**Курс «Программирование с использованием технологии Java и СУБД Oracle»**

**Тема: Введение в язык программирования Java.**

**Домашняя работа №1**

Выполнил студент гр. ВПУ-911: Мулин Николай Сергеевич

IDE: IntelliJ IDEA 2021.3

Ссылка на проект в GitHub: <https://github.com/Nickomu92/hw1_Java>

**Условия заданий**

*1.**Организовать ввод с клавиатуры даты рождения человека, программа должна вывести знак зодиака и название года по японскому календарю. Предполагаем, что пользователь всегда корректно вводит данные.*

*Пример входных данных:*

*5 декабря 1974 г*

*Вывод:*

*Знак: стрелец Год: тигра*

*2. Программа запрашивает шестизначное число, после ввода определяет будет ли являтся счастливым билет с таким номером (сумма первых трех цифр совпадает с сумой трех последних).*

*Пример входных данных:*

*421151*

*954832*

*Вывод:*

*Да*

*Нет*

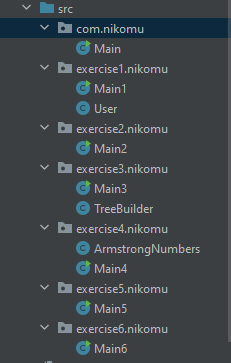
*3. Необходимо нарисовать ёлочку символом звёздочки. Каждый новый ярус должен быть шире предыдущего. С клавиатуры вводится количество ярусов, и высота первого (верхнего) яруса ёлочки. Ёлочка должна быть симметричная.*

*4. Показать на консоли все числа Армстронга в диапазоне от 0 до 10.000.000. Число Армстронга, или Самовлюблённое число, или совершенный цифровой инвариант или — натуральное число, которое равно сумме своих цифр, возведённых в степень, равную количеству его цифр. Например: 13 + 53 + 33 = 153.*

*5. Сформируйте массив из 10 элементов, заполнив его целыми случайными числами от –10 до 10. Определите среднее арифметическое элементов массива. Сформируйте новый массив, переписав в него элементы исходного массива, меньшие среднего. Выполните сортировку исходного массива по возрастанию элементов.*

*6. Сформируйте двумерный целочисленный массив. Число строк и столбцов нужно запросить у пользователя. Заполните массив случайными числами от 0 до 10 и выведите его на консоль. Удалите из массива одну строку (номер строки запросите у пользователя). Распечатайте полученный массив. Используйте статические методы для реализации каждого действия (инициализация массива, вывод на консоль, удаление строки).*

**Решение**



*Рис.1 – Общий вид структуры проекта.*

***Проект состоит из 7 пакетов и 10 классов:***

1. **Пакет com.nikomu (представляет главное меню проекта) c классами:**
   * **Main**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Общее "меню" для всех заданий для удобной работы.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

package com.nikomu;

// Импортируем пакеты, содержащие программы которые решают наши задачи

import exercise1.nikomu.Main1;

import exercise2.nikomu.Main2;

import exercise3.nikomu.Main3;

import exercise4.nikomu.Main4;

import exercise5.nikomu.Main5;

import exercise6.nikomu.Main6;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Переменная для хранения значения выбора пункта меню

String symb = null;

do {

System.out.println("\n\t\t\tMenu");

String menu = """

[1] - Задание № 1;

[2] - Задание № 2;

[3] - Задание № 3;

[4] - Задание № 4;

[5] - Задание № 5;

[6] - Задание № 6;

[0] - Выход.

