

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.

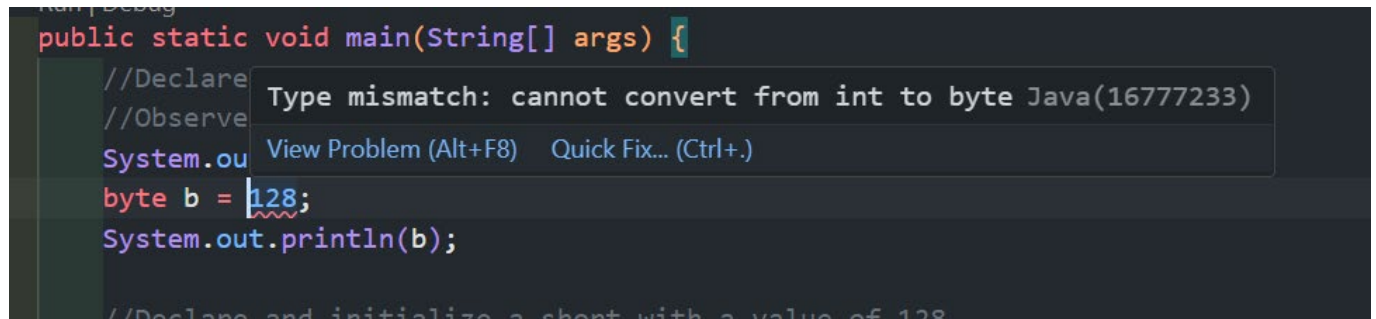
Санкт-Петербург, 2024

Задание 1 (по уроку 3.4)

1) Casting01.

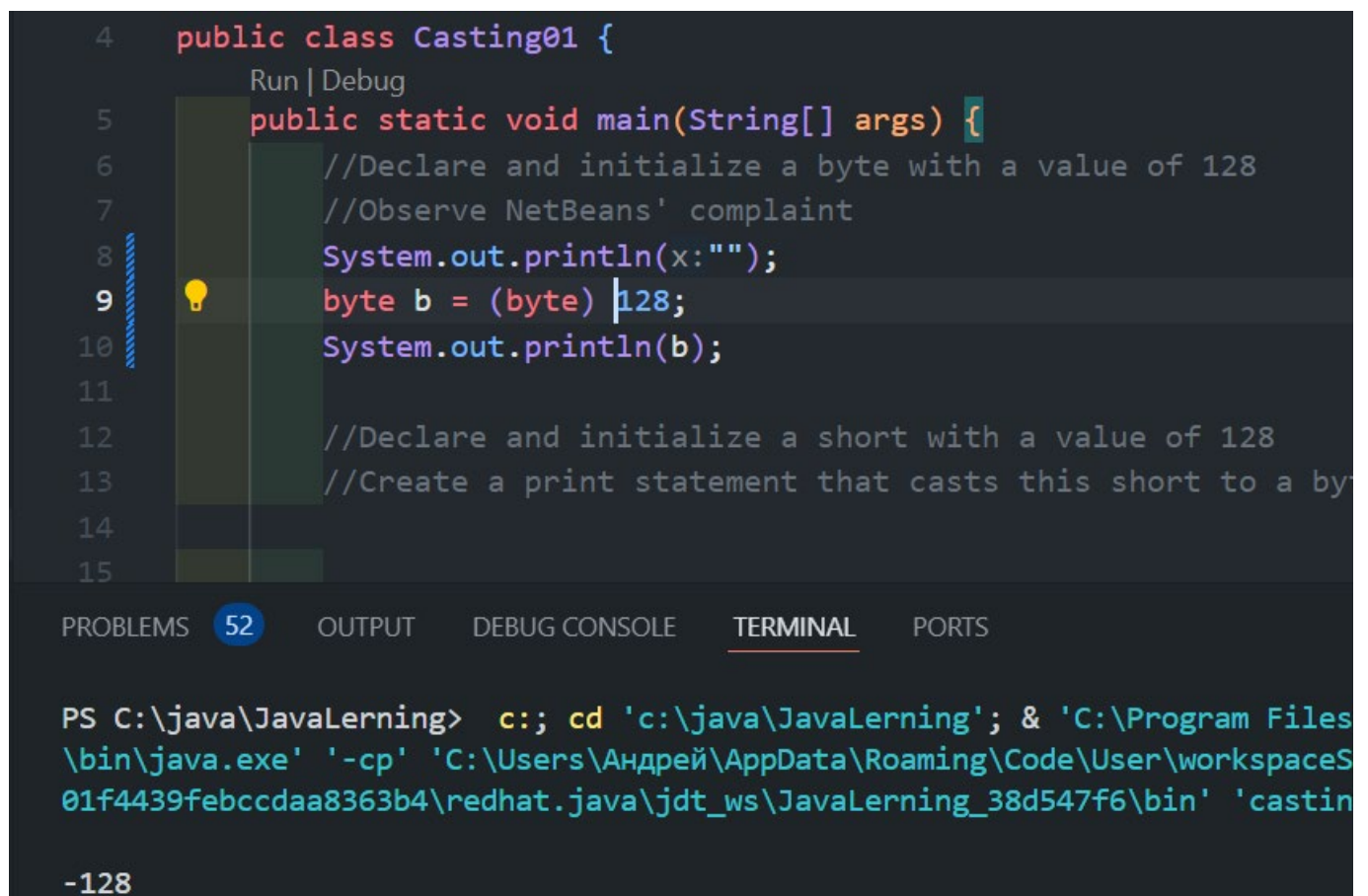
Объявляем переменную `b` с типом `byte` и задаем ей значение 128. Так как тип `byte` может хранить числа от -128 до 127 компилятор выдает ошибку:

```
public static void main(String[] args) {  
    //Declare  
    //Observe  
    System.out.println(b);  
    //Declare and initialize a short with a value of 128
```



