

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности
Высшая школа «Компьютерных технологий и информационных систем»

ОТЧЕТ

по дисциплине «Системный подход в разработке программного обеспечения»

Выполнил:

студент группы 5130902/20201

_____ А. И. Сафонов
подпись

Проверил:

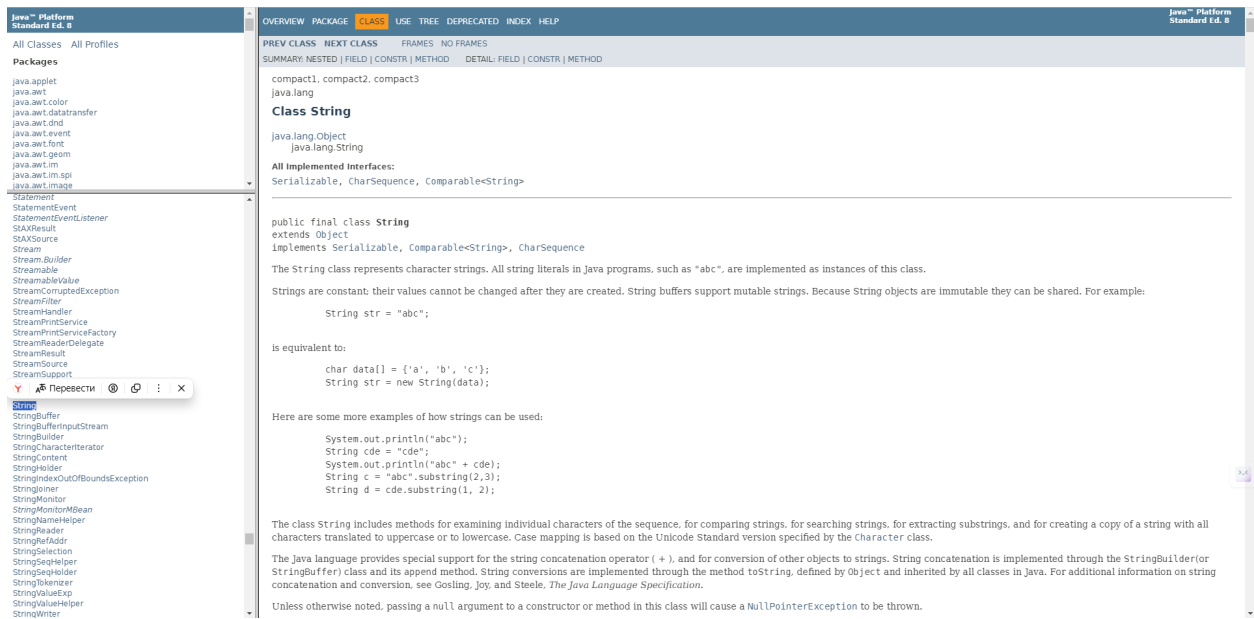
Кандидат тех. Наук., доцент

_____ С. А. Нестеров
подпись

«__» _____ 2024г.

1. Задания по уроку 4.3.

1) Работа с документацией класса String.



На снимке экрана показаны три главных панели веб-страницы:

- на левой верхней панели можно выбрать пакет;
- на левой нижней панели перечислены классы пакета, на этой панели выбран класс `String`;
- на главной панели справа отображаются сведения о классе `String`.

Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods	Deprecated Methods
Modifier and Type		Method and Description		
char		<code>charAt(int index)</code>	Returns the char value at the specified index.	
int		<code>codePointAt(int index)</code>	Returns the character (Unicode code point) at the specified index.	
int		<code>codePointBefore(int index)</code>	Returns the character (Unicode code point) before the specified index.	
int		<code>codePointCount(int beginIndex, int endIndex)</code>	Returns the number of Unicode code points in the specified text range of this <code>String</code> .	
int		<code>compareTo(String anotherString)</code>	Compares two strings lexicographically.	
int		<code>compareToIgnoreCase(String str)</code>	Compares two strings lexicographically, ignoring case differences.	
String		<code>concat(String str)</code>	Concatenates the specified string to the end of this string.	
boolean		<code>contains(CharSequence s)</code>	Returns true if and only if this string contains the specified sequence of char values.	
boolean		<code>contentEquals(CharSequence cs)</code>	Compares this string to the specified <code>CharSequence</code> .	
boolean		<code>contentEquals(StringBuffer sb)</code>	Compares this string to the specified <code>StringBuffer</code> .	
static String		<code>copyValueOf(char[] data)</code>	Equivalent to <code>valueOf(char[])</code> .	
static String		<code>copyValueOf(char[] data, int offset, int count)</code>	Equivalent to <code>valueOf(char[], int, int)</code> .	
boolean		<code>endsWith(String suffix)</code>	Tests if this string ends with the specified suffix.	

Также на главной панели представлен список методов класса. Имя каждого метода и типы его параметров представляют собой гиперссылки для просмотра дополнительных сведений.

charAt

```
public char charAt(int index)
```

Returns the char value at the specified index. An index ranges from 0 to length() - 1. The first char value of the sequence is at index 0, the next at index 1, and so on, as for array indexing.

If the char value specified by the index is a surrogate, the surrogate value is returned.

Specified by:

charAt in interface CharSequence

Parameters:

index - the index of the char value.

Returns:

the char value at the specified index of this string. The first char value is at index 0.

Throws:

IndexOutOfBoundsException - if the index argument is negative or not less than the length of this string.

indexOf

```
public int indexOf(int ch,  
                  int fromIndex)
```

Returns the index within this string of the first occurrence of the specified character, starting the search at the specified index.

If a character with value ch occurs in the character sequence represented by this String object at an index no smaller than fromIndex, then the index of the first such occurrence is returned. For values of ch in the range from 0 to 0xFFFF (inclusive), this is the smallest value k such that

```
(this.charAt(k) == ch) && (k >= fromIndex)
```

is true. For other values of ch, it is the smallest value k such that

```
(this.codePointAt(k) == ch) && (k >= fromIndex)
```

is true. In either case, if no such character occurs in this string at or after position fromIndex, then -1 is returned.

There is no restriction on the value of fromIndex. If it is negative, it has the same effect as if it were zero: this entire string may be searched. If it is greater than the length of this string, it has the same effect as if it were equal to the length of this string: -1 is returned.

All indices are specified in char values (Unicode code units).

Parameters:

ch - a character (Unicode code point).

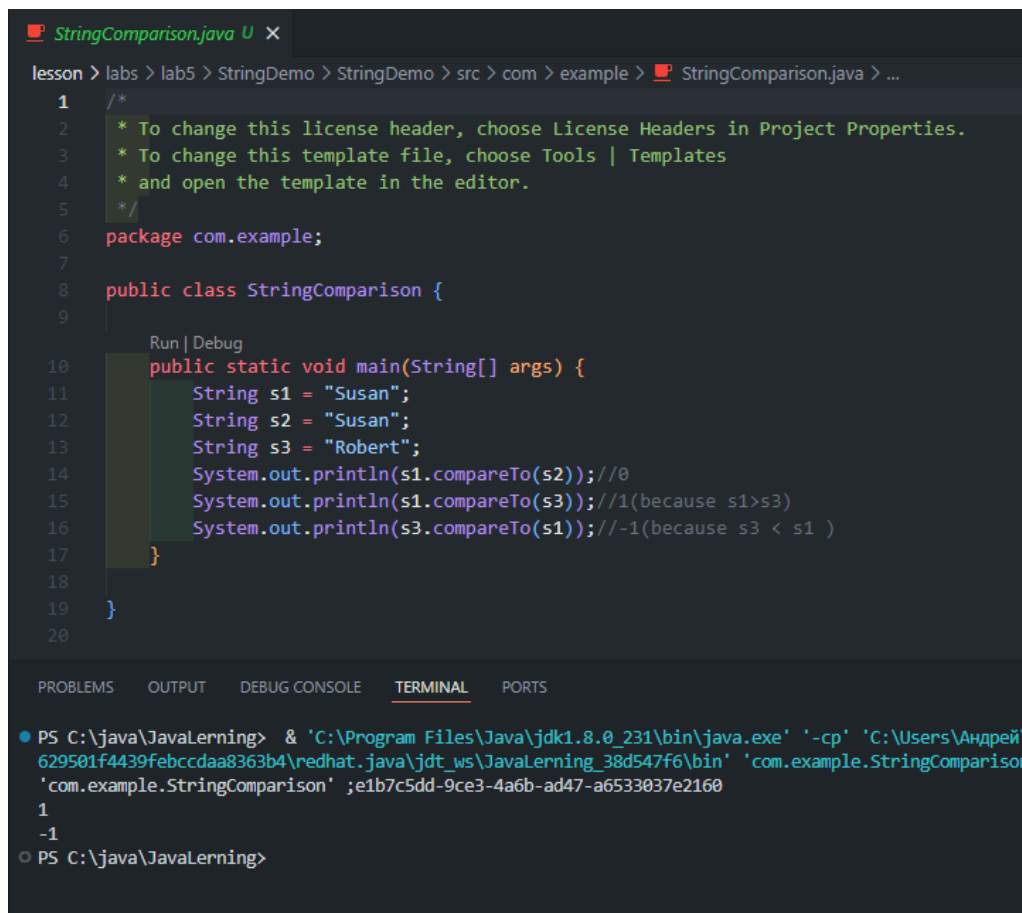
fromIndex - the index to start the search from.