""";

System.out.println(menu);

System.out.print("Ваш выбор: ");

symb = input.nextLine();

System.out.println();

switch (symb) {

case "1":

Main1.main(null);

break;

case "2":

Main2.main(null);

break;

case "3":

Main3.main(null);

break;

case "4":

Main4.main(null);

break;

case "5":

Main5.main(null);

break;

case "6":

Main6.main(null);

break;

}

} while(symb != "0");

}

}

1. **Пакет exercise1.nikomu (представляет решение задания №1) c классами:**

* **Main1**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Организовать ввод с клавиатуры даты рождения человека, программа должна

вывести знак зодиака и название года по японскому календарю. Предполагаем,

что пользователь всегда корректно вводит данные.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

package exercise1.nikomu;

import java.util.Scanner;

public class Main1 {

public static void main(String[] argc) {

Scanner input = new Scanner(System.in);

System.out.print("Введите дату Вашего рождения в формате [день месяц год]: ");

String date = input.nextLine();

// Создаем экземпляр нашего класса User

User user = new User(date);

user.GetInfo();

System.out.println("\n");

}

}

* **User**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Организовать ввод с клавиатуры даты рождения человека,

программа должна вывести знак зодиака и название года

по японскому календарю. Предполагаем, что пользователь

всегда корректно вводит данные.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

package exercise1.nikomu;

import java.time.LocalDate;

public class User {

private LocalDate date; // Поле для хранения даты

private String strDate; // Поле для хранения строкового представления даты

public User(String strDate) {

this.strDate = strDate;

}

// Метод для создания даты из её строкового представления

private void StringToDate() {

// Парисм дату рождения на составляющие - день, месяц, год

String[] dateElements = strDate.split("[\\s .,/\\-\_ ]+");

// Переменная для хранения дня

int day = Integer.parseInt(dateElements[0]);

// Переменная для хранения месяца

int month = 0;

// Переменная для хранения года

int year = Integer.parseInt(dateElements[2]);

// Временная переменная для проверки введенного названия месяца

String tempMonth = dateElements[1].toLowerCase();

if (tempMonth.contains("январ"))

month = 1;

else if (tempMonth.contains("феврал"))

month = 2;

else if (tempMonth.contains("март"))

month = 3;

else if (tempMonth.contains("апрел"))

month = 4;

else if (tempMonth.contains("ма"))

month = 5;

else if (tempMonth.contains("июн"))

month = 6;

else if (tempMonth.contains("июл"))

month = 7;

else if (tempMonth.contains("август"))

month = 8;

else if (tempMonth.contains("сентябр"))

month = 9;

else if (tempMonth.contains("октябр"))

month = 10;

else if (tempMonth.contains("ноябр"))

month = 11;

else if (tempMonth.contains("декабр"))

month = 12;

else if (tempMonth.matches("[-+]?\\d+")) {

if (Integer.parseInt(tempMonth) >= 1 && Integer.parseInt(tempMonth) <= 12)

month = Integer.parseInt(tempMonth);

}

// Создаем дату рождения на основании полученных от пользователя данных

date = LocalDate.of(year, month, day);

}

// Метод для получения информации

public void GetInfo() {

StringToDate();

System.out.println("Знак: " + GetClassicZodiac() + ";");

System.out.printf("Год: %s.", GetJapaneseZodiac());

}

// Метод для получение информации по классическому календарю

public String GetClassicZodiac() {

// Парсим дату рождения пользователя

int day = this.date.getDayOfMonth();

int month = this.date.getMonthValue();

// Переменная для хранения информации по знаку зодиака

String classicZodiac = null;

if ((day >= 21 && day <= 31 && month == 3) || (day <= 20 && month == 4))

classicZodiac = "Овен";

else if ((day >= 21 && day <= 30 && month == 4) || (day <= 21 && month == 5))

classicZodiac = "Телец";

else if ((day >= 22 && day <= 31 && month == 5) || (day <= 21 && month == 6))

classicZodiac = "Близнецы";

else if ((day >= 23 && day <= 30 && month == 6) || (day <= 22 && month == 7))

classicZodiac = "Рак";

else if ((day >= 23 && day <= 31 && month == 7) || (day <= 21 && month == 8))

classicZodiac = "Лев";

else if ((day >= 22 && day <= 31 && month == 8) || (day <= 23 && month == 9))

classicZodiac = "Дева";

else if ((day >= 24 && day <= 30 && month == 9) || (day <= 23 && month == 10))

classicZodiac = "Весы";