Если мы попробуем явно преобразовать число 128 к `byte`, программа заработает, но тогда в переменной будет храниться значение -128.

```
4 public class Casting01 {  
5     Run | Debug  
6     public static void main(String[] args) {  
7         //Declare and initialize a byte with a value of 128  
8         //Observe NetBeans' complaint  
9         System.out.println(x:"");  
10        byte b = (byte) 128;  
11        System.out.println(b);  
12        //Declare and initialize a short with a value of 128  
13        //Create a print statement that casts this short to a by  
14  
15
```



Объявите и инициализируйте переменную `short` со значением 128: Создайте оператор вывода, преобразующий `short` в `byte`.

```
4 public class Casting01 {
    Run | Debug
5 public static void main(String[] args) {
6 //Declare and initialize a byte with a value of 127
7 //Observe NetBeans' complaint
8 System.out.println(x:"");
9 byte b = (byte) 128;
10 System.out.println(b);
11
12 short s = 128;
13 System.out.println((byte) s);
14
15 //Declare and initialize a short with a value of 127
16 //Create a print statement that casts this short to a byte
}
```

PROBLEMS 52 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\java\JavaLerning> c::; cd 'c:\java\JavaLerning'; & 'C:\Program Files\Java\jdk-9.0.4\bin\java.exe' '-cp' 'C:\Users\Андрей\AppData\Roaming\Code\User\workspace\01f4439febccdaa8363b4\redhat.java\jdt_ws\JavaLerning_38d547f6\bin'
-128
PS C:\java\JavaLerning> ^C
PS C:\java\JavaLerning>
PS C:\java\JavaLerning> c::; cd 'c:\java\JavaLerning'; & 'C:\Program Files\Java\jdk-9.0.4\bin\java.exe' '-cp' 'C:\Users\Андрей\AppData\Roaming\Code\User\workspace\01f4439febccdaa8363b4\redhat.java\jdt_ws\JavaLerning_38d547f6\bin'
5cjdt_ws\x5cJavaLerning_38d547f6\x5cbin' 'casting01.Casting01' ;636
-128
-128
PS C:\java\JavaLerning>
```

Мы можем наблюдать, что программа запускается, но корректно вывести число 128 не получается, происходит переполнение.

Объявите и инициализируйте переменную byte со значением 127. Прибавьте 1 и выведите результат. Можно увидеть что произошло переполнение, и переменная стала равна -128. Ещё раз прибавьте 1 и выведите. Это привело к тому что переменная стала равна -127.

```
2  package casting01;
3
4  public class Casting01 {
5      Run | Debug
6      public static void main(String[] args) {
7          System.out.println(x:"");
8          byte b = (byte) 128;
9          System.out.println(b);
10
11         short s = 128;
12         System.out.println((byte) s);
13
14         byte a = 127;
15         System.out.println(++a);
16         System.out.println(++a);
17     }
18 }
```

PROBLEMS 52 OUTPUT DEBUG CONSOLE TERMINAL PORTS

01f4439febccdaa8363b4\redhat.java\jdt_ws\JavaLerning_38d547f6\bin

-128
-128
-128
-127

2) Casting02.

Импортируйте и откройте проект Casting02. В этой программе есть несколько ошибок. Эти ошибки можно устранить, применив следующие знания

- Знание типов данных
- Знание принципа повышения
- Знание принципа преобразования

Исправленная версия программы

```
3 public class Casting02 {
    Run | Debug
4     public static void main(String[] args) {
5
6         //World population today
7         long currentWorldPop = 7_000_000_000L; // Добавили литерал L
8         System.out.println("Current World Population: " +currentWorldPop);
9
10        //World population in 1950, Source: United Nations Population Division
11        //2.52 Billion
12        int africaPop = 221_000_000;
13        int asiaPop = 1_402_000_000;
14        int europePop = 547_000_000;
15        int americasPop = 339_000_000;
16        int oceanaPop = 13_000_000;
17        // Явно привели одну из переменных к типу long
18        // и добавили скобки, чтобы переменные сначала сложились
19        // а потом конвертировались в строку
20        System.out.println("World Population in 1950: "
21            + ((long)(africaPop) +asiaPop +europePop +americasPop +oceanaPop));
22
23        //The current population of Asia is an estimated 60% of the world population
24        double percentAsia = 0.6;
25        // percentAsia - вещественное число, поэтому нужно явно
26        // преобразовать к long
27        long currentAsiaPop = (long) (currentWorldPop * percentAsia);
28        System.out.println("Current Asia Population: " +currentAsiaPop);
29    }
30 }
31
```

PROBLEMS 70 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
at.java\jdt_ws\JavaLerning_38d547f6\bin' 'casting02.Casting02'
Current World Population: 7000000000
World Population in 1950: 2522000000
Current Asia Population: 4200000000
PS C:\java\JavaLerning>
```

3) Parsing01.

- Импортируйте и откройте проект Parsing01.
- Объявите и инициализируйте 3 объекта String со предоставленными данными.
- Выполните синтаксический анализ и умножьте `shirtPrice*taxRate`, чтобы получить значение налога.
 - Выведите это значение.
- Попробуйте проанализировать `taxRate` как `int`.
 - Прочтите сообщение об ошибке.
- Попробуйте проанализировать `gibberish` как `int`.
 - Прочтите сообщение об ошибке.

```
package parsing01;

public class Parsing01 {
    public static void main(String[] args) {
        String shirtPrice = "15", taxRate = "0.05", gibberish = "dfasdvkmcxv;lkzmd;jkli";

        int shirtPriceInt = Integer.parseInt(shirtPrice);
        double taxRateDouble = Double.parseDouble(taxRate);
        double shirtTax = shirtPriceInt * taxRateDouble;
        System.out.println(shirtTax);

        int taxRateInt = Integer.parseInt(taxRate);
        int gibberishInt = Integer.parseInt(gibberish);
    }
}
```

