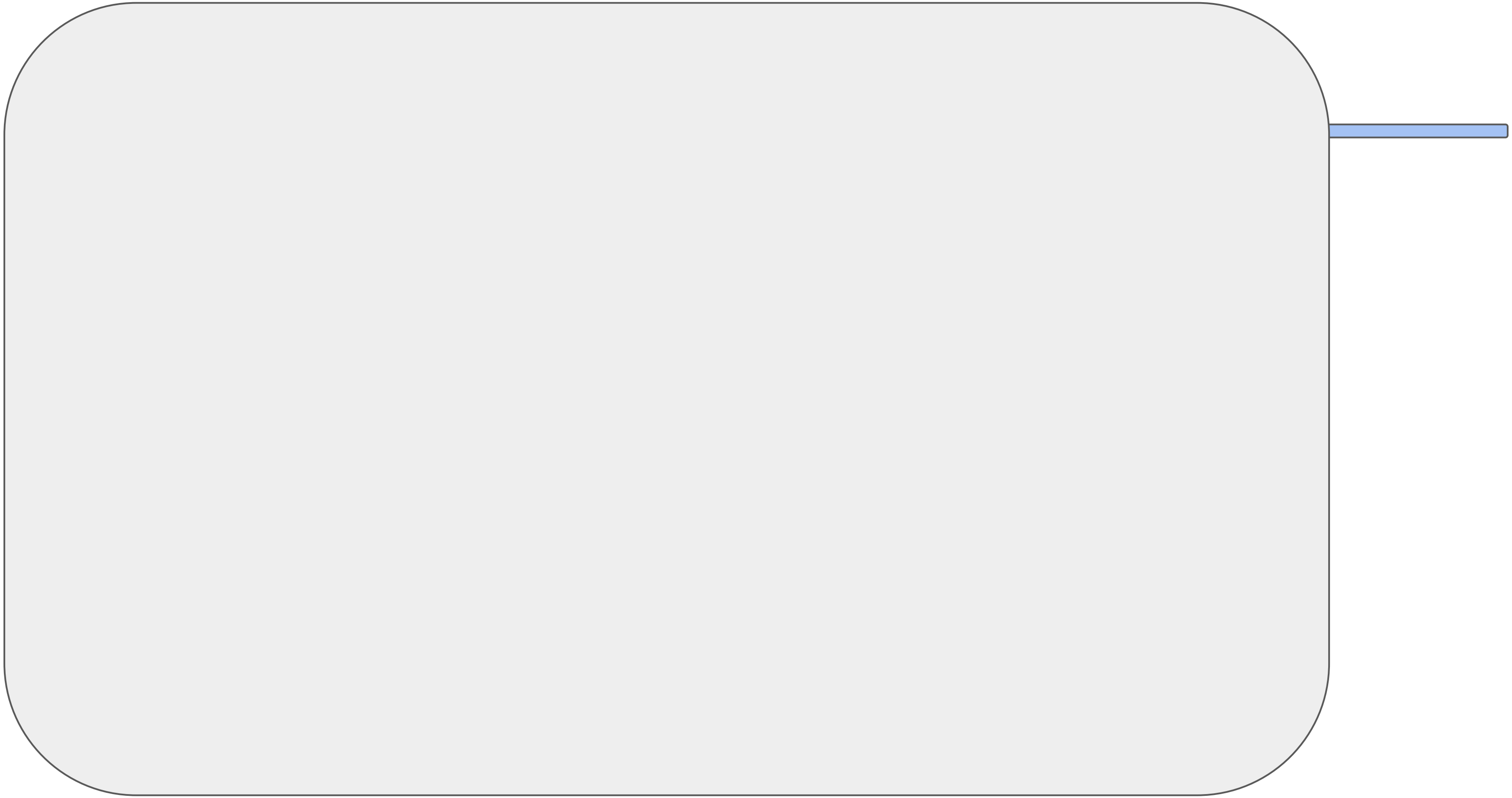
The code `System.out.println(++i);` evaluates to 6, because the prefix version of `++` evaluates to the incremented value. The next line, `System.out.println(i++);` evaluates to the current value (6), then increments by one. So "7" doesn't get printed until the next line.

Consider the following class:

```
public class IdentifyMyParts {
    public static int x = 7;
    public int y = 3;
}
```

1. What are the class variables?
2. What are the instance variables?

K



Išvestis?

```
class F {
  int a;
  public static void main(String[] args){
    System.out.println();

    F f1 = new  F();
    System.out.println(f1.a);

    f1.a = 1;
    F f2 = new  F(); f2.a = 2;

    System.out.println(f1.a == f2.a);
    System.out.println(f1 == f2);

    f2.a = 1;
    System.out.println(f1.a == f2.a);
    System.out.println(f1 == f2);

    f1 = f2; f2.a = 2;
    System.out.println(f1.a == f2.a);
    System.out.println(f1 == f2);
  }
}
```

Klausimai

?