

Проверки

Логически изрази и проверки. Условна конструкция
if-else



True

Condition

Do this

False

Do that or
do nothing

СофТУни

Преподавателски екип



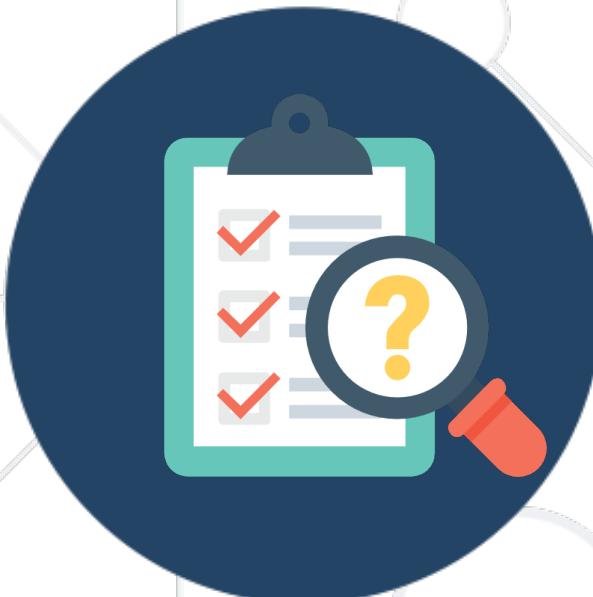
SoftUni



Software University
<https://softuni.bg>

- Преговор
- Логически изрази и проверки
- Оператори за сравнение
- Условни конструкции
- Закръгляне и форматиране
- Серия от проверки
- Живот на променлива
- Решаване на изпитна задача