Parsing01 x

```
"C:\Program Files\Eclipse Adoptium\jdk-21.0.4-hotspot\bin\java.exe" "-javaagent:C:\Program Files\Je
Exception in thread "main" java.lang.NumberFormatException: Create breakpoint : For input string: "0.05"
    at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
    at java.base/java.lang.Integer.parseInt(Integer.java:662)
    at java.base/java.lang.Integer.parseInt(Integer.java:778)
    at parsing01.Parsing01.main(Parsing01.java:12)
0.75
```

При попытке проанализировать `taxRate` как `int` и `gibberish` как `int` возникает ошибка «`NumberFormatException`» (Исключение числового формата) потому, что эти строки не имеют правильного формата для этих преобразований.

1. Задания по уроку 3.5.

1) Input01.

- Импортируйте и откройте проект `Input01`.
- Создайте `JOptionPane`:
 - В `NetBeans` отобразится предупреждение.
 - Следуя рекомендации `NetBeans`, импортируйте `javax.swing.JOptionPane`
- Сохраните введенные данные как объект `String`.
- Выведите переменную `String`.
- Выполните синтаксический анализ `String` как отдельной переменной `int`.
 - Потребуется ввести значение, доступное для синтаксического анализа.
 - Выведите значение `+1`.

- Попробуйте создать диалоговое окно, проанализировать его и инициализировать `int` в одной строке. Используйте только одну точку с запятой (;).

```
package input01;

import javax.swing.JOptionPane;

public class Input01 {
    public static void main(String[] args) {
        String input = JOptionPane.showInputDialog("Введите текст: ");
        System.out.println(input);

        int x = Integer.parseInt(input);
        System.out.println(x + 1);

        x = Integer.parseInt(JOptionPane.showInputDialog("Введите текст: "));
        System.out.println(x);
    }
}
```

Input01 x

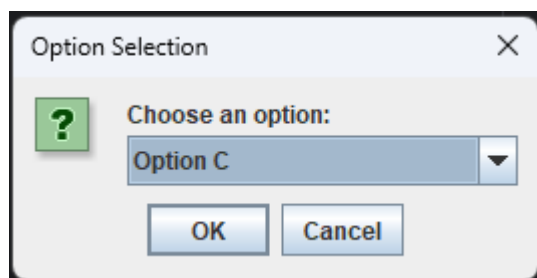
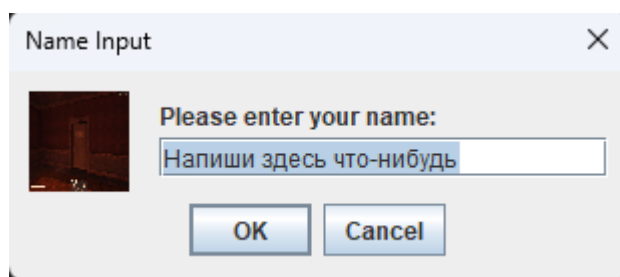
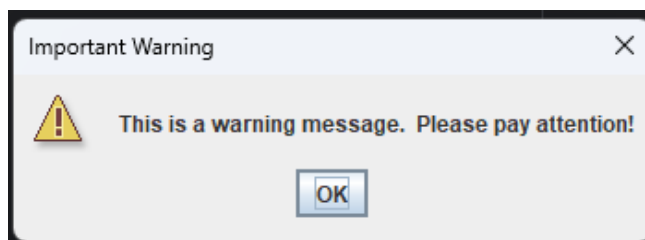
"C:\Program Files\Eclipse Adoptium\jdk-21.0.4.7-hotspot\bin\java.exe" "-javaagent:
12
13
32

2) Input02.

- Импортируйте и откройте проект Input02.
- Поэкспериментируйте с кодом и попробуйте изменить следующее ...
 - Заголовок сообщения
 - Сообщение
 - Любой введенный текст по умолчанию
 - Значок диалогового окна
- Анализируйте, выполняйте операции и выводите входные данные.

При изменении цифры от 0 до 3 меняется значок: 0 – красный крест, 1 – фиолетовый круг, 2 – желтый треугольник с восклицательным знаком, 3 – зеленый квадрат с вопросительным знаком.

Также можно менять сообщение(первые кавычки), заголовок(вторые кавычки), введенный текст(третьи кавычки) и изменять значения выбора:




```

import java.awt.image.BufferedImage;
import java.awt.Image;
import java.io.File;
import java.io.IOException;

import static java.awt.Image.SCALE_FAST;

public class Input02 {
    public static void main(String[] args) throws IOException {
        String imagePath = "123.png";
        File imageFile = new File(imagePath);
        BufferedImage bufferedImage = ImageIO.read(imageFile);
        Image image = bufferedImage.getScaledInstance(50, 50, SCALE_FAST);
        ImageIcon icon = new ImageIcon(image);

        JOptionPane.showMessageDialog(null,
            "This is a warning message. Please pay attention!",
            "Important Warning",
            JOptionPane.WARNING_MESSAGE);

        String input1 = (String) JOptionPane.showInputDialog(
            "Please enter your name:",
            "Name Input",
            JOptionPane.QUESTION_MESSAGE,
            icon,
            null,
            "Напиши здесь что-нибудь");

        String[] acceptableValues = {"Option A", "Option B", "Option C"};
        String input2 = (String)JOptionPane.showInputDialog(
            "Choose an option:",
            "Option Selection",
            JOptionPane.QUESTION_MESSAGE,
            null,
            acceptableValues,
            acceptableValues[2]);
    }
}

```

3) Input03.

