Частное учреждение образования

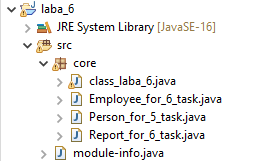
«Колледж бизнеса и права»

Тема: «Работа со строками в языке Java»

Учащийся М.Н. Гончаров.

группа: Т-992

2021



Рисунок

Текст задачи 1,2,3,4,5,6,7,8,9,10

1. е) Найти позицию подстроки “Java” в строке “I like Java!!!”. – Т.к я 6 вариант

2. Напишите метод, заменяющий в строке каждое второе вхождение

«object-oriented programming» (не учитываем регистр символов) на «OOP».

Например строка. "Object-oriented programming is a programming language

model organized around objects rather than "actions" and data rather than logic.

Object-oriented programming blabla. Object-oriented programming bla."

должна быть преобразована в "Object-oriented programming is

a programming language model organized around objects rather than "actions"

and data rather than logic. OOP blabla. Object-oriented programming bla."

3. Даны строки разной длины (длина - четное число), необходимо

вернуть ее два средних знака: "string" → "ri", "code" → "od",

"Practice"→"ct".

4. Создать строку, используя форматирование: Студент [Фамилия]

получил [оценка] по [предмету]. Форматирование и вывод строки на

консоль написать в отдельном методе, который принимает фамилию, оценку и название предмета в качестве параметров. Выделить под

фамилию 15 символов, под оценку 3 символа, предмет – 10.

5. Переделать toString() для класса любого класса (например Person)

используя форматирование.

6. Создать класс Employee, у которого есть переменные класса -

fullname, salary. Создать массив содержащий несколько объектов этого

типа. Создать класс Report, который будет содержать статический

метод generateReport, в котором выводится информация о зарплате всех

сотрудников. Используйте форматировании строк. Пусть salary будет

выровнено по правому краю, десятичное значение имеет 2 знака после

запятой и можете добавить что-нибудь свое.

7. Дана строка “Versions: Java 5, Java 6, Java 7, Java 8, Java 12.”

Найти все подстроки "Java X" и распечатать их.

8. Найти слово, в котором число различных символов минимально.

Слово может содержать буквы и цифры. Если таких слов несколько, найти

первое из них. Например в строке "fffff ab f 1234 jkjk" найденное слово

должно быть "fffff".

9. Предложение состоит из нескольких слов, разделенных

пробелами. Например: "One two three раз два три one1 two2 123 ". Найти

количество слов, содержащих только символы латинского алфавита.

10. Предложение состоит из нескольких слов, например: "Если есть

хвосты по дз, начните с 1 не сданного задания. 123 324 111 4554". Среди

слов, состоящих только из цифр, найти слово-палиндром.

**// Main и функции**

package core;

import static java.lang.System.out;

import java.util.ArrayList;

import java.util.Formatter;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import java.util.Scanner;

public class class\_laba\_6 {

// 3

public static String return\_two\_char\_on\_center(String s)

{

String s\_2 = "";

int sered = (int)(s.length() - 1) / 2;

s\_2 += s.charAt(sered);

// Если не четная и там 1 символ в середине

if (s.length() % 2 == 0)

{

s\_2 += s.charAt(sered+1);

}

return s\_2;

}

// 7

public static void print\_Java\_X(String str)

{

String pStr = "Java\\s+\\d{1,2}";

Pattern p = Pattern.compile(pStr);

Matcher m = p.matcher(str);

while (m.find())

{

System.out.println(m.group());

}

}

// 9

public static int count\_eng\_word(String str)

{

int flag = 0;

String[] arr\_str = str.split(" "); // Разбил по пробелам

for(int i = 0; i < arr\_str.length; i++)

{

if (arr\_str[i].matches("[a-zA-z]+")) // Соответствие на eng буквы

{

flag++;

}

}

return flag;

}

// 10.1

public static boolean is\_palindrome(int x)

{

char[] s = Integer.toString(x).toCharArray();

for (int i = 0, i\_end = s.length - 1; i < (int)(s.length-1) / 2; i++, i\_end--)

{

if (s[i] != s[i\_end])

{

return false;

}

}

return true;

}

// 10.2

public static ArrayList<Integer> find\_palindromes(String x) // поиск чисел

{

String[] words = x.split(" ");

ArrayList<Integer> palindromes = new ArrayList<Integer>(); // тупо массив

for(int i = 0; i < words.length; i++)