Returns:

the index of the first occurrence of the character in the character sequence represented by this object that is greater than or equal to fromIndex, or -1 if the character does not occur.

2) Проект StringDemo.

Метод compareTo:



```
StringComparison.java U X
lesson > labs > lab5 > StringDemo > StringDemo > src > com > example > StringComparison.java > ...
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package com.example;
7
8  public class StringComparison {
9
10     Run | Debug
11     public static void main(String[] args) {
12         String s1 = "Susan";
13         String s2 = "Susan";
14         String s3 = "Robert";
15         System.out.println(s1.compareTo(s2)); //0
16         System.out.println(s1.compareTo(s3)); //1(because s1>s3)
17         System.out.println(s3.compareTo(s1)); //-1(because s3 < s1 )
18     }
19 }
20
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\java\JavaLerning> & 'C:\Program Files\Java\jdk1.8.0_231\bin\java.exe' '-cp' 'C:\Users\Андрей\629501f4439febccdaa8363b4\redhat.java\jdt_ws\JavaLerning_38d547f6\bin' 'com.example.StringComparison' ;e1b7c5dd-9ce3-4a6b-ad47-a6533037e2160
1
-1
○ PS C:\java\JavaLerning>
```

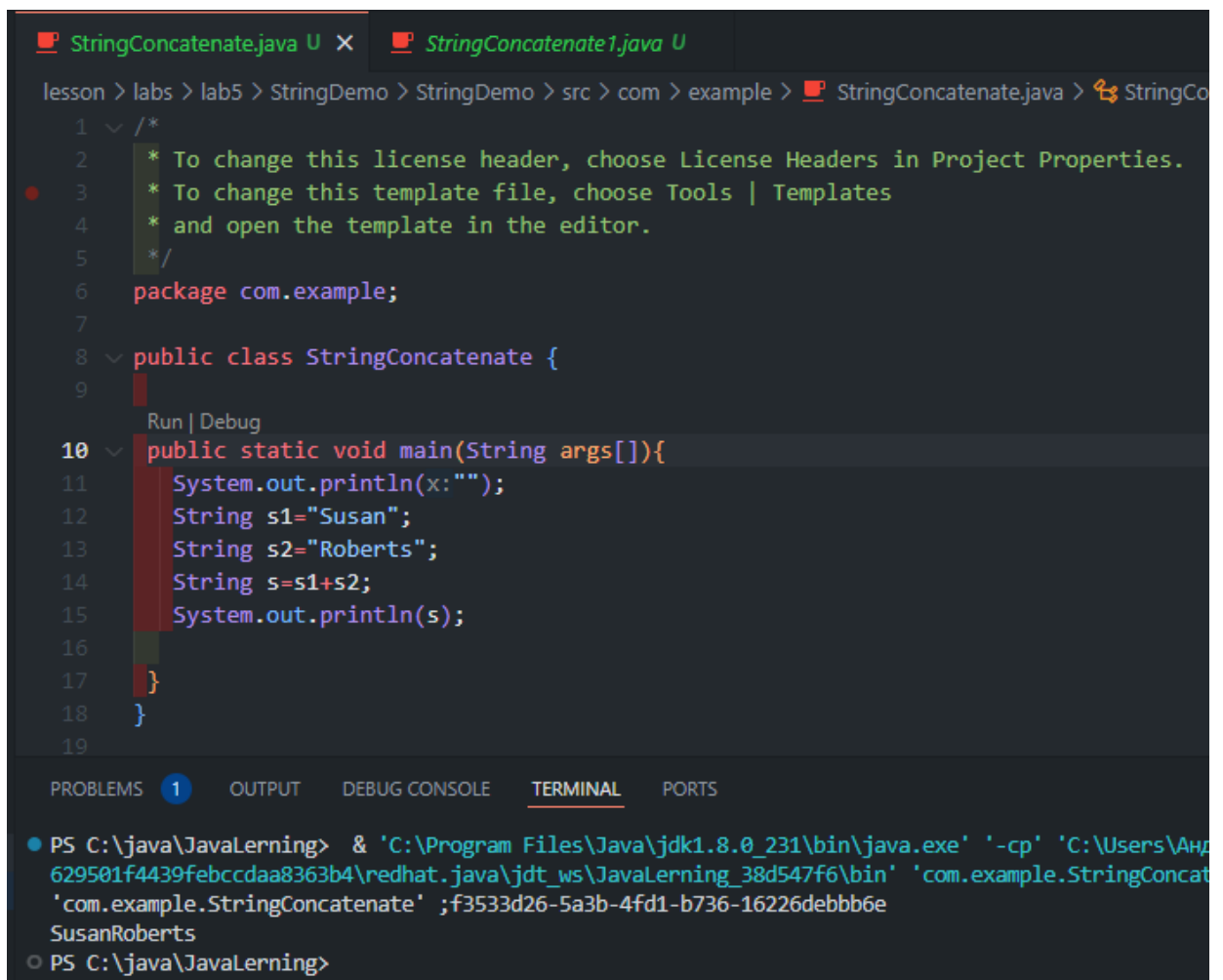
Сравнивает две строки лексикографически. Сравнение основано на значениях Unicode каждого символа в строках. Возвращается:

- «0» если строки равны;
- значение меньше нуля если строка лексикографически меньше аргумента String;
- значение больше нуля если строка лексикографически больше аргумента String;

Если строки различны, то метод возвращает разницу двух символьных значений в позиции k (наименьший индекс, в котором символы строк отличаются). Если строки одинаковы, то метод вернет разницу длин строк.