- Импортируйте и откройте проект Input03.
- Создайте класс Scanner:
- В NetBeans отобразится предупреждение.
- Следуя рекомендации NetBeans, импортируйте java.util.Scanner
- Не забудьте закрыть Scanner.
- С помощью класса Scanner и System.in напишите программу, которая будет выполнять следующие задачи ...
- Вычислять и выводить сумму трех целых чисел, введенных пользователем.
- Попробуйте использовать менее трех маркеров.

— Попробуйте использовать маркер, который нельзя проанализировать как int.

```
import java.util.Scanner;

class Input03 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int x = sc.nextInt();
        int y = sc.nextInt();
        int z = sc.nextInt();

        System.out.println(x + y + z);

        sc.close();
    }
}

Input03 x
```

При попытке ввести маркер, который нельзя проанализировать как `int`, программа прекратила работу и вывела ошибку:

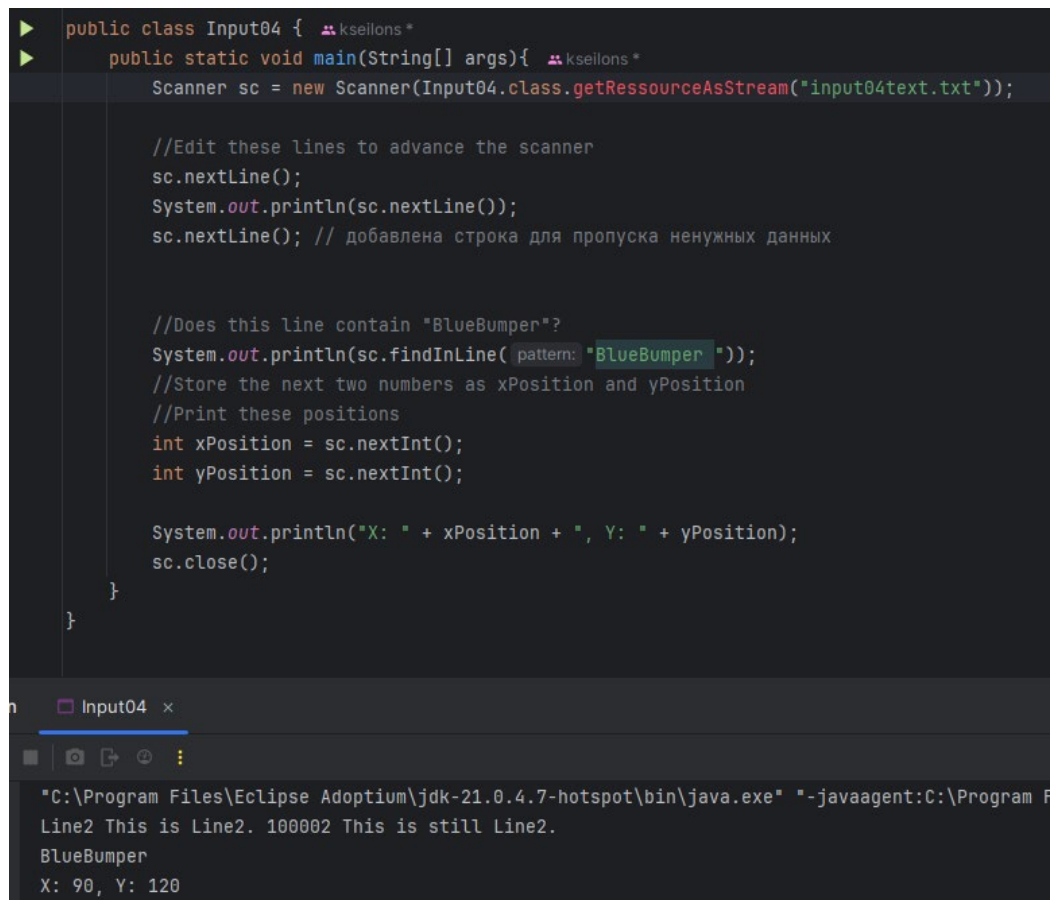
```
2ddfg
Exception in thread "main" java.util.InputMismatchException Create breakpoint
    at java.base/java.util.Scanner.throwFor(Scanner.java:947)
    at java.base/java.util.Scanner.next(Scanner.java:1602)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2267)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2221)
    at input03.Input03.main(Input03.java:9)
```

4) Input04.

Часть 1.

- Импортируйте и откройте проект Input04.
- Выполните код и изучите результат.

- Пролитайте все строки, пока не дойдете до "BlueBumper".
- Два числовых значения после "BlueBumper" — это объекты xPosition и yPosition. Сохраните эти координаты как значения integer и выведите их.
- При необходимости изучите файл input04text.txt.



```

public class Input04 {
    public static void main(String[] args){
        Scanner sc = new Scanner(Input04.class.getResourceAsStream("input04text.txt"));

        //Edit these lines to advance the scanner
        sc.nextLine();
        System.out.println(sc.nextLine());
        sc.nextLine(); // добавлена строка для пропуска ненужных данных

        //Does this line contain "BlueBumper"?
        System.out.println(sc.findInLine( pattern: "BlueBumper "));
        //Store the next two numbers as xPosition and yPosition
        //Print these positions
        int xPosition = sc.nextInt();
        int yPosition = sc.nextInt();

        System.out.println("X: " + xPosition + ", Y: " + yPosition);
        sc.close();
    }
}

```

Input04 x

```

"C:\Program Files\Eclipse Adoptium\jdk-21.0.4.7-hotspot\bin\java.exe" "-javaagent:C:\Program F
Line2 This is Line2. 100002 This is still Line2.
BlueBumper
X: 90, Y: 120

```

Часть 2.