{

if (words[i].matches("[0-9]+")) // проверка на регулярку ( + это от 1 и больше элементов того что слева)

{

int x\_znach = Integer.parseInt(words[i].trim()); // Получение самого числа

if (is\_palindrome(x\_znach)) // проверка его на палиндром

{

palindromes.add(x\_znach);

}

}

}

return palindromes;

}

public static void main(String[] args) {

// Консольный ввод import java.util.Scanner;

Scanner in = new Scanner(System.in);

// а

System.out.println("\nTask 1");

//System.out.println("Введите строку: (Считает с первого пробела)"); /// ЧТОБЫ КАЖДЫЙ РАЗ НЕ ВВОДИТЬ

//String str = in.nextLine();

/\*

// б

System.out.println(str.charAt(str.length()-1));

// в

System.out.println(str.endsWith("!!!") ? true:false);

// г

System.out.println(str.startsWith("I like") ? true : false);

// д

System.out.println(str.contains("Java") ? true : false);

\*/

// e

String str = "I love Java!";

System.out.println(str.indexOf("Java")); // I love Java! -- 7

// ж

//System.out.println(str.replace('a', 'o'));

// 2

System.out.println("\nTask 2");

// Исходная строка

String str2 = "Object-oriented programming is a programming language model organized around objects rather than \"actions\" and data rather than logic. Object-oriented programming blabla. Object-oriented programming bla.";

/\*

\*

"Object-oriented programming is

a programming language model organized around objects rather than "actions"

and data rather than logic. OOP blabla. Object-oriented programming bla."

кое что воял в итоге внизу все одной функцией

String[] arr\_str2 = str2.split(".");

for (int i = 0; i < arr\_str2.length; i++) // Начинаю с 1 т.к там будет первое вхождение

{

System.out.println(arr\_str2[i]);

}

str2 = "";

for (int i = 1; i < arr\_str2.length; i++) // Начинаю с 1 т.к там будет первое вхождение

{

if(i % 2 != 0)

{

arr\_str2[i] = arr\_str2[i].replace("Object-oriented programming", "OOP");

}

str2 += arr\_str2[i];

}

\*/

System.out.println(str2.replaceFirst(" Object-oriented programming"," OOP"));

// 3

System.out.println("\nTask 3");

String str\_1 = "string";

String str\_2 = "code";

String str\_3 = "Practice";

System.out.println(return\_two\_char\_on\_center(str\_1));

System.out.println(return\_two\_char\_on\_center(str\_2));

System.out.println(return\_two\_char\_on\_center(str\_3));

// 4

System.out.println("\nTask 4");

String name = "Matvey";

int mark = 10;

String predmet = "Java";

System.out.println(String.format("Студент %15s получил %3d по %10s", name, mark, predmet));

// 5

System.out.println("\nTask 5");

Person\_for\_5\_task p1 = new Person\_for\_5\_task(); // Там по дефолту указал значения для теста

System.out.println(p1); // Сработает метод toString()

// 6

System.out.println("\nTask 6");

Employee\_for\_6\_task[] emp1 = new Employee\_for\_6\_task[] { new Employee\_for\_6\_task("fullname1",557.10), new Employee\_for\_6\_task("fullname2",725.20), new Employee\_for\_6\_task("fullname3",345.56)}; // Заполнил

Report\_for\_6\_task.generateReport(emp1);

// 7

System.out.println("\nTask 7");

String str7 = "Versions: Java 5, \_Java X, Java 6, Java 7, Java 8, Java 12.";

print\_Java\_X(str7);

// 8

System.out.println("\nTask 8");

String str8 = "fffff ab f 1234 jkjk";

String[] words = str8.split(" ");

/\*

Метод distinct возвращает stream без дубликатов, при этом для упорядоченного стрима (например, коллекция на основе list) порядок стабилен, для неупорядоченного — порядок не гарантируется

\*/

int wordIndex = (int)words[0].chars().distinct().count(); // индекс первого слова с кол-вом подсчета не повторающихся символов

String word = words[wordIndex];

for(int i = 0; i < words.length; i++)

{

if (words[i].chars().distinct().count() < words[wordIndex].chars().distinct().count())

{

wordIndex = i;

word = words[wordIndex];

}

}

System.out.println(words[wordIndex]);

// 9

System.out.println("\nTask 9");