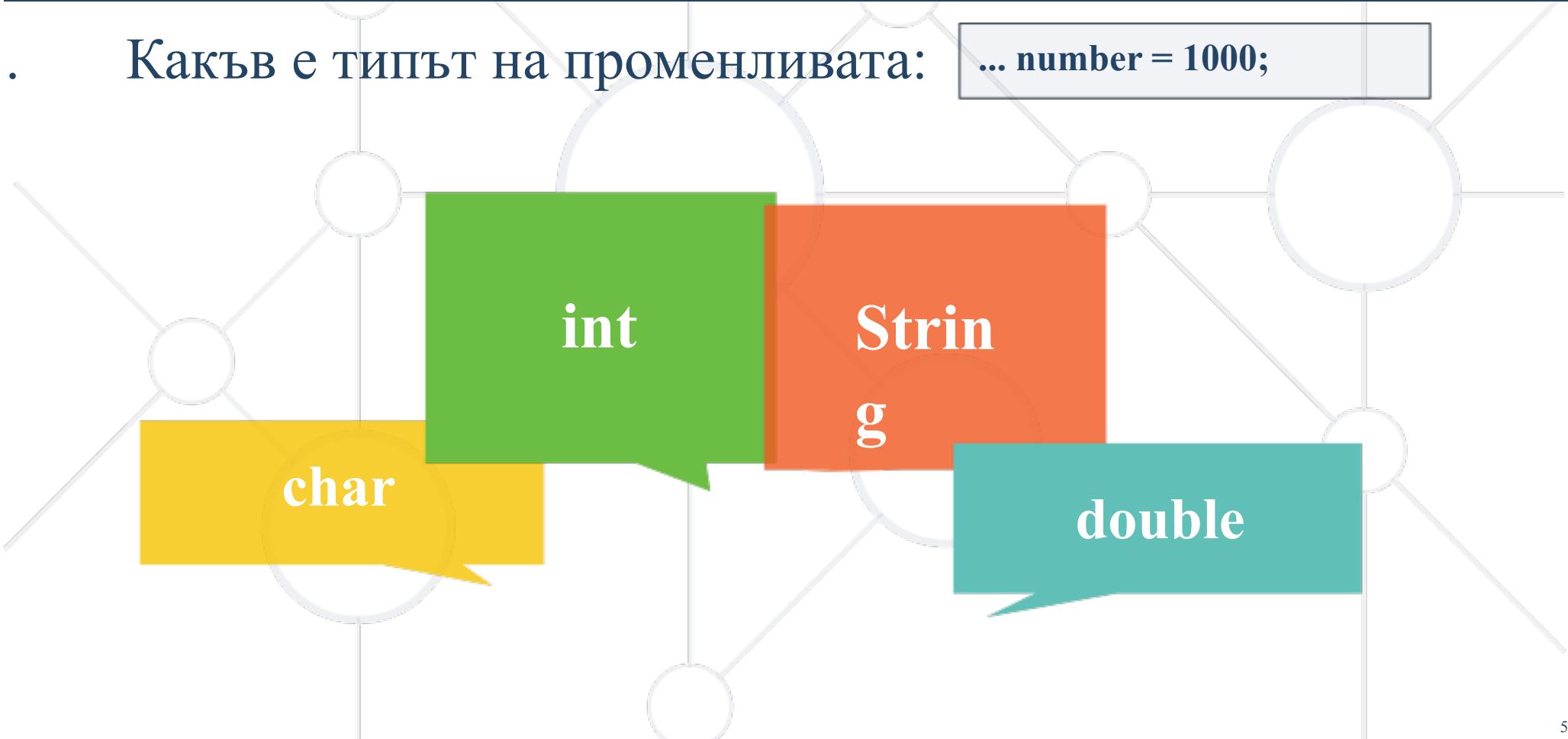


Преговор

- Какъв е типът на променливата:
`... number = "1000";`



- . Какъв е типът на променливата: ... number = 1000;



- . Как се нарича долепването на два текста (низа)?

Събиран
е

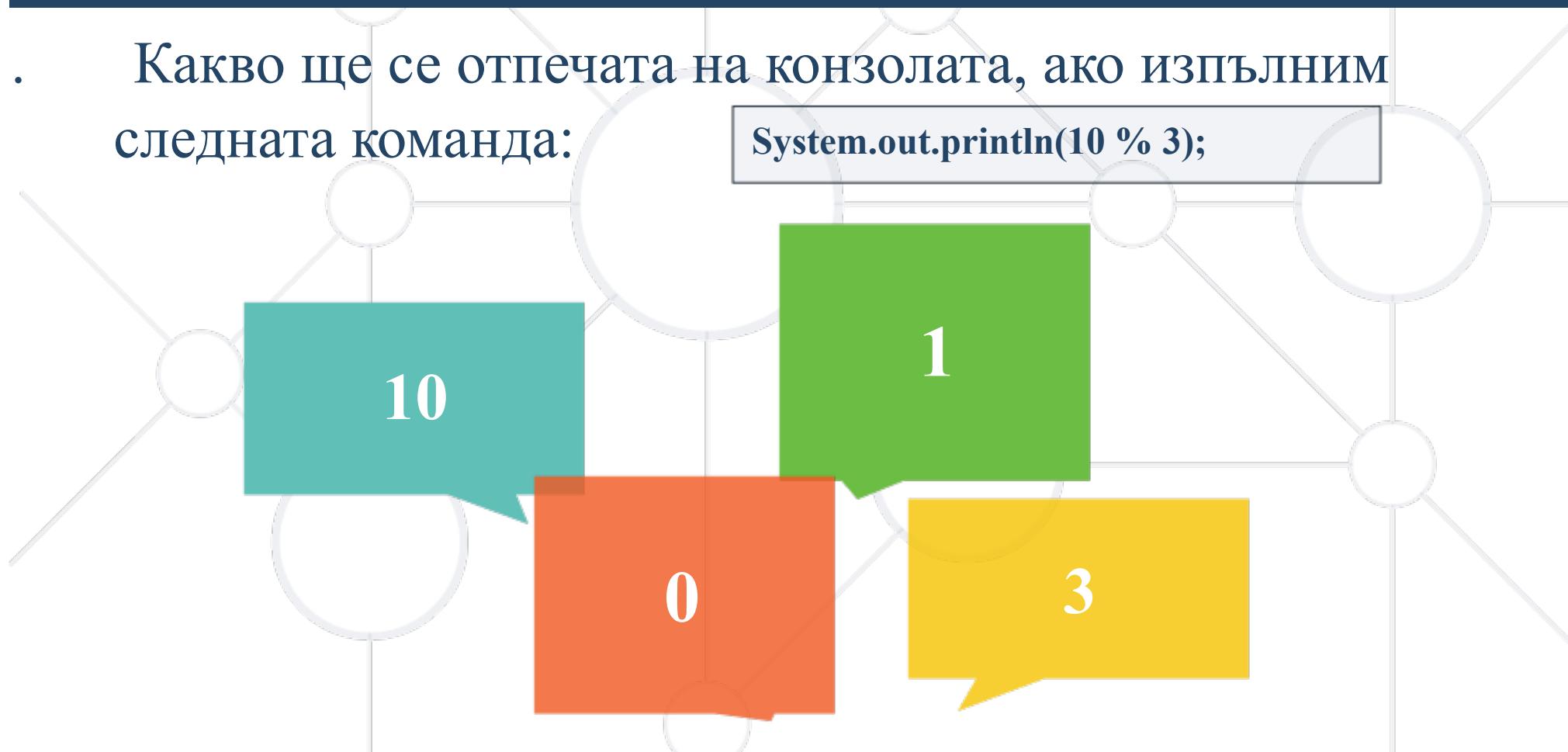
Конкатенаци
я

Кулминаци
я

Съединяване

- Какво ще се отпечата на конзолата, ако изпълним следната команда:

```
System.out.println(10 % 3);
```