Если вам интересно, изучите файл Level05.txt:

- В этом файле хранятся данные уровней игры Java Puzzle Ball.
- Операции чтения и анализа данных уровней немного сложнее, чем в этом упражнении.
- Но если вы выполнили это упражнение, вы приблизились к пониманию того, как это происходит.

```

import java.io.InputStream;
import java.util.Scanner;

public class Input05 { new *
    public static void main(String[] args) { new *
        InputStream inputStream = Input05.class.getResourceAsStream(name: "Level05.txt");
        if (inputStream == null) {
            System.err.println("@айл Level05.txt не найден!");
            return;
        }
        Scanner sc = new Scanner(inputStream);

        while (sc.hasNextLine()) {
            String line = sc.nextLine();
            // Пропускаем строку, если она начинается с двух слешов
            if (line.startsWith("//")) {
                continue;
            }
            String[] parts = line.split(regex: "\\s+"); // Разделяем строку по пробелам

            if (parts.length >= 3) { // Проверяем, достаточно ли частей в строке
                String name = parts[0];
                try {
                    int x = Integer.parseInt(parts[1]);
                    int y = Integer.parseInt(parts[2]);
                    System.out.println(name + ":\n x = " + x + ", y = " + y);
                } catch (NumberFormatException e) {
                    System.err.println("Ошибка при парсинге координат в строке: " + line);
                }
            } else {
                System.out.println(line); // Выводим строку как есть, если координат нет
            }
        }

        sc.close();
    }
}

```

1. Упражнения – Раздел 3.

Цель — создать программу, подобную JavaLibs. Написать рассказ, отдельные части конечного текста которого изменяются в зависимости от ввода пользователя. Предложить пользователю различные варианты ввода.

```

public class Main { new *
    public static void main(String[] args) { new *
        JOptionPane.showMessageDialog(parentComponent: null, message: "Привет, ответь на парочку вопросов, а мы составим интересную историю про тебя");
        String name = (String) JOptionPane.showInputDialog(parentComponent: null,
            message: "Как вас зовут?",
            title: "Введите имя",
            JOptionPane.QUESTION_MESSAGE,
            icon: null,
            selectionValues: null,
            initialSelectionValue: "Например, Андрей");

        int age = Integer.parseInt((String) JOptionPane.showInputDialog(parentComponent: null,
            message: "Сколько вам лет?",
            title: "Введите возраст",
            JOptionPane.QUESTION_MESSAGE,
            icon: null,
            selectionValues: null,
            initialSelectionValue: "Например, 18"));

        double height = Double.parseDouble((String) JOptionPane.showInputDialog(parentComponent: null,
            message: "Ваш рост (в метрах)?",
            title: "Введите рост",
            JOptionPane.QUESTION_MESSAGE,
            icon: null,
            selectionValues: null,
            initialSelectionValue: "Например, 185"));

        int numberOfPets = Integer.parseInt((String) JOptionPane.showInputDialog(parentComponent: null,
            message: "Сколько у вас питомцев?",
            title: "Ваши животные",
            JOptionPane.QUESTION_MESSAGE,
            icon: null,
            selectionValues: null,
            initialSelectionValue: "Например, 2"));
    }
}

```

```

double weight = Double.parseDouble((String) JOptionPane.showInputDialog( parentComponent: null,
    message: "Ваш вес (в килограммах)?",
    title: "Ваш вес",
    JOptionPane.QUESTION_MESSAGE,
    icon: null,
    selectionValues: null,
    initialValue: "Например, 55"));

String city = (String) JOptionPane.showInputDialog( parentComponent: null,
    message: "В каком городе вы живете?",
    title: "Город",
    JOptionPane.QUESTION_MESSAGE,
    icon: null,
    selectionValues: null,
    initialValue: "Например, Кировск");

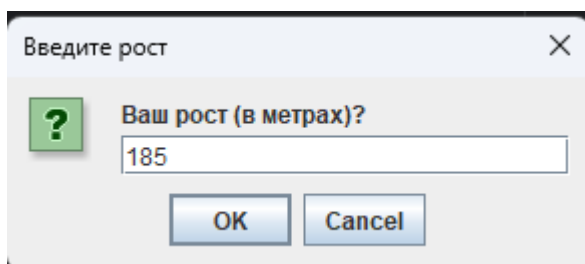
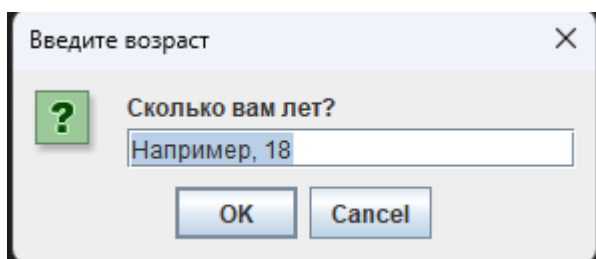
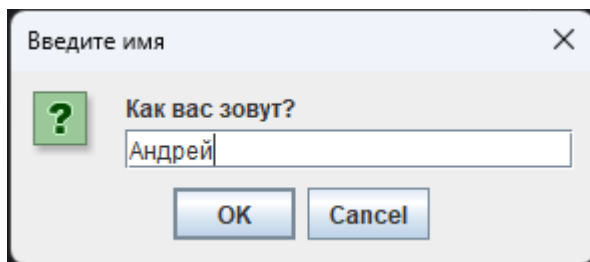
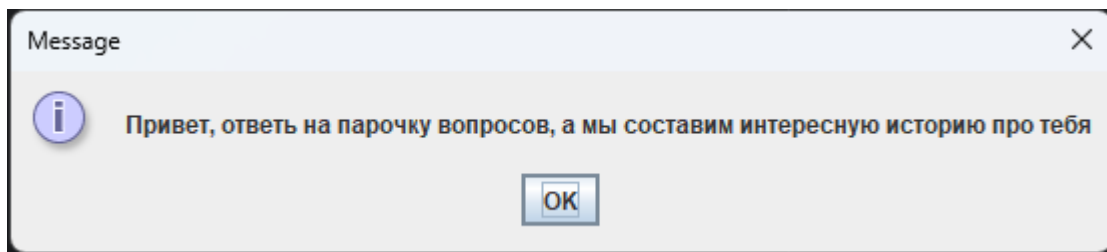
int luckyNumber = Integer.parseInt((String) JOptionPane.showInputDialog( parentComponent: null,
    message: "Ваше счастливое число?",
    title: "Счастливое число",
    JOptionPane.QUESTION_MESSAGE,
    icon: null,
    selectionValues: null,
    initialValue: "Например, 123"));

int ageNextYear = age + 1;
double bmi = weight / (height * height);

String story = "Однажды, в городе " + city + ", жил человек по имени " + name + ".\n";
story += "Ему было " + age + " лет, и в следующем году ему исполнится " + ageNextYear + ".\n";
story += "Рост " + name + " составлял " + height + " метров, а вес - " + weight + " килограммов. Его индекс массы тела равен " + String.format("%.2f", bmi) + ".\n";
story += "У " + name + " было " + numberOfPets + " питомцев, и его счастливое число - " + luckyNumber + ".\n";
story += "И жили они долго и счастливо (или нет...).";
JOptionPane.showMessageDialog( parentComponent: null, story, title: "Ваша история", JOptionPane.PLAIN_MESSAGE);
}
}