else if ((day >= 24 && day <= 31 && month == 10) || (day <= 22 && month == 11))

classicZodiac = "Скорпион";

else if ((day >= 23 && day <= 30 && month == 11) || (day <= 22 && month == 12))

classicZodiac = "Стрелец";

else if ((day >= 23 && day <= 31 && month == 12) || (day <= 20 && month == 1))

classicZodiac = "Козерог";

else if ((day >= 21 && day <= 31 && month == 1) || (day <= 19 && month == 2))

classicZodiac = "Водолей";

else if ((day >= 20 && day <= 29 && month == 2) || (day <= 20 && month == 3))

classicZodiac = "Рыбы";

return classicZodiac;

}

// Метод для получение информации по японскому календарю

public String GetJapaneseZodiac() {

// Парсим дату рождения пользователя

int year = this.date.getYear();

// 12 - количество знаков зодиака

int zodiacNum = year % 12;

// Переменная для хранения информации по японскому календарю

String japaneseZodiac = null;

switch (Math.abs(zodiacNum)) {

case 0:

japaneseZodiac = "Обезьяны";

break;

case 1:

japaneseZodiac = "Петуха";

break;

case 2:

japaneseZodiac = "Собаки";

break;

case 3:

japaneseZodiac = "Свиньи";

break;

case 4:

japaneseZodiac = "Крысы";

break;

case 5:

japaneseZodiac = "Быка";

break;

case 6:

japaneseZodiac = "Тигра";

break;

case 7:

japaneseZodiac = "Кролика";

break;

case 8:

japaneseZodiac = "Дракона";

break;

case 9:

japaneseZodiac = "Змеи";

break;

case 10:

japaneseZodiac = "Лошади";

break;

case 11:

japaneseZodiac = "Козы";

break;

}

return japaneseZodiac;

}

}

1. **Пакет exercise2.nikomu (представляет решение задания №2) c классами:**

* **Main2**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2. Программа запрашивает шестизначное число, после ввода определяет

будет ли являтся счастливым билет с таким номером (сумма первых трех

цифр совпадает с сумой трех последних).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

package exercise2.nikomu;

import java.util.Scanner;

import java.util.regex.Pattern;

public class Main2 {

// Функция для проверки является ли номер билета счастливым

public static String CheckTicketNumber(int number) {

// Переменные для хранения левой и правой частей номера билета

int left = 0, right = 0;

// Делим номер билета на две части - правую и левую (по 3 цифры в каждой)

for(int i = 0; i < 6; i++) {

// Создаем правую часть номера билета

if(i < 3)

right += number % 10;

// Создаем левую часть номера билета

else

left += number % 10;

number /= 10;

}

// Если полученная левая и правая части билета равны, то билет счастливый

if(left == right)

return "счасливый";

// Иначе - несчастливый

else

return "обычный";

}

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Введите шестизначное число номера билета для проверки: ");

String stringTicketNumber = input.nextLine();

// В цикле проверяем введеные пользователем даные и если пользователь ввел не

// шестизначное число, то:

while (!Pattern.compile("\\d{6}").matcher(stringTicketNumber).matches()) {

// Выводим сообщение об ошибке

System.out.println("Ошибка ввода!!! Укажите шестизначное число.\n");

// Запрашиваем повторный ввод

System.out.print("Введите шестизначное число номера билета для проверки: ");

stringTicketNumber = input.nextLine();

}

// Переменная для хранения шестизначного номера билета

int number = Integer.parseInt(stringTicketNumber);

System.out.printf("Ваш билет - %s.\n", CheckTicketNumber(number));

}

}

1. **Пакет exercise3.nikomu (представляет решение задания №3) c классами:**

* **Main3**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

3. Необходимо нарисовать ёлочку символом звёздочки. Каждый

новый ярус должен быть шире предыдущего. С клавиатуры вводится

количество ярусов, и высота первого (верхнего) яруса ёлочки.