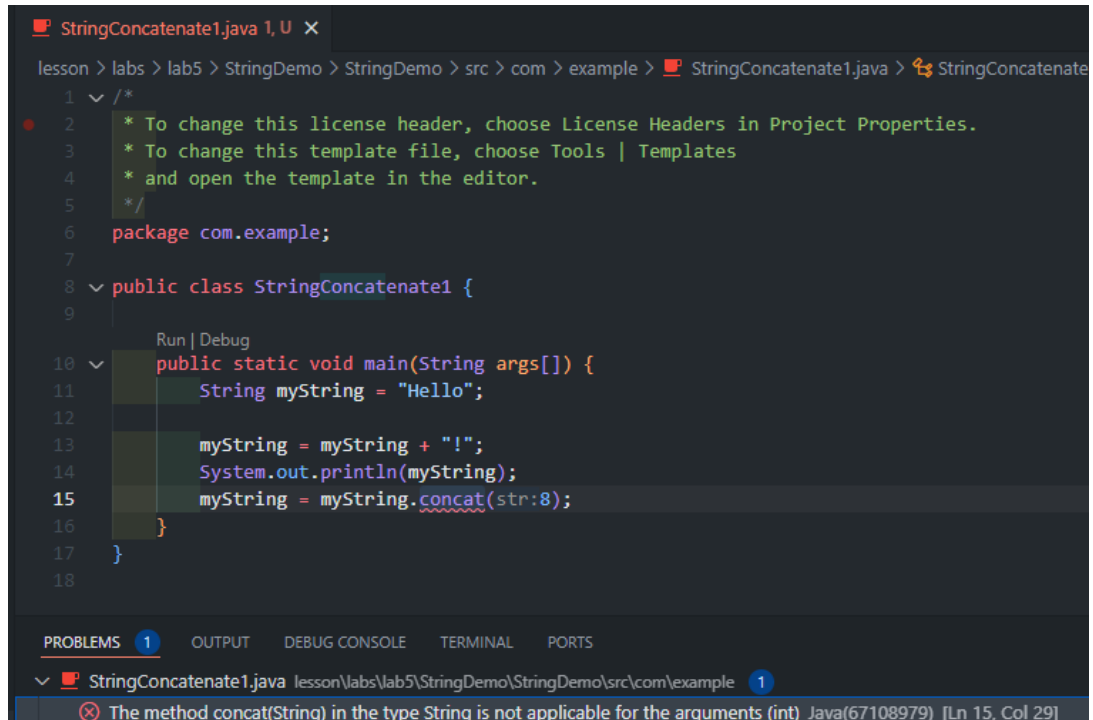
Метод concat:

Конкатенация строк — это операция склеивания двух строк с использованием знака «+». Может объединять строки и значения с типом данных char, int, double или float. Преобразует значение в формат строки перед объединением.

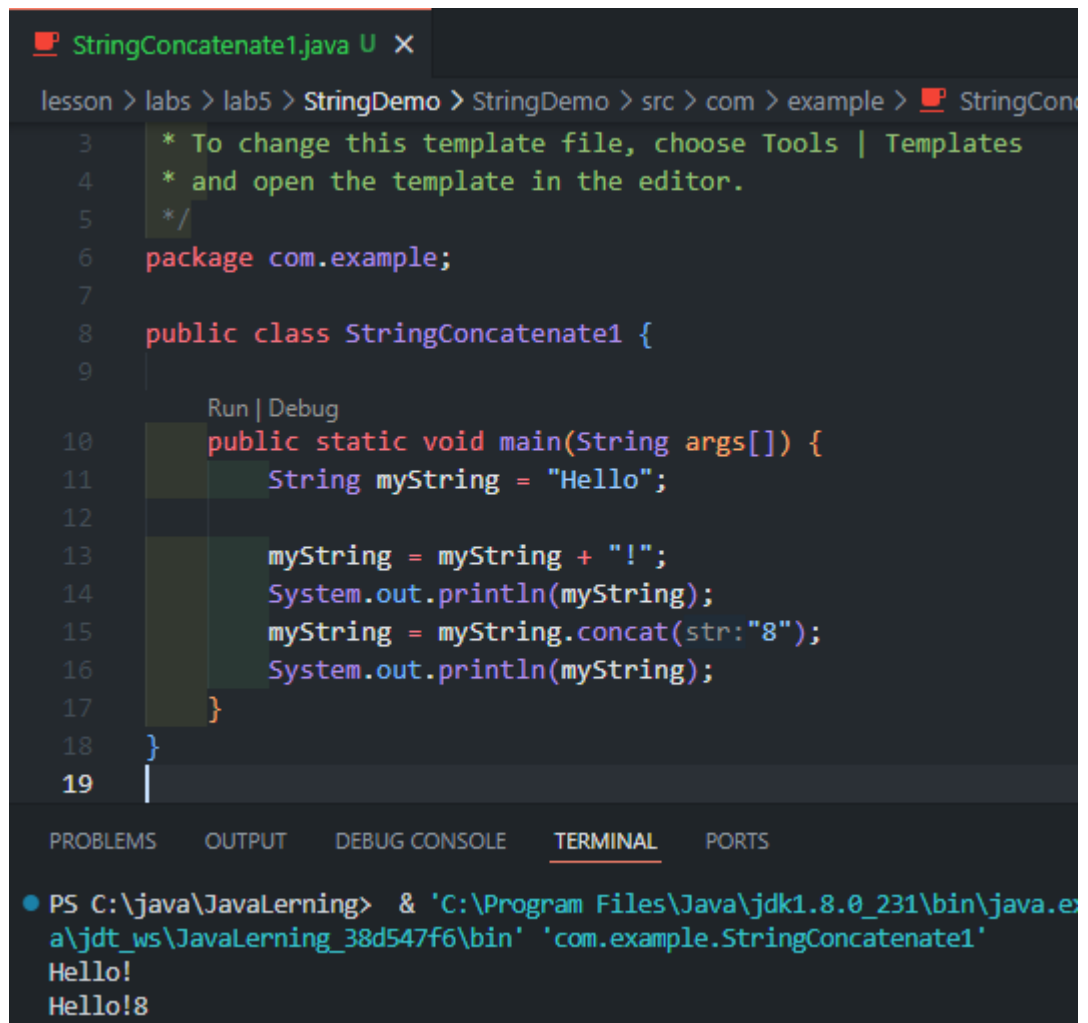


```
StringConcatenate.java U x StringConcatenate1.java U
lesson > labs > lab5 > StringDemo > StringDemo > src > com > example > StringConcatenate.java > StringCo
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package com.example;
7
8   public class StringConcatenate {
9
10      Run | Debug
11      public static void main(String args[]){
12          System.out.println(x:"");
13          String s1="Susan";
14          String s2="Roberts";
15          String s=s1+s2;
16          System.out.println(s);
17      }
18  }
19
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\java\JavaLerning> & 'C:\Program Files\Java\jdk1.8.0_231\bin\java.exe' '-cp' 'C:\Users\Андрей\629501f4439febccdaa8363b4\redhat.java\jdt_ws\JavaLerning_38d547f6\bin' 'com.example.StringConcatenate'
SusanRoberts
○ PS C:\java\JavaLerning>
```

Метод `concat` объединяет строки (вызывается только для строк: проверяет совместимость типов данных, и в случае обнаружения несоответствий в среде компиляции отображается ошибка).



```
StringConcatenate1.java 1, U ✕
lesson > labs > lab5 > StringDemo > StringDemo > src > com > example > StringConcatenate1.java > StringConcatenate1.java
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package com.example;
7
8  public class StringConcatenate1 {
9
10     public static void main(String args[]) {
11         String myString = "Hello";
12
13         myString = myString + "!";
14         System.out.println(myString);
15         myString = myString.concat(str:8);
16     }
17 }
18
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
StringConcatenate1.java lesson\labs\lab5\StringDemo\StringDemo\src\com\example 1
✕ The method concat(String) in the type String is not applicable for the arguments (int) Java(67108979) [Ln 15, Col 29]
```



The screenshot shows an IDE window with a file named `StringConcatenate1.java`. The file path is `lesson > labs > lab5 > StringDemo > StringDemo > src > com > example > StringConcatenate1.java`. The code in the file is as follows:

```
3  /* To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package com.example;
7
8  public class StringConcatenate1 {
9
10     Run | Debug
11     public static void main(String args[]) {
12         String myString = "Hello";
13
14         myString = myString + "!";
15         System.out.println(myString);
16         myString = myString.concat(str:"8");
17         System.out.println(myString);
18     }
19 }
```

Below the code editor, the **TERMINAL** tab is active, showing the command and output:

```
PS C:\java\JavaLerning> & 'C:\Program Files\Java\jdk1.8.0_231\bin\java.exe'
a\jdt_ws\JavaLerning_38d547f6\bin' 'com.example.StringConcatenate1'
Hello!
Hello!8
```