```

Работа программы:



Ваши животные

Сколько у вас питомцев?

2

OK Cancel

Ваш вес

Ваш вес (в килограммах)?

55

OK Cancel

Город

В каком городе вы живете?

Кировск

OK Cancel

Счастливое число

Ваше счастливое число?

Например, 123

OK Cancel

Вывод:

Ваша история

Однажды, в городе Кировск, жил человек по имени Андрей.
Ему было 18 лет, и в следующем году ему исполнится 19.
Рост Андрей составлял 185.0 метров, а вес — 55.0 килограммов. Его индекс массы тела равен 0,00.
У Андрей было 2 питомцев, и его счастливое число — 123.
И жили они долго и счастливо (или нет...).

OK

2. Задания по уроку 4.1.

1) Tip01.

У Алекса день рождения. Вы собрались группой из восьми человек в местном ресторане, чтобы отметить праздник. Когда вам принесли счет, никто толком не знал, сколько должен заплатить. Вам известна только общая сумма без учета налога (5%) и чаевых (15%). Но вам повезло! Вы взяли с собой ноутбук, и вас попросили написать программу, которая рассчитает общую сумму для каждого человека.

```
import java.util.HashMap;
import java.util.Map;

public class Tip01 {
    public static void main(String[] args) {
        double taxRate = 0.05, tipRate = 0.15;
        Map<String, Double> initialAmounts = new HashMap<>();
        initialAmounts.put("Person 1", 10.0);
        initialAmounts.put("Person 2", 12.0);
        initialAmounts.put("Person 3", 9.0);
        initialAmounts.put("Person 4", 8.0);
        initialAmounts.put("Person 5", 7.0);
        initialAmounts.put("Person 6", 15.0);
        initialAmounts.put("Person 7", 11.0);
        initialAmounts.put("Person 8", 30.0);

        for (Map.Entry<String, Double> entry : initialAmounts.entrySet()) {
            String person = entry.getKey();
            double amount = entry.getValue();
            double total = amount * (1 + taxRate + tipRate);
            System.out.println(person + ": $" + String.format("%.2f", total));
        }
    }
}
```

Tip01 x

"C:\Program Files\Eclipse Adoptium\jdk-21.0.4-hotspot\bin\java.exe" "-javaagent:C:
Person 1: \$12,00
Person 2: \$14,40
Person 7: \$13,20
Person 8: \$36,00
Person 3: \$10,80
Person 4: \$9,60
Person 5: \$8,40
Person 6: \$18,00

2) Tip02.

- Импортируйте и откройте проект Tip02.
- Дополните метод findTotal(), который должен выполнять следующие функции:
 - Вычислять итоговое значение с учетом полей tax, tip и originalPrice
 - Выводить итог для каждого человека
- В главном методе:
 - Создайте экземпляр объекта Calculator с именем calc.

- Посмотрите, что произойдет в NetBeans, если ввести слово "calc".
- Получите доступ к полям и методам этого объекта, чтобы вывести итог для каждого гостя на дне рождения.
- Измените tip и tax, если предпочитаете использовать другие значения.

```
package Tip02;

public class Calculator { 2 usages
    public double tax = .05; 1 usage
    public double tip = .15; //Change tax and tip if you prefer different values 1 usage
    public double originalPrice = 0; 2 usages

    public void findTotal(){ 1 usage
        double total = originalPrice * (1 + tax + tip);
        System.out.printf("%.2f\n", total);
    }
}
```

CalculatorTest x

"C:\Program Files\Eclipse Adoptium\jdk-21.0.4.7-hotspot\bin\java.exe" "-javaagent:C:\Program Files\

12,00
14,40
13,20
36,00
10,80
9,60
8,40
18,00


```

public class CalculatorTest {
    public static void main(String[] args) {
        Calculator calc = new Calculator();
        Map<String, Double> initialAmounts = new HashMap<>();
        initialAmounts.put("Person 1", 10.0);
        initialAmounts.put("Person 2", 12.0);
        initialAmounts.put("Person 3", 9.0);
        initialAmounts.put("Person 4", 8.0);
        initialAmounts.put("Person 5", 7.0);
        initialAmounts.put("Person 6", 15.0);
        initialAmounts.put("Person 7", 11.0);
        initialAmounts.put("Person 8", 30.0);

        for (Map.Entry<String, Double> entry : initialAmounts.entrySet()) {
            String person = entry.getKey();
            calc.originalPrice = entry.getValue();
            calc.findTotal();
        }
    }
}

```

3) Tip03.