Ёлочка должна быть симметричная.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

package exercise3.nikomu;

import java.util.Scanner;

public class Main3 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Укажите количество ярусов ёлочки: ");

// Ввод количества ярусов ёлочки, проверка на ввод числа. Если введено не число,

// то выводим предупреждение об ошибке и повторно запрашиваем от пользователя ввод данных

while (!input.hasNextInt()) {

System.out.println("Ошибка ввода!!! Укажите целое число.\n");

System.out.print("Укажите количество ярусов ёлочки: ");

input.nextLine();

}

int tierCount = input.nextInt();

System.out.print("Укажите высоту первого (верхнего) яруса ёлочки: ");

// Ввод высоты первого (верхнего) яруса ёлочки, проверка на ввод числа. Если введено не число,

// то выводим предупреждение об ошибке и повторно запрашиваем от пользователя ввод данных

while (!input.hasNextInt()) {

System.out.println("Ошибка ввода!!! Укажите целое число.\n");

System.out.print("Укажите высоту первого (верхнего) яруса ёлочки: ");

input.nextLine();

}

int tierHeight = input.nextInt();

// Создаем экземпляр нашего класса TreeBuilder и передаем в его

// конструктор количество ярусов ёлочки и высоту её первого (верхнего) яруса

TreeBuilder tree = new TreeBuilder(tierCount, tierHeight);

tree.ShowTree();

System.out.println("\n");

}

}

* **TreeBuilder**

package exercise3.nikomu;

// Класс для представляющий сущность ёлочка

public class TreeBuilder {

int height; // Поле (переменная класса) для хранения высоты верхнего яруса

int count; // Поле для хранения количества ярусов

// Конструктор

public TreeBuilder(int count, int height) {

this.height = height;

this.count = count;

}

// Метод для отображения ёлочки

public void ShowTree() {

// Переменная для хранения значения максимальной ширины ёлочки

int treeWidth = 2 \* (this.height + this.count - 2) + 1;

// Цикл для создания всех ярусов ёлочки

for (int i = 0; i < this.count; i++) {

// Цикл для создания каждого яруса (отвечает за строки)

for (int j = 0; j < this.height + i; j++) {

// Цикл для создания столбцов в строке

for (int k = 0; k < treeWidth; k++) {

// Переменная для хранения "координаты" середины (оси симметрии) ёлочки

int middle = treeWidth / 2;

// Прорисовка ёлочки

// Если текущая "координата" представляет контур елки, то отображаем "\*" + "пробел"

if (k >= middle - j && k <= middle + j) {

System.out.print("\*" + " ");

}

// Иначе - 2 х "пробел"

else {

System.out.print(" ");

}

}

System.out.println();

}

}

}

}

1. **Пакет exercise4.nikomu (представляет решение задания №4) c классами:**

* **Main4**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

4. Показать на консоли все числа Армстронга в диапазоне

от 0 до 10.000.000. Число Армстронга, или Самовлюблённое

число, или совершенный цифровой инвариант или — натуральное

число, которое равно сумме своих цифр, возведённых в

степень, равную количеству его цифр.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

package exercise4.nikomu;

public class Main4 {

public static void main(String[] args) {

// Переменные для ханения нижнего и верхнего диапазона

final int START = 0, END = 10000000;

// Создаем экземпляр нашего класса ArmstrongNumbers

ArmstrongNumbers armstrongNums = new ArmstrongNumbers(START, END);

armstrongNums.ShowNumbers();

System.out.println("\n");

}

}

* **ArmstrongNumbers**

package exercise4.nikomu;

// Класс который представляет сущность - числа Армстронга в указанном диапазоне

public class ArmstrongNumbers {

private int start; // Поле для хранения начала диапазона

private int end; // Поле для хранения конца диапазона

// Конструктор

public ArmstrongNumbers(int start, int end) {

this.start = start;

this.end = end;

}

// Метод для нахождения суммы всех цифр возведенных в степень (количество цифр в числе), составляющих число

private int Sum(int number) {

// Переменная для хранения суммы цифр, возведенных в степень

int sum = 0;