10

0

1

3

- Каква стойност държи променливата **result**:

```
int a = 5;  
int b = 2;  
double result = a / b;
```

2.5

7

1

2.0

- . Какъв би бил резултатът, ако се опитаме да изпълним следната команда:

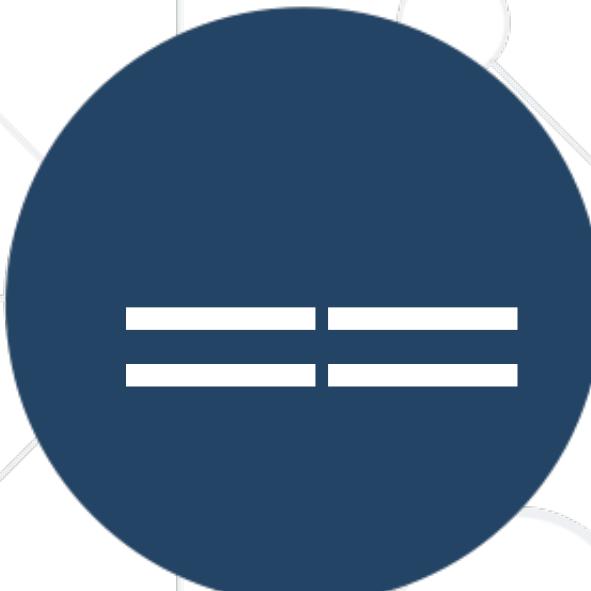
```
System.out.println(1 + 1 + "4" + 2 + 1);
```

Compile
time error

9

243

2421



Логически изрази и проверки

Оператори за сравнение

Оператори за сравнение



Оператор	Означение	Работи за
Равенство	$==$	
Различно	$!=$	
По-голямо	$>$	
По-голямо или равно	\geq	
По-малко	$<$	
По-малко или равно	\leq	числа, символи, други сравними типове

Сравняване на стойности (1)

- В програмирането можем да сравняваме стойности
- Резултатът от логическите изрази е **true** или **false**

```
int a = 5;  
int b = 10;  
  
System.out.println(a < b);  
System.out.println(a > 0);  
System.out.println(a > 100);  
  
System.out.println(a < a);  
System.out.println(a <= 5);  
  
System.out.println(b == 2 * a);
```

// true
// true
// false
// false
// true
// true



Сравняване на стойности (2)

Сравняване на текст чрез $==$ по адрес в паметта

```
String a = "Example";
```

```
String b = a;
```

```
System.out.println(a == b); // true
```

```
Scanner scanner = new Scanner(System.in);
```

```
String a = scanner.nextLine();
```

```
String b = scanner.nextLine();
```

```
System.out.println(a == b); // false
```

Въвеждане на
еднаква стойност

Сравняване на стойности (3)

Променливи от тип **String** сравняваме чрез метода **equals**

Сравняване на текст чрез **equals** по стойност:

```
Scanner scanner = new Scanner(System.in);  
String a = scanner.nextLine();  
String b = scanner.nextLine();  
System.out.println(a.equals(b));
```

Въвеждане на
еднаква стойност

// *true*

- **boolean** – ключова дума, с която се инициализира булева променлива
- Има само следните две стойности **true** (вярно) или **false** (грешно)
- Може да се създаде и с условие, което се свежда до true или false