Метод `substring`:

```
6 package com.example;
7
8 public class StringExtraction {
9     Run | Debug
10     public static void main(String args[]) {
11
12         String greeting = "Hello, World!";
13         String sub = greeting.substring(beginIndex:0, endIndex:5);
14         System.out.println(sub);
15         String w = greeting.substring(beginIndex:7, endIndex:11);
16         System.out.println(w);
17         String tail = greeting.substring(beginIndex:7);
18         System.out.println(tail);
19     }
20 }
```

PROBLEMS 66 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
ning_38d547f6\bin' 'com.example.StringExtraction'
Hello
Worl
World!
```

Метод `substring` возвращает строку, которая является подстрокой данной строки. Может содержать один или два параметра типа `int`:

- один параметр — начальный индекс включительно: подстрока начинается с символа с указанным индексом и продолжается до конца данной строки;
- два параметра — начальный индекс включительно, конечный индекс не включительно: подстрока начинается с начального индекса и продолжается до конечного индекса.

Возникает ошибка `IndexOutOfBoundsException`: если индекс больше длины строки, индекс отрицательный или конечный индекс меньше начального.

Метод `indexOf`:

Метод `indexOf` возвращает индекс первого вхождения символа в строке. Может иметь один или два параметра:

- один параметр типа `int` (кодировка символов Unicode) или `char`: возвращает индекс первого вхождения символа в строке или «-1», если символа в строке нет;
- два параметра — один типа `int` (`char`) и один типа `int` (индекс, с которого следует начать поиск): возвращает индекс первого вхождения символа в

строке, начиная со второго параметра и до конца строки, возвращает «-1», если такого нет.

```
6 package com.example;
7
8 public class StringIndexOf {
9
10     Run | Debug
11     public static void main(String args[]) {
12
13         String phoneNum = "404-543-2345";
14         int idx1 = phoneNum.indexOf(ch: '-');
15         System.out.println("index of first dash: " + idx1); //3
16         int idx2 = phoneNum.indexOf(ch: '-', fromIndex: 4);
17         System.out.println("second dash idx: " + idx2); // 7
18     }
19 }
20
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
ning_38d547f6\bin' 'com.example.StringIndexOf'
index of first dash: 3
second dash idx: 7
```

Метод replace:

Метод replace содержит два параметра типа char (String): старый символ и новый символ. Метод возвращает строку, полученную из данной строки путем замены ВСЕХ вхождений первого указанного параметра на второй.

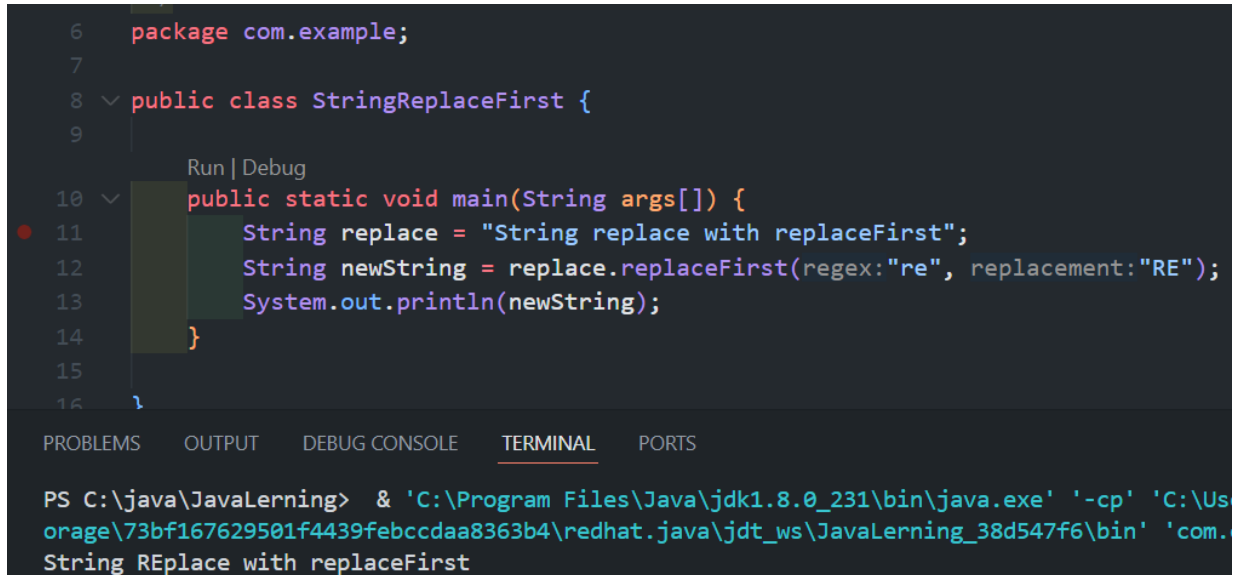
```
6 package com.example;
7
8 public class StringReplace {
9
10     Run | Debug
11     public static void main(String args[]) {
12         String replace = "Using String replace to replace character";
13         String newString = replace.replace(target:"r", replacement:"R");
14         System.out.println(newString);
15     }
16 }
17
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
orange\73bf167629501f4439febccdaa8363b4\redhat.java\jdt_ws\JavaLerning_38d547f6\bin
Using StRing Replace to Replace chaRacter
PS C:\java\JavaLerning>
```


Метод `replaceFirst`:

Метод `replaceFirst` содержит два параметра типа `String`. Заменяет подстроку, представленную первым параметром, на другую подстроку, заданную вторым параметром.



```
6 package com.example;
7
8 public class StringReplaceFirst {
9
10     Run | Debug
11     public static void main(String args[]) {
12         String replace = "String replace with replaceFirst";
13         String newString = replace.replaceFirst(regex:"re", replacement:"RE");
14         System.out.println(newString);
15     }
16 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\java\JavaLerning> & 'C:\Program Files\Java\jdk1.8.0_231\bin\java.exe' '-cp' 'C:\Us
orage\73bf167629501f4439febccdaa8363b4\redhat.java\jdt_ws\JavaLerning_38d547f6\bin' 'com.
String REplace with replaceFirst
```

Рассмотрим еще два метода:

- Метод `charAt` возвращает символ строки, находящийся в индексе, переданном в качестве параметра.
- С помощью метода `length` в классе `String` можно вычислить длину строки. Возвращает длину или количество символов в имени в виде целочисленного значения.

3) Проект StringEx.

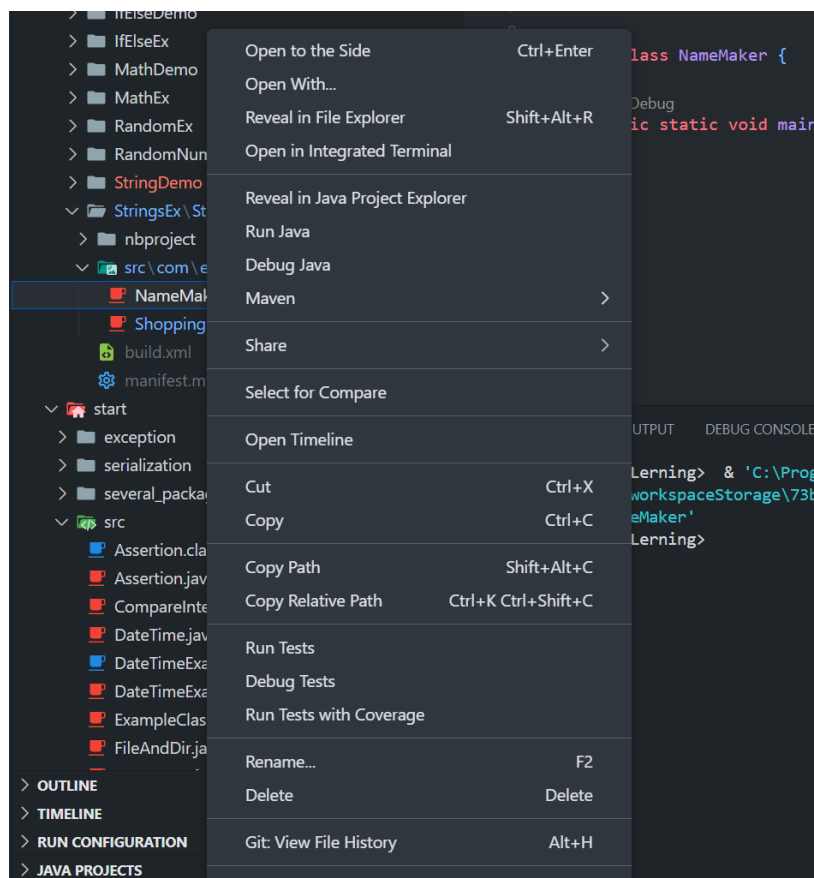
- Импортируйте и откройте проект `StringsEx`.
- Проверьте `ShoppingCart.java`. Выполните следующие действия.
 - С помощью метода `indexOf` получите индекс символа пробела (" ") в `custName`. Присвойте его значение индексу `spaceIdx`.
 - С помощью метода `substring` и `spaceIdx` получите первую часть имени `custName`. Присвойте это значение переменной `firstName` и выведите `firstName`.