- Импортируйте и откройте проект Tip03.
- В главном методе:
 - Используйте экземпляр объекта Calculator и передайте аргументы в метод findTotal(), чтобы вывести итог для каждого человека.
 - Подсказка. Изучите метод findTotal() в классе Calculator и выясните, сколько аргументов в него можно передать.
К кому относится каждое итоговое значение?
- Измените метод findTotal(), чтобы передать в него дополнительный аргумент String name. Объедините оператор вывода, чтобы включить аргумент name.
- В NetBeans появится сообщение об ошибке в главном методе. Проверьте вызовы метода findTotal().

```
public class CalculatorTest {
    public static void main(String[] args) {
        Calculator calc = new Calculator();

        Map<String, Double> initialAmounts = new HashMap<>();
        initialAmounts.put("Person 1", 10.0);
        initialAmounts.put("Person 2", 12.0);
        initialAmounts.put("Person 3", 9.0);
        initialAmounts.put("Person 4", 8.0);
        initialAmounts.put("Person 5", 7.0);
        initialAmounts.put("Person 6", 15.0);
        initialAmounts.put("Person 7", 11.0);
        initialAmounts.put("Person 8", 30.0);

        for (Map.Entry<String, Double> entry: initialAmounts.entrySet()) {
            String person = entry.getKey();
            Double price = entry.getValue();
            calc.findTotal(person, price);
        }
    }
}

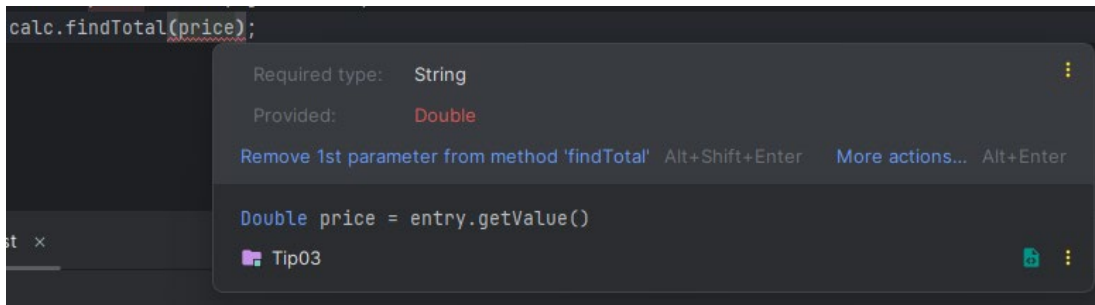
CalculatorTest x
C:\Program Files\Eclipse Adoptium\jdk-21.0.4.7-hotspot\bin\java.exe "-javaage
Person 1: $12.0
Person 2: $14.399999999999999
Person 7: $13.2
Person 8: $36.0
Person 3: $10.799999999999999
Person 4: $9.6
Person 5: $8.4
Person 6: $18.0
```

```
package tip03;

public class Calculator { 2 usages
    public double tax = .05; 1 usage
    public double tip = .15; 1 usage

    public void findTotal(String name, double price){ 1 usage
        double total = price*(1+tax+tip);
        System.out.println(name + ": $" +total);
    }
}
```

Появятся ошибки, т.к. мы подаем один аргумент вместо двух.



4) Tip04.

- Отредактируйте решение Tip03 или импортируйте проект Tip04.
- Вычислите и выведите итоговое значение счета для всех гостей с учетом налога и чаевых.
— Для этого потребуется отредактировать метод findTotal(), чтобы он возвращал вычисленное значение.
- Гость 8 забыл бумажник. А ужин для Алекса был подарком на его день рождения. Измените метод findTotal(), чтобы разделить стоимость остальных блюд поровну между остальными гостями.
- Пересчитайте итог для всех гостей. Это значение не должно измениться.

```
package tip04;

public class Calculator { 2 usages
    public double tax = .05; 1 usage
    public double tip = .15; 1 usage

    //Include the cost of Alex's and Forgetful's meals in your calculations
    //Return the total after calculating
    public double findTotal(double price, String name){ 1 usage
        double total = price*(1+tax+tip);
        System.out.println(name + ": $" + total);
        return total;
    }
}
```

```
public class CalculatorTest {
    public static void main(String[] args) {
        Calculator calc = new Calculator();

        Map<String, Double> initialAmounts = new HashMap<>();
        initialAmounts.put("Person 1", 10.0);
        initialAmounts.put("Person 2", 12.0);
        initialAmounts.put("Person 3", 9.0);
        initialAmounts.put("Person 4", 8.0);
        initialAmounts.put("Person 5", 7.0);
        initialAmounts.put("Alex", 15.0);
        initialAmounts.put("Person 7", 11.0);
        initialAmounts.put("Forgetful", 30.0);
        double total = 0;
        for (Map.Entry<String, Double> entry: initialAmounts.entrySet()) {
            String person = entry.getKey();
            Double price = entry.getValue();
            total += calc.findTotal(price, person);
        }

        System.out.println("Общий счет: $" + total);
    }
}
```

CalculatorTest x

"C:\Program Files\Eclipse Adoptium\jdk-21.0.4-hotspot\bin\java.exe" "-javaage
Alex: \$18.0
Person 1: \$12.0
Person 2: \$14.399999999999999
Forgetful: \$36.0
Person 7: \$13.2
Person 3: \$10.799999999999999
Person 4: \$9.6
Person 5: \$8.4
Общий счет: \$122.4