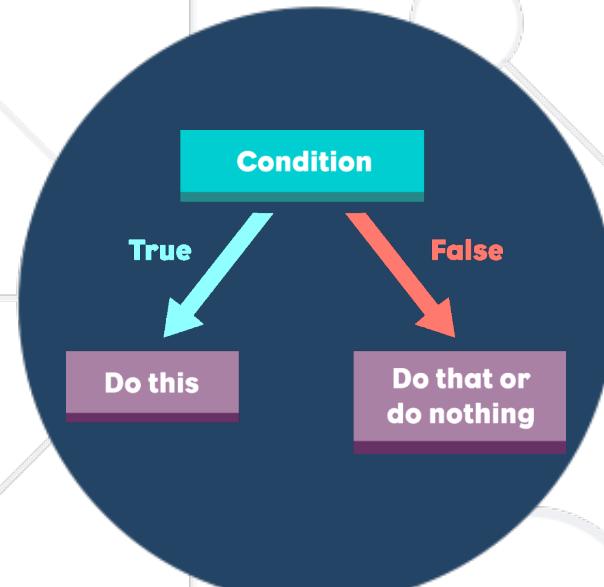
```
boolean isValid = true;
```

```
boolean isPositive = a > 0;
```

Булева променлива - Пример

```
int a = 5;  
boolean isPositive = a > 0;  
System.out.println(isPositive); // true
```

```
int a = -5;  
boolean isPositive = a > 0;  
System.out.println(isPositive); // false
```



Условни конструкции

Прости проверки

Прости проверки

- Често проверяваме условия и извършваме действия според резултата



Условие
(булев израз)

```
if (...) {  
    // код за изпълнение  
}
```

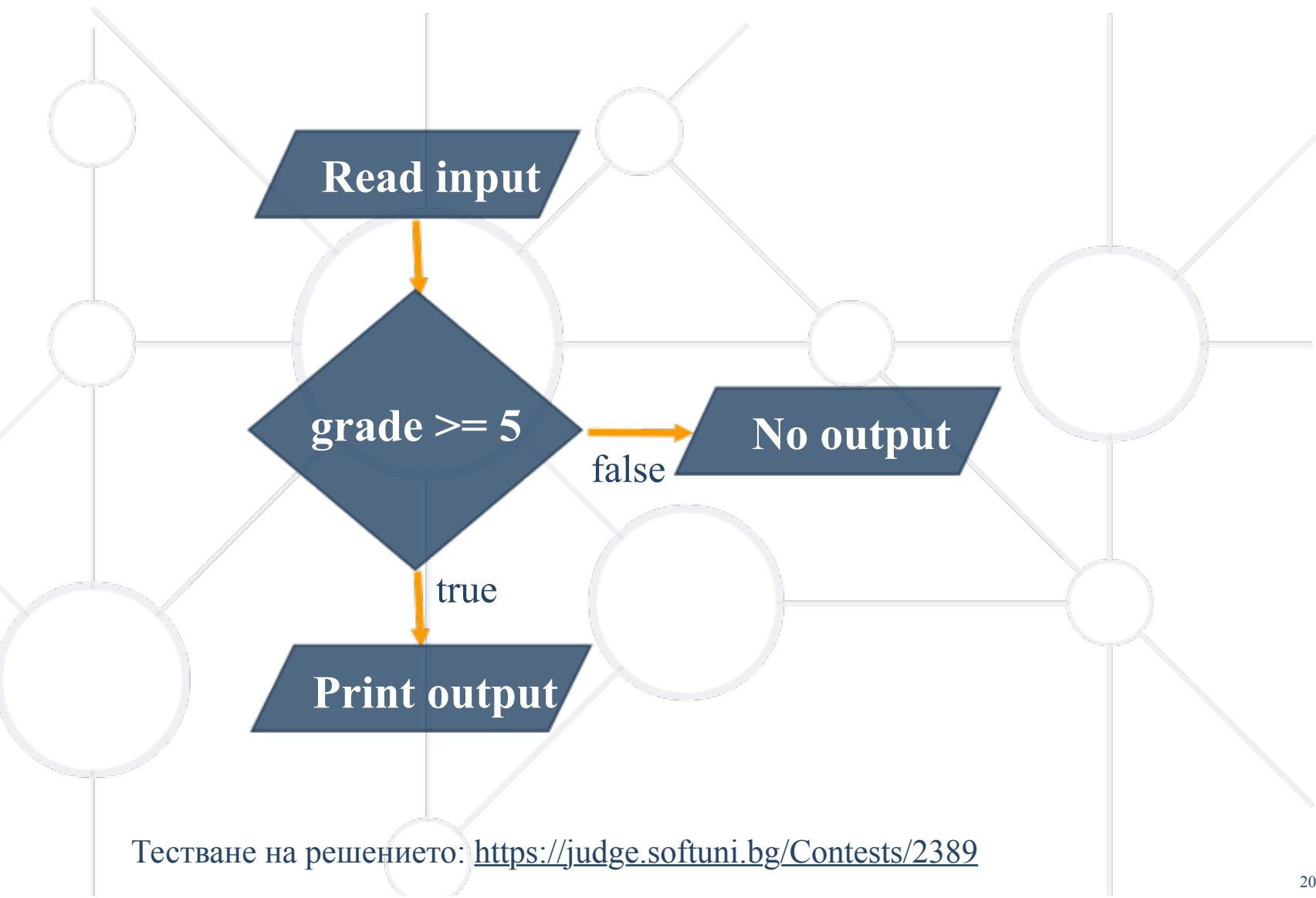
Код за изпълнение при
вярност на условието