```
12 public class ShoppingCart {
    Run | Debug
13 public static void main (String[] args){
14     String custName = "Steve Smith";
15     String firstName;
16     int spaceIdx = custName.indexOf(ch:' ');
17
18     firstName = custName.substring(beginIndex:0, spaceIdx);
19     System.out.println(firstName);
20 }
21
22 }
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\java\JavaLerning> c:; cd 'c:\java\JavaLerning'; & 'C:\Program Files\Ja
\Roaming\Code\User\workspaceStorage\73bf167629501f4439febccdaa8363b4\redhat.j
rt'
rt'
Steve
PS C:\java\JavaLerning> 
```

Если в проекте есть два файла .java с главными методами, то запуск через кнопку Run Java в VS Code (или сочетание клавиш Ctrl+F5).



- Изучите NameMaker.java.

- Выполните следующие действия.
 - Объявите переменные String: firstName, middleName, lastName и fullName
 - Создайте запрос, чтобы пользователи могли ввести свое имя, отчество и фамилию, и считайте эти данные с клавиатуры.
 - Задайте и выведите fullName как firstName+a blank char+middleName+a blank char+lastName.

```

8  import java.util.Scanner;
9
10 public class NameMaker {
11
12     Run | Debug
13     public static void main(String args[])
14     {
15         String firstName, middleName, lastName, fullName;
16
17         Scanner input = new Scanner(System.in);
18
19         System.out.println(x:"Enter name");
20         firstName = input.nextLine();
21         System.out.println(x:"Enter middle name");
22         middleName = input.nextLine();
23         System.out.println(x:"Enter last name");
24         lastName = input.nextLine();
25
26         fullName = firstName + ' ' + middleName + ' ' + lastName;
27         System.out.println(fullName);
28
29         input.close();
30
31     }
32 }
33

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

ning_38d547f6\bin' 'com.example.NameMaker'
ning_38d547f6\bin' 'com.example.NameMaker'
Enter name
Enter name
Andrey
Andrey
Enter middle name
Safonov
Enter last name
Igorovich
Andrey Safonov Igorovich
PS C:\java\JavaLerning>

```

2. Задания по уроку 4.4.

- 1) Программа генерирует случайное число типа double.

```

6   package com.example;
7
8   import java.util.Random;
9
10  public class FlipCoin {
11
12      Run | Debug
13      public static void main(String[] args) {
14
15          // 50% chance heads, 50% chance tails
16          Random rand = new Random();
17          double chance = rand.nextDouble();
18          System.out.println(chance);
19      }

```

Добавим условие:

— Если вероятность $< 0,5$, получается результат "орел". В противном случае получается результат "решка".

```

10  public class FlipCoin {
11
12      Run | Debug | Run main | Debug main
13      public static void main(String[] args) {
14
15          // 50% chance heads, 50% chance tails
16          Random rand = new Random();
17          double chance = rand.nextDouble();
18          System.out.println(chance);
19          if (chance < 0.5) {
20              System.out.println(x: "Орел");
21          } else {
22              System.out.println(x: "Решка");
23          }
24      }
25
26

```

PROBLEMS 69 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> & 'C:\Program Files\Java\jdk-11.0.10\bin\java.exe' -Xmx1024m -Xms1024m -Djava.class.path=C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\21aebbd47484\0.48650699514495543
Орел

```

2) Генерируется случайное целое число в диапазоне от 0 до 3 (не включительно), и в зависимости от сгенерированного числа программа выводит «камень», «бумага» или «ножницы».

```
8 import java.util.HashMap;
9 import java.util.Map;
10 import java.util.Random;
11
12 public class RockPaperScissor {
13
14     public static void main(String[] args) {
15         Map<Integer, String> choices = new HashMap<>();
16         choices.put(key:0, value:"камень");
17         choices.put(key:1, value:"бумага");
18         choices.put(key:2, value:"ножницы");
19
20         Random rand = new Random();
21         int num = rand.nextInt(bound:3);
22
23         System.out.println(choices.get(num));
24     }
25 }
26
27
```

Run | Debug | Run main | Debug main

PROBLEMS 72 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> & 'C:\Program Data\Roaming\Code\User\workspaceStorage\21aebbd47484cc76f5eb38c222ed6 бумага

3) RandomSeriesSeed.

```
8 import java.util.Random;
9
10 public class RandomSeriesSeed {
11     Run | Debug | Run main | Debug main
12     public static void main(String[] args) {
13
14         Random rand = new Random(seed:20);
15         System.out.println("Random Number 1: " + rand.nextInt(bound:100));
16         System.out.println("Random Number 2: " + rand.nextInt(bound:100));
17         System.out.println("Random Number 3: " + rand.nextInt(bound:100));
18         System.out.println(x:"Changing seed to change to sequence");
19         rand.setSeed(seed:5);
20         System.out.println("Random Number 4: " + rand.nextInt(bound:100));
21         System.out.println("Random Number 5: " + rand.nextInt(bound:100));
22         System.out.println("Random Number 6: " + rand.nextInt(bound:100));
23         System.out.println(x:"Setting seed 40 to produce the previous sequence");
24         rand.setSeed(seed:20);
25         System.out.println("Random Number 7: " + rand.nextInt(bound:100));
26         System.out.println("Random Number 8: " + rand.nextInt(bound:100));
27         System.out.println("Random Number 9: " + rand.nextInt(bound:100));
28     }
29 }
30 }
```

PROBLEMS 62 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\21aebbd47484cc76f5eb38c222ed6681\re
Random Number 1: 53
Random Number 2: 36
Random Number 3: 1
Changing seed to change to sequence
Random Number 4: 87
Random Number 5: 92
Random Number 6: 74
Setting seed 40 to produce the previous sequence
Random Number 7: 53
Random Number 8: 36
Random Number 9: 1
```

Во время создания экземпляра класса `Random` передаем постоянное целое число, чтобы задать начальное значение. Для того, чтобы изменить начальное значение, вызываем метод `setSeed()`.

При задании одинакового начального значения мы получаем одинаковые последовательности и одну отличающуюся от них последовательность (начальное значение равно 20).

Т.е. каждый раз, когда передается одно и то же начальное значение, возвращается одна и та же случайная последовательность.

3. Задания по уроку 4.5.

- 1) Класс `Java Math` содержит методы для выполнения математических расчетов. Методы класса `Math` являются статическими. Это означает,

что вам не придется создавать объект класса `Math` для вызова методов. Входит в пакет `java.lang`. Содержит следующие статистические поля:

- `PI` – значение типа `double`, которое ближе чем любое другое к π (отношение длины окружности к ее диаметру). Число «Пи» равно 3.141592653589793.

<code>static double</code>	<code>sqrt(double a)</code> Returns the correctly rounded positive square root of a double value.
----------------------------	--

PI

```
public static final double PI
```

The double value that is closer than any other to π , the ratio of the circumference of a circle to its diameter.

See Also:

[Constant Field Values](#)

2) PrintMathResult.