Выполним вторую часть задания:

```
package tip04;

public class Calculator { 2 usages
    public double tax = .05; 2 usages
    public double tip = .15; 2 usages

    //Include the cost of Alex's and Forgetful's meals in your calculations
    //Return the total after calculating
    public double findTotal(double price, String name){ 1 usage
        double common_price = (15 + 30) * (1 + tax + tip) / 6;
        double total = price*(1+tax+tip) + common_price;
        System.out.println(name + ": $" + total);
        return total;
    }
}
```

```
public class CalculatorTest {
    public static void main(String[] args) {
        Calculator calc = new Calculator();

        Map<String, Double> initialAmounts = new HashMap<>();
        initialAmounts.put("Person 1", 10.0);
        initialAmounts.put("Person 2", 12.0);
        initialAmounts.put("Person 3", 9.0);
        initialAmounts.put("Person 4", 8.0);
        initialAmounts.put("Person 5", 7.0);
        // initialAmounts.put("Alex", 15.0);
        initialAmounts.put("Person 7", 11.0);
        // initialAmounts.put("Forgetful", 30.0);
        double total = 0;
        for (Map.Entry<String, Double> entry: initialAmounts.entrySet()) {
            String person = entry.getKey();
            Double price = entry.getValue();
            total += calc.findTotal(price, person);
        }

        System.out.println("Общий счет: $" + total);
    }
}

CalculatorTest x
Person 1: $21.0
Person 2: $23.4
Person 7: $22.2
Person 3: $19.799999999999997
Person 4: $18.6
Person 5: $17.4
Общий счет: $122.4
```

Общая сумма не поменялась.

У третьего гостя сумма в таком виде потому, что не все десятичные числа можно представить в типе double в виде конечного двоичного числа.

3. Задания по уроку 4.2.

- Импортируйте и откройте проект ImportEx.
- Изучите AddImport.java.
- Выполните следующие действия.
 - Замените полное имя, используемое для доступа к компоненту JLabel, оператором import.
 - Чтобы импортировать классы из пакета util, замените несколько операторов import одним оператором import.

```
package com.example;

import javax.swing.JLabel;
import java.util.*;

public class AddImport {  kseilons *

    public static void main(String args[]) {  kseilons *
        javax.swing.JLabel label = new JLabel(text: "hello");
    }

}
```

 AddImport x

"C:\Program Files\Eclipse Adoptium\jdk-21.0.4.7-hotspot\bin\java.exe" "-java

Process finished with exit code 0

1. Конспект.

Кроме базовых типов данных широко используются соответствующие им классы оболочки (wrapper – classes): Boolean, Character, Integer, Byte, Short, Long, Float, Double. Объекты этих классов могут хранить те же значения, что и соответствующие им базовые типы, но преобразуются в надлежащие классы, у которых появляется возможность использовать методы этих классов.

Другие символы, которые также называются операторами в языке Java, в том числе:

- точка "." указывает на имена пакетов и вызывает методы;
- круглые скобки "(") отделяют список параметров, разделенных запятыми, от имени метода;
- new создает экземпляр объекта (если за ним следует имя конструктора объекта).

Конструктор - специальный метод класса, который используется для создания объекта.

Класс - это некоторый шаблон, на основании которого создаются экземпляры класса - объекты.

Язык Java содержит операторы, которые позволяют программе принимать решения. Операторы отношения (>, >=, <=, ==, !=) используются, чтобы проверить отношение между двумя операндами, и возвращают boolean значение "истина" (true) или "ложь" (false) в зависимости от результата сравнения значений отношения.

Рассмотрим отложенные операторы. Оператор >> означает в языке Java сдвиг вправо. Он перемещает все биты своего левого операнда вправо на число позиций, заданное правым операндом. Когда биты левого операнда выдвигаются за самую правую позицию слова, они теряются. При сдвиге вправо освобождающиеся старшие (левые) разряды сдвигаемого числа заполняются предыдущим содержимым знакового разряда. Такое поведение называют расширением знакового разряда. Оператор << выполняет сдвиг влево всех битов своего левого операнда на число позиций, заданное правым операндом. При этом часть битов в левых разрядах выходит за границы и теряется, а соответствующие правые позиции заполняются нулями. Здесь используется автоматическое повышение типа всего выражения до int, в том случае, если в выражении присутствуют операнды типа int или целые типы меньшего размера. Если же хотя бы один из операндов в выражении имеет тип long, то и тип всего выражения повышается до long.

В методе printf сначала задается шаблон, согласно которому будет отформатирована строка, а потом передаются объекты для форматирования. Вот как выглядит метод printf с параметрами: System.out.printf(String format, Object... args); Формат обычно заключается в двойные кавычки, и определяет содержание вывода и тип выводимых данных с помощью следующих символов, начинающихся со знака "%" (Далее для простоты текста вместо System.out.printf() пишется printf()): %s - для типа String, например, printf("Hello %s!", "World"); // выводится "Hello World!" %n в составе формата выполняет переход на другую строку в выводе. Для типов byte, short, int, long: %d - вывод в десятичном формате. %x - вывод в шестнадцатеричном формате. Здесь также можно указывать

ширину поля вывода, например, так: `%7d` - в десятичном формате и минимальной шириной поля 7 знаков. Оператор `printf("%7d", 1);` // выводит " 1"

Для типов `float`, `double`. `%f` - Десятичное число с точкой. `%e` - Десятичное число с точкой и экспонентой. Например, `%.10f` выводит с точностью 10 знаков после запятой. В отличие от `println`, `printf` использует не конкатенацию строк, а список переменных через запятую в порядке, определенном строкой формата.

Язык Java поддерживает условную операцию, которая также называется тернарная и записывается следующим образом: `? : < выражение2>`; Если истинно, то результатом будет , в противном случае . Например, `int x = a < b ? a : b;` объявляет переменную `x` и присваивает вычисленный минимум значений из `a` и `b`.

Блок, или составной оператор — это произвольный состав простых операторов языка Java, заключенных в фигурные скобки. Блоки (как и методы) определяют область видимости своих переменных.