- Резултатът е **true** или **false**

Отлична оценка - условие

- Напишете **програма**, която:
 - Чете оценка (**число**), въведена от потребителя
 - Проверява дали е отлична
 - Отпечатва на конзолата "Excellent", ако оценката е по-голяма или равна на 5
- Пример:





Тестване на решението: <https://judge.softuni.bg/Contests/2389>

Прости проверки - if-else

- При **невярност (false)** на условието, можем да изпълним други действия - чрез **else** конструкция



```
if (...) {  
    // код за изпълнение  
} else {  
    // код за изпълнение  
}
```

Код за изпълнение
при невярност на
условието

Блок от код (1)

- Къдравите скоби {} въвеждат блок (група команди)

```
String color = "red";
if (color.equals("red"))
    System.out.println("tomato");
else
    System.out.println("banana");
System.out.println("bye");
```

```
"C:\Program Files\Java\jdk-11.0.2\bin>
tomato
bye
Process finished with exit code 0
```

Изпълнява се винаги - не е
част от if/else конструкцията

Блок от код (2)

Ако **включим скоби**, се изпълнява съответния блок

```
String color = "red";
if (color.equals("red")) {
    System.out.println("tomato");
} else {
    System.out.println("banana");
    System.out.println("bye");
}
```

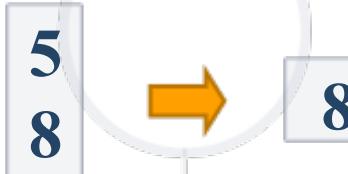
Изпълняват се редовете в
съответния блок

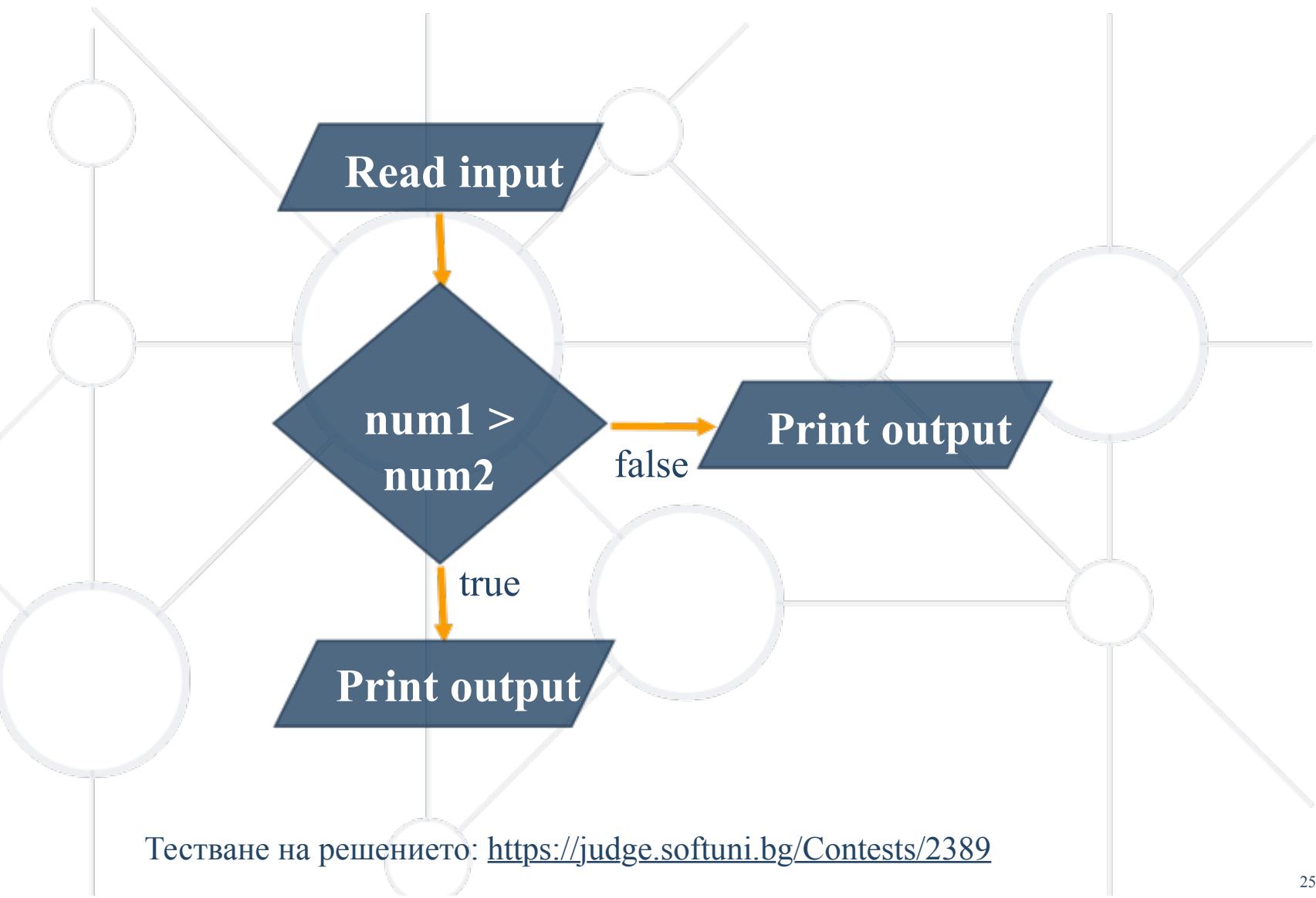
"C:\Program Files\Java\jdk-11.0.2
tomato