```
8 public class PrintMathResult {
9
10     Run | Debug | Run main | Debug main
11     public static void main(String args[]) {
12
13         Math.sqrt(a:121.0); //no output
14         System.out.println("Square root: " + Math.sqrt(a:121.0));
15         double result = Math.min(a:3, b:7) + Math.abs(-50);
16         System.out.println("Result is " + result);
17     }
18
19 }
```

PROBLEMS 63 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> & 'C:\Program Files\Ecl
'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\21aebbd47484cc76f5
Square root: 11.0
Result is 53.0
```

Методы класса `Math` не выводят результаты на консоль. Каждый метод возвращает числовой результат. С возвращаемым значением можно выполнять многие другие действия, помимо вывода.

Чтобы увидеть результат, необходимо вывести его или сохранить в переменную (первый вывод), а также результаты можно объединить и использовать в более крупном выражении (второй вывод).

3) «На бумаге».

Упр 2.

1) `Math.abs(-123)` // 1, 2 3

2) `Math.pow(3, 2)` // 9

3) `Math.sqrt(21, 0) - Math.sqrt(256, 0)`
// $11 - 16 = 5$

4) `Math.abs(Math.min(-3, -5))` // 5

Упр 3.

1) `age = Math.max(age, 0)`

2) `age = Math.min(age, 40)`

Сравним с AgeLimit:

```
9 import java.util.Scanner;
10
11 public class AgeLimit {
12     Run | Debug | Run main | Debug main
13     public static void main(String args[]) {
14         int age = 0;
15         Scanner sc = new Scanner(System.in);
16         System.out.print(s:"Enter age ");
17         age = sc.nextInt();
18         // age=Math.max(age,0);
19         age=Math.min(a:40, age);
20         System.out.print("Age is "+age);
21         sc.close();
22     }
23 }
```

PROBLEMS 63 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> c:: c
bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
ava_9fb3eade\bin' 'com.example.AgeLimit'
Enter age -1
Age is 0
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> c:: c
bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
ava_9fb3eade\bin' 'com.example.AgeLimit'
Enter age 50
Age is 40
```

Действительно, выражение `math.max(age,0)` меняет отрицательные значения возраста на 0, а `math.min(40,age)` - ограничивает максимальный возраст значением 40.

4) Площадь круга.

```
10 public class AreaOfCircle {
11
12     Run | Debug | Run main | Debug main
13     public static void main(String args[]) {
14         Scanner sc = new Scanner(System.in);
15         System.out.print(s:"Enter the radius: ");
16         double radius = sc.nextDouble();
17         double area = Math.PI *radius * radius;
18         System.out.println("The area of circle is: " + area);
19     }
20 }
```

PROBLEMS 63 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> & 'C:\Program Files\E
'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\21aebbd47484cc76f
Enter the radius: 12
The area of circle is: 452.3893421169302
```

Использование поля Math.PI для расчета площади дает более точный результат, чем использование постоянного значения π , например 3,14.

5) Индекс массы тела человека:

```
11
12 public class ComputeBMI {
13     Run | Debug | Run main | Debug main
14     public static void main(String[] args)
15     {
16         Scanner sc = new Scanner(System.in);
17         System.out.print(s:"Введите вес в фунтах:");
18         double weight = sc.nextDouble();
19         System.out.print(s:"Введите рост в дюймах:");
20         double height = sc.nextDouble();
21         sc.close();
22         double BMI = Math.round(weight / (height * height) * 703 * 10.0) / 10.0;
23         System.out.println("Индекс массы вашего тела равен " + BMI);
24     }
25 }
```

PROBLEMS 72 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> c:.; cd 'c:\Users\kseilons\YandexDisk\university\3 курс\Java\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\kseilons\AppData\Roaming\Code\bin\java_9fb3eade\bin' 'com.example.ComputeBMI'
Введите вес в фунтах:132,5
Введите рост в дюймах:62,5
Индекс массы вашего тела равен 23.8
```

4. Упражнение 4.

1) Проблема 1.

- Создайте java-файл ComputeMethods.java и определите следующие три метода: public double fToC(double degreesF) public double hypotenuse(int a, int b) public int roll()
- Создайте второй java-файл TestClass.java и выполните следующее: добавьте метод main в метод main: создайте экземпляр ComputeMethods, вызовите методы, определенные в файле ComputeMethods.java этого экземпляра, и отобразите их результаты.

```

1  package labs.lab5.practic4;
2
3  import java.util.Random;
4
5  public class ComputeMethods {
6      public double fToC (double degreesF) {
7          double C = (5 * (degreesF - 32)) / 9;
8          return C;
9      }
10     public double hypotenuse(int a, int b) {
11         double c = Math.sqrt(a * a + b * b);
12         return c;
13     }
14     public int roll() {
15         Random rand = new Random();
16         int firstThrow = rand.nextInt(bound:6) + 1;
17         int secondThrow = rand.nextInt(bound:6) + 1;
18         int sum = firstThrow + secondThrow;
19         return sum;
20     }
21 }

```

```

1  package labs.lab5.practic4;
2
3  public class TestClass {
4      public static void main(String[] args) {
5          ComputeMethods compMeth = new ComputeMethods();
6          double C = compMeth.fToC(degreesF:233);
7          double c = compMeth.hypotenuse(a:4, b:8);
8          int sum = compMeth.roll();
9
10         System.out.println("Температура в градусах Цельсия: " + C);
11         System.out.println("Гипотенуза треугольника равна : " + c);
12         System.out.println("Сумма значений на гранях равна : " + sum);
13     }
14 }
15

```

PROBLEMS 86 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> & 'C:\Program Files\Eclipse Adoptium\
'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\21aebbd47484cc76f5eb38c222ed668
Температура в градусах Цельсия: 111.66666666666667
Гипотенуза треугольника равна : 8.94427190999916
Сумма значений на гранях равна: 6

```

Методы fToC (преобразует температуру, заданную в градусах по Фаренгейту, в градусы по Цельсию), hypotenuse (вычисляет длину гипотенузы треугольника на основе длин его катетов) и roll (моделирует бросок двух кубиков и отображает сумму значений на их верхних гранях.).

2) Проблема 2.

```
3 import java.util.Scanner;
4
5 public class ProcessName {
6     Run | Debug | Run main | Debug main
7     public static void main(String[] args) {
8         try (Scanner sc = new Scanner(System.in)) {
9             System.out.print(s:"Введите ваше имя: ");
10            String fullName = sc.nextLine();
11            int indSpace = fullName .indexOf(str:" ");
12            System.out.println(indSpace);
13            String initial = fullName .substring(beginIndex:0 , endIndex:1);
14            String lastName = fullName .substring(indSpace + 1);
15            System.out.println("Ваше имя: " + lastName + ", " + initial + '.');
16        }
17    }
18 }
19
```

PROBLEMS 78 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> ^C
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> c:: cd 'c:\Users\kseilons\YandexDisk\university\3 курс\Java' & java -XX:+ShowCodeDetailsInExceptionMessages -cp 'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\21aebbd4\workspace\labs.lab5.practic4.ProcessName'
Введите ваше имя: Andrey Safonov
6
Ваше имя: Safonov, A.
```

5. IfElseDemo.

1) Тип данных boolean объявляется с помощью ключевого слова boolean и может иметь только два возможных значения: true и false.