String str9 = "One two three раз два три one1 two2 123 ";

System.out.println("Кол-во только eng слов: " + count\_eng\_word(str9));

// 10

System.out.println("\nTask 10");

String str10 = "asdfasd a 123321 sdfa 1551 dsf 3436 234asdfasdfasdf";

for ( int i : find\_palindromes(str10))

System.out.println(i);

}

}

// Employee\_for\_6\_task

**package** core;

**public** **class** Employee\_for\_6\_task

{

**private** String fullName;

**private** **double** salary;

**public** Employee\_for\_6\_task() {}

**public** Employee\_for\_6\_task(String fullName, **double** salary)

{

**this**.fullName = fullName;

**this**.salary = salary;

}

**public** String getFullName()

{

**return** fullName;

}

**public** **void** setFullName(String fullName)

{

**this**.fullName = fullName;

}

**public** **double** getSalary()

{

**return** **this**.salary;

}

**public** **void** setSalary(**int** salary)

{

**this**.salary = salary;

}

}

// Person\_for\_5\_task

**package** core;

**public** **class** Person\_for\_5\_task {

**private** String name;

**private** String surname;

// Сразу заполнил чтобы тупо показать что работает, при создании объекта

**public** Person\_for\_5\_task()

{

**this**.name = "default\_name";

**this**.surname = "default\_surname";

}

**public** Person\_for\_5\_task(String name, String surname)

{

**this**.name = name;

**this**.surname = surname;

}

@Override

**public** String toString() {

**return** String.*format*("Name: %s Surname: %s", **this**.name, **this**.surname);

}

}

// Report\_for\_6\_task

**package** core;

**public** **class** Report\_for\_6\_task

{

**public** **static** **void** generateReport(Employee\_for\_6\_task[] employees)

{

**for** (Employee\_for\_6\_task emp : employees)

System.***out***.printf(String.*format*("FullName: %s Зарплата: %5.2f\n", emp.getFullName(), emp.getSalary()));

}

}

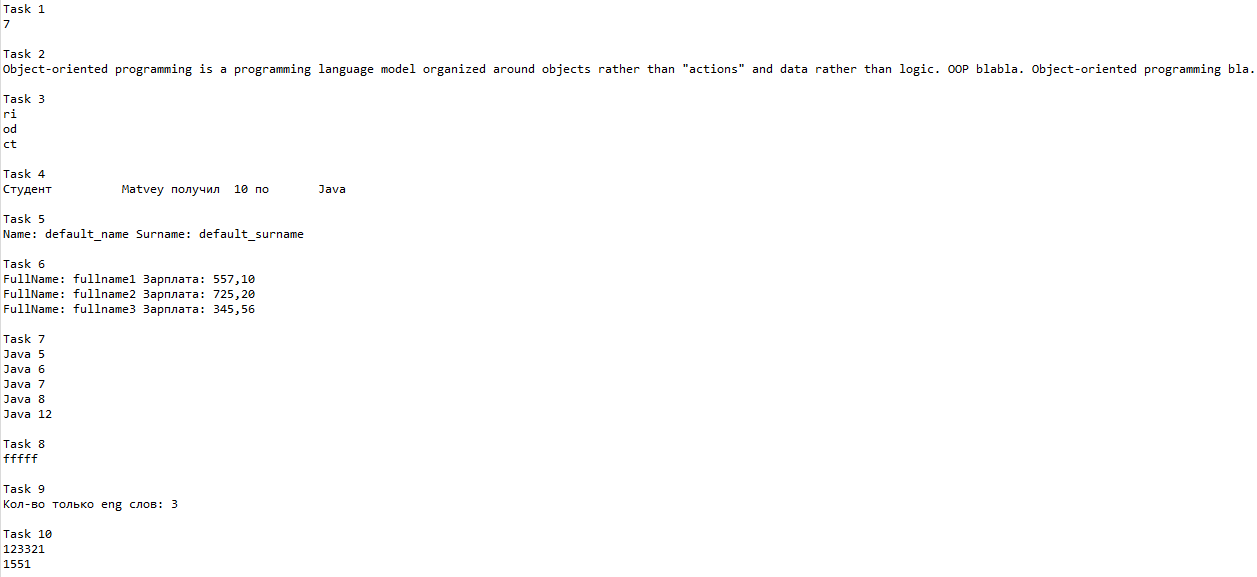


Рисунок 2