Process finished with exit code 0

По-голямото число - условие

- Напишете програма, която:
 - Чете две **цели** числа
 - Извежда "Greater number: "
 - Отпечатва на конзолата **по-голямото** от тях
- Пример:





Тестване на решението: <https://judge.softuni.bg/Contests/2389>

Четно или нечетно число - условие

- Напишете програма, която:
 - Проверява дали едно число е **четно** или **нечетно**
 - Ако е четно отпечатва на конзолата "even"
 - Ако е нечетно отпечатва на конзолата "odd"
- Пример:



Четно или нечетно - решение

```
Scanner scanner = new Scanner(System.in);
int num = Integer.parseInt(scanner.nextLine());
if (num % 2 == 0) {
    System.out.println("even");
} else {
    System.out.println("odd");
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/2389>



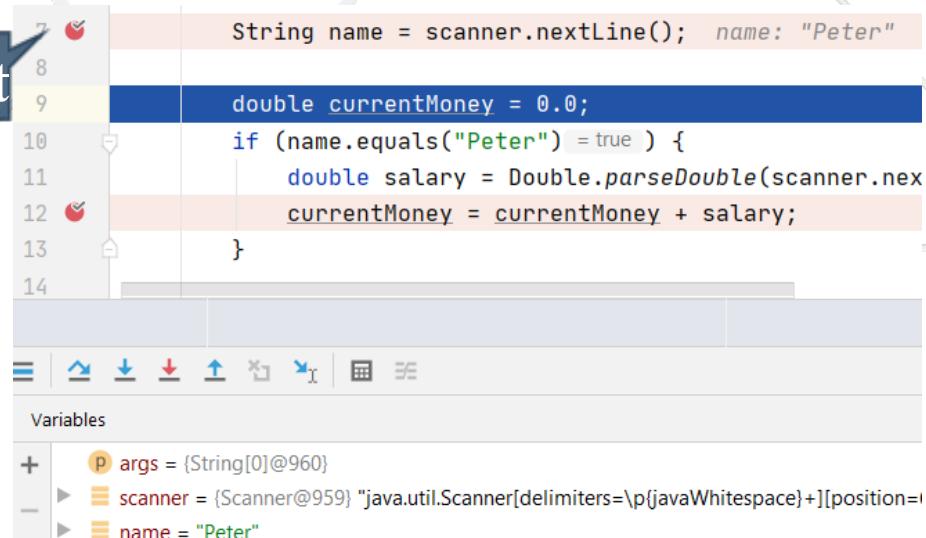
Дебъгване

Прости операции с дебъгер

Дебъгване

- Процес на проследяване на изпълнението на програмата
- Това ни позволява да откриваме грешки (бъгове)

Breakpoint



The screenshot shows a Java debugger interface. The code editor displays the following code:

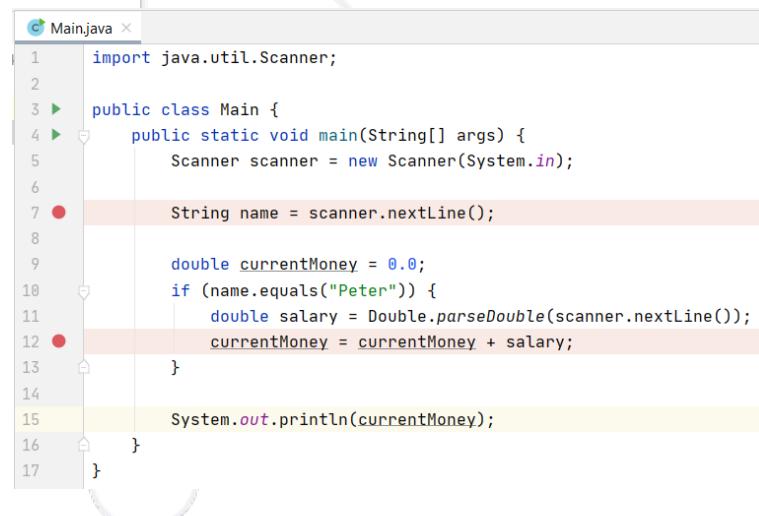
```
String name = scanner.nextLine(); name: "Peter"
double currentMoney = 0.0;
if (name.equals("Peter") = true ) {
    double salary = Double.parseDouble(scanner.nex
currentMoney = currentMoney + salary;
}
```

The line `if (name.equals("Peter") = true) {` is highlighted in blue, indicating it is the current line of execution. The word `name` is underlined in red, indicating it is a variable being watched. The variables panel at the bottom shows:

args	{String[0]@960}
scanner	{Scanner@959} "java.util.Scanner[delimiters=\p{javaWhitespace}+][position=1]
name	"Peter"

Дебъгване в IntelliJ IDEA

- Натискане на **[Shift + F9]** ще стартира програмата в debug режим
- Можем да преминем към следващата стъпка с **[F8]**
- Можем да създаваме **[Ctrl + F8]** стопери - breakpoints
 - До тях можем директно да стигнем използвайки **[F9]**