```
6 package com.example;
7
8
9 public class BooleanDemo {
10     Run | Debug | Run main | Debug main
11     public static void main(String args[]) {
12
13         boolean passed, largeVenue, grade;
14
15         passed = true; // Assigning Value
16         largeVenue = false; // Assigning Value
17         grade = passed; // Assigning Variable
18
19         System.out.println(passed); // Printing Value
20         System.out.println(largeVenue); // Printing Value
21         System.out.println(grade); // Printing Value
22
23     }
24
25 }
```

PROBLEMS 72 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> & 'C:\Program Files\Java\jdk-11.0.2\bin\java.exe' -XX:+ShowCodeDetailsInExceptionMessages -cp 'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\21aebbd4\workspace\labs.lab5.practic4.BooleanDemo'
true
false
true
```

2) == является оператором отношения. Этот оператор проверяет, являются ли обе части выражения boolean равными. Переменной boolean можно присвоить любое значение, которое возвращает выражение boolean. В приведенном ниже примере переменной res1 присваивается значение, возвращающее true (value1 = 15 и оно сравнивается со значением 15), а переменной res 2 – значение, возвращающее false (сравниваются value1 и value2, а они не равны).

```
6 package com.example;
7
8 /**
9  *
10  * @author anshenoy
11  */
12 public class EqualityTest {
13
14     Run | Debug | Run main | Debug main
15     public static void main(String args[]) {
16         int value1 = 15;
17         int value2 = 24;
18         boolean res1 = value1 == 15;
19         System.out.println("res1:" + res1);
20         boolean res2 = value1 == value2;
21         System.out.println("res2: " + res2);
22     }
23 }
24
25
```

PROBLEMS 73 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> 8
'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStor
res1:true
res2: false
```

3) Работа с операторами отношения:

```
5 package com.example;
6
7
8 public class RelationalDemo {
9
10     Run | Debug | Run main | Debug main
11     public static void main(String args[]) {
12
13         int a = 10;
14         int b = 20;
15
16         System.out.println("a == b = " + (a == b)); //checks for equality
17         System.out.println("a != b = " + (a != b));
18         System.out.println("a > b = " + (a > b));
19         System.out.println("a < b = " + (a < b));
20         System.out.println("b >= a = " + (b >= a));
21         System.out.println("b <= a = " + (b <= a));
22     }
23 }
```

PROBLEMS 72 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> & 'C:\Program Files\Eclipse Ado
'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\21aebbd47484cc76f5eb38c222
● a == b = false
a != b = true
a > b = false
a < b = true
b >= a = true
b <= a = false
```

4) Оператор if состоит из выражения boolean, после которого идет один или несколько операторов. Если выражение boolean имеет значение true: Оператор выполнен. Если выражение boolean имеет значение false: Оператор пропущен. В примере выражение boolean во втором операторе if возвращает выражение true. Поэтому в консоли выводится "Drive straight".

```
6 package com.example;
7
8
9 public class BooleanIfDriving {
10     Run | Debug | Run main | Debug main
11     public static void main(String args[]) {
12         String left = "museum";
13         String straight = "gym";
14         String right = "restaurant";
15
16         if (left.equals(anObject:"gym")) {
17             System.out.print(s:"Turn Left");
18         }
19         if (straight.equals(anObject:"gym")) {
20             System.out.print(s:"Drive straight");
21         }
22         if (right.equals(anObject:"gym")) {
23             System.out.print(s:"Turn Right");
24         }
25     }
26 }
```

PROBLEMS 78 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\kse
ava_9fb3eade\bin' 'com.example.BooleanIfDriving'
Drive straight
```

5) В примере проверяется возраст человека, и если он ниже 18, то выводит предупреждение о несовершеннолетнем за рулем (можно делать через if или просто присвоением значения переменной типа boolean):

```
1 package com.example;
2
3 public class IfDemo {
4     Run | Debug | Run main | Debug main
5     public static void main(String args[]) {
6
7         boolean drivingUnderAge;
8
9         int age = 16;
10
11         // if (age <= 18) {
12         //     drivingUnderAge = true;
13         //     System.out.println("Was Driving Under Age: " + drivingUnderAge);
14         // }
15
16         drivingUnderAge= (age <=18);
17         System.out.println("Was Driving Under Age: " + drivingUnderAge);
18     }
19 }
```

PROBLEMS 78 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> & 'C:\Program Files\Eclipse Adoptium\jdk-21.0.4.7-hotspot\bin\java.exe' -cp 'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\21aebbd47484cc76f5eb38c222ed6681\redhat.java\jd
Was Driving Under Age: true
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java>
```

6) В данной программе введенное пользователем число сравнивается с «7»: если оно равно «7», то это удачное число, иначе – неудачное.


```
7
8 import java.util.Scanner;
9
10 public class IfElseDemo1 {
11
12     Run | Debug | Run main | Debug main
13     public static void main(String[] args) {
14         Scanner in = new Scanner(System.in);
15         int value = 0;
16         System.out.println(x:"Enter a number:");
17         value = in.nextInt();
18         if (value == 7) {
19             System.out.println(x:"That's lucky!");
20         } else {
21             System.out.println(x:"That's unlucky!");
22         }
23     }
24 }
25 }
```

PROBLEMS 79 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> & 'C:\Program Files\Java\jdk-11.0.10\bin\java.exe' -cp 'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\21aebbd\src\main\java' IfElseDemo1
Enter a number:
12
That's unlucky!
```

7) В примере представлены два оператора if: первый выполняет проверку значений, которые больше 88, второй выполняет проверку значений, которые меньше или равны 88. Оба оператора if оценены, даже если первый из них имеет значение true.

```
6 package com.example;
7
8 public class IfDemo1 {
9
10
11 Run | Debug | Run main | Debug main
12 public static void main(String args[]) {
13     int grade = 85;
14     if (grade > 88) {
15         System.out.println(x:"You made the Honor Roll.");
16     }
17     if (grade <=88) {
18         System.out.println(x:"You are eligible for tutoring.");
19     }
20
21 }
22 }
```

PROBLEMS 79 OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> & 'C:\Program Files\Eclipse
'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\21aebbd47484cc76f5eb3
You are eligible for tutoring.

8) Можно заменить два оператора if на оператор if/else. Оператор if/else более эффективен, так как выполняется только одно сравнение.

```
6 package com.example;
7
8 /**
9  *
10  * @author anshenoy
11  */
12 public class IfElseDemo {
13
14 Run | Debug | Run main | Debug main
15 public static void main(String args[]) {
16     int grade = 85;
17     if (grade > 88) {
18         System.out.println(x:"You made the Honor Roll.");
19     }
20     else {
21         System.out.println(x:"You passed.");
22     }
23 }
24 }
25 }
26 }
```

PROBLEMS 80 OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> & 'C:\Program Files
'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\21aebbd47484cc
You passed.

9) Сравнивается введенная пользователем строка со строкой «Elvis»: если равны, то выводит «You are the king of rock and roll», иначе «You are not the king». Сравниваем переменные String с помощью метода equals(). Он

принимает аргумент String, проверяет равенство содержимого String, затем возвращает выражение boolean.

```
6  package com.example;
7
8  /**
9   *
10  * @author anshenoy
11  */
12
13  import java.util.Scanner;
14  public class IfElseDemo2 {
15      Run | Debug | Run main | Debug main
16      public static void main(String[] args){
17          Scanner in = new Scanner(System.in);
18          String name = "";
19          System.out.println(x:"Enter your name:");
20          name = in.next();
21          if( name.equals(anObject:"Elvis"))
22              System.out.println(x:"You are the king of rock and roll");
23          else
24              System.out.println(x:"You are not the king");
25      }
26  }
```

PROBLEMS 81 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> & 'C:\Program Files\Eclip
'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\21aebbd47484cc76f5el
Enter your name:
Elvis
You are the king of rock and roll
```

10) Сравниваем переменные String с помощью метода equals().

```
6 package com.example;
7
8 /**
9  *
10  * @author anshenoy
11  */
12 public class EqualityString {
13
14     Run | Debug | Run main | Debug main
15     public static void main(String args[]) {
16         String name1 = "Fred Smith";
17         String name2 = "Sam Smith";
18         if (name1.equals(name2)) {
19             System.out.println(x:"Same name.");
20         } else {
21             System.out.println(x:"Different name.");
22         }
23     }
24 }
```

PROBLEMS 82 OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> & 'C:\Program Files\Java\jdk-11.0.10\bin\java.exe' -cp 'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\21aebbf1-4b4d-412c-8c6c-4f4b4b4b4b4b' Different name.

11) Сравниваем переменные String с помощью метода `equalsIgnoreCase()`. Он выполняет лексикографическое сравнение строк, при котором различия регистра игнорируются.

```
6 package com.example;
7
8 public class EqualityString1 {
9
10     Run | Debug | Run main | Debug main
11     public static void main(String args[]) {
12         String name1 = "Fred Smith";
13         String name2 = "fred smith";
14         if (name1.equalsIgnoreCase(name2)) {
15             System.out.println(x:"Same name.");
16         } else {
17             System.out.println(x:"Different name.");
18         }
19     }
20 }
```

PROBLEMS 82 OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> & 'C:\Program Files\Java\jdk-11.0.10\bin\java.exe' -cp 'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\21aebbf1-4b4d-412c-8c6c-4f4b4b4b4b4b' Same name.

12) Почти никогда не следует выполнять сравнение переменных String с помощью ==. Оператор == в Java сравнивает ссылки на объекты, а не сами объекты. В случае строк это означает, что == проверяет, указывают ли две переменные на один и тот же объект в памяти, а не на то, содержат ли они одинаковую последовательность символов.

```
5  */
6  package com.example;
7
8  public class EqualityString2 {
9
10     Run | Debug | Run main | Debug main
11     public static void main(String args[]) {
12         String name1 = "Fred Smith";
13         String name2 = "Fred Smith";
14
15         if (name1 == name2) {
16             System.out.println(x:"Same name.");
17         } else {
18             System.out.println(x:"Different name.");
19         }
20     }
21 }
22
```

PROBLEMS 83 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> & 'C:\Program Files\Java\jdk-11.0.10\bin\java.exe' -cp 'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\21aef...' EqualityString2
Same name.
```

6. IfElseEx.

1) Упражнение 1.

Ввод пользователями своего возраста. Объявление переменной boolean, drivingUnderAge. Инициализация drivingUnderAge до значения false. Создание выражения boolean для проверки, является ли введенный пользователем возраст меньше или равен 18, затем присвоение переменной drivingUnderAge значение true. Вывод значения drivingUnderAge.

```
8  import java.util.Scanner;
9
10 public class AgeValidity {
11
12     Run | Debug
13     public static void main(String[] args) {
14         Scanner sc = new Scanner(System.in);
15         System.out.println(x:"Enter your age");
16         int age = sc.nextInt();
17         sc.close();
18         boolean drvingUnderAge = age > 18;
19         System.out.println(drvingUnderAge);
20     }
21 }
```

PROBLEMS 69 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\kseilons\YandexDisk\university\3 кypc\Java> c::;
bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
ava_9fb3eade\bin' 'labs.lab5.IfElseEx.IfElseEx.src.com.examp]
Enter your age
18
false
PS C:\Users\kseilons\YandexDisk\university\3 кypc\Java> c::;
bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
● ava_9fb3eade\bin' 'labs.lab5.IfElseEx.IfElseEx.src.com.examp]
Enter your age
20
true
```

2) Упражнение 2.

Проверка числа на четность.

```
8
9 import java.util.Scanner;
10
11 public class ChkOddEven {
12
13     Run | Debug | Run main | Debug main
14     public static void main(String args[]) {
15
16         Scanner in = new Scanner(System.in);
17         int num = 0;
18         System.out.println(x:"Enter a number:");
19         num = in.nextInt();
20         in.close();
21
22         String message = "Число " + num + " является ";
23         message += num % 2 == 0 ? "четным." : "нечетным.";
24         System.out.println(message);
25     }
26 }
```

PROBLEMS 76 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\kseilons\AppData\Roaming\Cod
ava_9fb3eade\bin' 'labs.lab5.IfElseEx.IfElseEx.src.com.example.ChkOddEven'
Enter a number:
223
Число 223 является нечетным.
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java> c:: cd 'c:\Users\kseilons\YandexDisk\univer
bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\kseilons\AppData\Roaming\Cod
ava_9fb3eade\bin' 'labs.lab5.IfElseEx.IfElseEx.src.com.example.ChkOddEven'
Enter a number:
222
Число 222 является четным.
```

3) Упражнение 3.

Первое решение проблемы:

```
public class AgeCheck {  ⚙️ Andrey Safonov *  
  
    public static void main(String[] args) {  ⚙️ Andrey Safonov *  
        int myAge = 19;    // I am 19; let me see if I can drive yet  
  
        if (myAge >= 16) {  
            System.out.println("I'm old enough to have a driver's license!");  
        }  
  
        if (myAge < 16) {  
            System.out.println("I'm not old enough yet... :*(");  
        }  
    }  
}
```

ChkOddEven × AgeCheck ×

⚙️ ⋮

"C:\Program Files\Eclipse Adoptium\jdk-21.0.4.7-hotspot\bin\java.exe" "-javaagent:C:\Program Fi
I'm old enough to have a driver's license!

Второе:


```
public class AgeCheck {  Andrei Safonov *

    public static void main(String[] args) {  Andrei Safonov *
        int myAge = 19;    // I am 19; let me see if I can drive yet

        if (myAge >= 16) {
            System.out.println("I'm old enough to have a driver's license!");
        }
        else {
            System.out.println("I'm not old enough yet... :*(");
        }
    }
}
```

ChkOddEven x AgeCheck x

"C:\Program Files\Eclipse Adoptium\jdk-21.0.4.7-hotspot\bin\java.exe" "-javaagent:
I'm old enough to have a driver's license!

4) Упражнение 4.

Объявление и инициализация переменной boolean outOfStock. Если количество > 1, меняем переменную сообщения, чтобы указать множественное число. Если позиция отсутствует, сообщаем пользователю о ее недоступности. Или выводим сообщение и общую стоимость.

```
public class ShoppingCart {  🧑 Andrey Safonov *
    public static void main(String[] args) {  🧑 Andrey Safonov *
        double price = 21.99;
        int quantity = 2;
        double tax = 1.04;
        double total;
        String message = custName+" wants to purchase "+quantity+" "+itemDesc;

        // Calculating total cost
        total = (price*quantity)*tax;

        // Declare outOfStock variable and initialize it
        boolean outOfStock = false;

        // Test quantity and modify message if quantity > 1.
        if (quantity > 1) {
            itemDesc = "Shirts";
        }

        // Test outOfStock and notify user in either case.
        if (outOfStock) {
            System.out.println("The product is out of stock");
        }
        else {
            System.out.println(message);
            System.out.println("Total price is " + total);
        }
    }
}
```

ShoppingCart x

"C:\Program Files\Eclipse Adoptium\jdk-21.0.4.7-hotspot\bin\java.exe" -javaagent:
Mary Smith wants to purchase 2 Shirt
Total price is 45.7392

5) Упражнение 5.

- Объявите имя переменной String.
- Пользователь должен ввести значение для имени. —
- Убедитесь, что указано имя "Мое", затем введите текст "Вы король рок-н-ролла".
- Или введите текст "Вы не король".

- Не используйте ==

```
import java.util.Scanner;
public class StringEquality {  🧑 Andrey Safonov *
    public static void main(String[] args) {  new *
        System.out.print("Enter your name: ");
        Scanner sc = new Scanner(System.in);
        String name = sc.next();
        sc.close();

        if (name.equals("My")) {
            System.out.println("You're the kind of rock and roll");
        }
        else {
            System.out.println("You're not the kind");
        }
    }
}
```

StringEquality x

"C:\Program Files\Eclipse Adoptium\jdk-21.0.4.7-hotspot\bin\java.exe"
Enter your name: *My*
You're the kind of rock and roll