```
Main.java x
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         String name = scanner.nextLine();
8
9         double currentMoney = 0.0;
10        if (name.equals("Peter")) {
11            double salary = Double.parseDouble(scanner.nextLine());
12            currentMoney = currentMoney + salary;
13        }
14
15        System.out.println(currentMoney);
16    }
17}
```

The screenshot shows the IntelliJ IDEA interface with the Main.java file open. Two red circular markers are placed on the code at line 7 and line 12, indicating breakpoints. The line 'String name = scanner.nextLine();' is highlighted in pink, and the line 'currentMoney = currentMoney + salary;' is also highlighted in pink. The line 'System.out.println(currentMoney);' is highlighted in yellow. A vertical toolbar on the left side of the editor has a yellow circle icon with two black arrows pointing in opposite directions, which corresponds to the 'Step Over' (F8) function.



Закръгляне и Форматиране

- В програмирането можем да закръгляме дробни числа

- Закръгление до следващо (по-голямо) цяло число:

```
double up = Math.ceil(23.45); // 24.0
```

- Закръгление до предишно (по-малко) цяло число:

```
double down = Math.floor(45.67); // 45.0
```

- Намиране на абсолютна стойност

```
int example1 = Math.Abs(-50); // 50
```

```
int example2 = Math.Abs(50); // 50
```



- Форматиране до 2 знака след десетичната запетая:

```
System.out.printf("%.2f", 123.456);
```

// 123.46

Брой символи след
десетичната запетая



Серии от проверки
По-сложни условни конструкции

Серии от проверки

Конструкцията **if/else - if/else...** е серия от проверки



```
if (...)  
    // код за изпълнение  
else if (...)  
    // код за изпълнение  
else if (...)  
    // код
```

TRUE  FALSE?

При истинност на едно условие, **не се продължава** към проверяване на следващите условия

Серия от проверки - пример

- Програмата проверява **първото условие**, установява, че е вярно, и приключва

```
int a = 7;  
if (a > 4)  
    System.out.println("Bigger than 4");  
else if (a > 5)  
    System.out.println("Bigger than 5");  
else  
    System.out.println("Equal to 7");
```

Извежда на конзолата само "Bigger than 4"



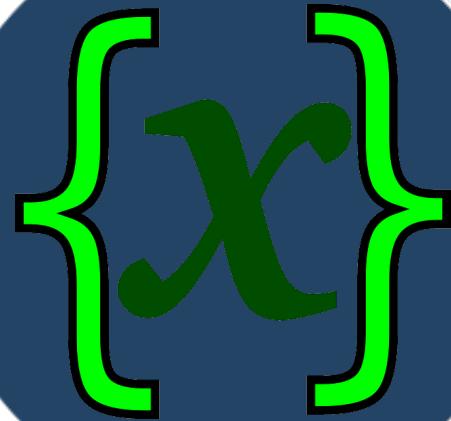
Серия от проверки - пример

- Програмата проверява **всяко едно условие**, установява дали е вярно, и приключва

```
int a = 7;  
if (a > 4)  
    System.out.println("Bigger than 4");  
if (a > 5)  
    System.out.println("Bigger than 5");  
if (a == 7)  
    System.out.println("Equal to 7");
```

Извежда на конзолата трите резултата
"Bigger than 4"
"Bigger than 5"
"Equal to 7"





Живот на променлива

Диапазон на използване

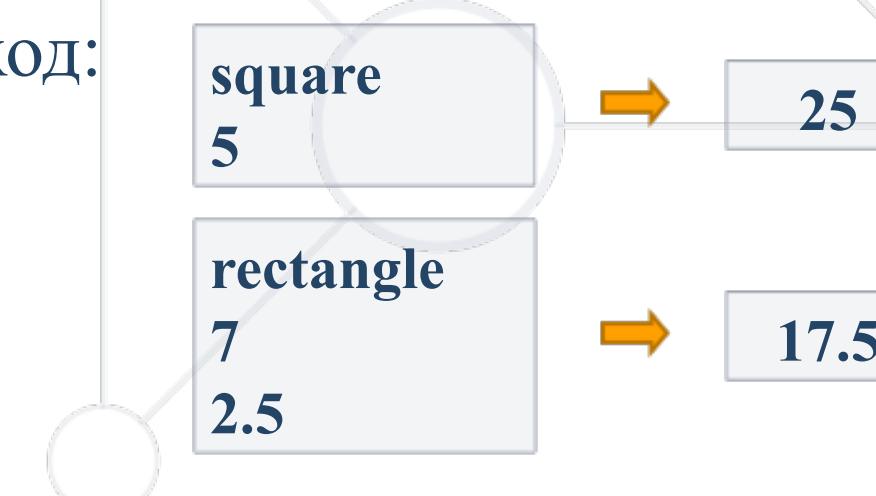
- Обхват, в който може да бъде използвана
- Пример: Променливата **salary** съществува **само** в блока от код на **if**-конструкцията

```
String currentDay = "Monday";
if (currentDay.equals("Monday")) {
    double salary = Double.parseDouble(scanner.nextLine());
}
System.out.println(salary); // Error
```



Лица на фигури - условие

- Напишете програма, която:
 - Прочита **вид на геометрична фигура** ("square", "rectangle", "circle" или "triangle")
 - Пресмята **лицето** спрямо вида на фигурата
- Примерен вход и изход:



Лица на фигури - решение

```
String shape = scanner.nextLine();
double area = 0.0;
if (shape.equals("square")) {
    double side = Double.parseDouble(scanner.nextLine());
    area = side * side;
} else if (shape.equals("rectangle")) {
    double sideA = Double.parseDouble(scanner.nextLine());
    double sideB = Double.parseDouble(scanner.nextLine());
    area = sideA * sideB;
} //TODO: add more conditions
System.out.println(area);
```

Тестване на решението: <https://judge.softuni.bg/Contests/2389>



- Конструкции за проверка на условие - **if** и **if-else**
- Дебъгване
- Закръгляне и форматиране
- Живот на променливата



Въпроси?



SoftUni



Software
University



SoftUni
Svetlina



SoftUni
Creative



SoftUni
Digital



SoftUni
Foundation



SoftUni
Kids

- Този курс (презентации, примери, демонстрационен код, упражнения, домашни, видео и други активи) представлява **защитено авторско съдържание**
- Нерегламентирано копиране, разпространение или използване е незаконно
- © СофтУни - <https://softuni.org>
- © Софтуерен университет - <https://softuni.bg>



Обучения в Софтуерен университет (Софтуни)



- Софтуерен университет - качествено образование, професия и работа за софтуерни инженери
 - softuni.bg
- Фондация "Софтуерен университет"
 - softuni.foundation
- Софтуерен университет @ Facebook
 - facebook.com/SoftwareUniversity
- Дискусионни форуми на Софтуни
 - forum.softuni.bg