// Переменная для хранения значения степени

int power = Integer.toString(number).length();

// Переменная для хранения значений цифр

int digit = 0;

do {

digit = number % 10;

number /= 10;

sum += Math.pow(digit, power);

} while(number > 0);

return sum;

}

// Метод для отображения чисел Армстронга

public void ShowNumbers() {

System.out.println("числа Армстронга");

for(int i = this.start, count = 0; i <= this.end; i++) {

if(i == Sum(i))

System.out.println(++count + "-е число = " + i);

}

}

}

1. **Пакет exercise5.nikomu (представляет решение задания №5) c классами:**

* **Main5**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

5. Сформируйте массив из 10 элементов, заполнив его целыми случайными

числами от –10 до 10. Определите среднее арифметическое элементов

массива. Сформируйте новый массив, переписав в него элементы

исходного массива, меньшие среднего. Выполните сортировку исходного

массива по возрастанию элементов.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

package exercise5.nikomu;

import java.util.Arrays;

public class Main5 {

public static void main(String[] args) {

// Начальный массив

int[] array1 = new int[10];

for(int i = 0; i < array1.length; i++) {

array1[i] = (int)(Math.random() \* (20 + 1)) - 10;

}

float arithmeticMean = ArithmeticMean(array1);

// Новый массив

int[] array2 = new int[NewArraySize(array1, arithmeticMean)];

for(int i = 0, count = 0; i < array1.length; i++) {

if(array1[i] < arithmeticMean)

array2[count++] = array1[i];

}

System.out.println("Начальный массив = " + Arrays.toString(array1));

System.out.println("Среднее арифметическое: " + arithmeticMean);

System.out.println("Новый массив (до сортировки) = " + Arrays.toString(array2));

Arrays.sort(array2);

System.out.println("Новый массив (после сортировки) = " + Arrays.toString(array2));

}

// Метод для нахождения среднего арифметического

public static float ArithmeticMean(int[] arr) {

int result = 0;

for(int item : arr) {

result += item;

}

return result / (float)arr.length;

}

// Метод для нахождения размера нового массива

public static int NewArraySize(int[] arr, float arithmeticMean) {

int count = 0;

for(int item : arr) {

if(item < arithmeticMean)

count++;

}

return count;

}

}

1. **Пакет exercise6.nikomu (представляет решение задания №6) c классами:**

* **Main6**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

6. Сформируйте двумерный целочисленный массив. Число строк и

столбцов нужно запросить у пользователя. Заполните массив

случайными числами от 0 до 10 и выведите его на консоль.

Удалите из массива одну строку (номер строки запросите у

пользователя). Распечатайте полученный массив. Используйте

статические методы для реализации каждого действия

(инициализация массива, вывод на консоль, удаление строки).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

package exercise6.nikomu;

import java.util.\*;

public class Main6 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Введите количество строк массива: ");

while (!input.hasNextInt()) {

System.out.println("Ошибка ввода!!! Укажите целое число.\n");

System.out.print("Введите количество строк массива: ");

input.nextLine();

}

// Переменная для хранения количества строк

int rowCount = input.nextInt();

System.out.print("Введите количество столбцов массива: ");

while (!input.hasNextInt()) {

System.out.println("Ошибка ввода!!! Укажите целое число.\n");

System.out.print("Введите количество столбцов массива: ");

input.nextLine();

}

// Переменная для хранения количества столбцов

int columnCount = input.nextInt();

// Начальный массив

int[][] array1 = (CreateArray(rowCount, columnCount));

System.out.println("Массив до удаления строки: \n");

ShowArray(array1);

System.out.println();

System.out.print("Введите номер строки массива, которую необходимо удалить: ");

while (!input.hasNextInt()) {

System.out.println("Ошибка ввода!!! Укажите целое число.\n");

System.out.print("Введите номер строки массива, которую необходимо удалить: ");

input.nextLine();

}

// Переменная для хранения строки, которую будем удалять

int deleteRow = input.nextInt();

// Массив после удаления строки

int[][] array2 = DeleteRow(array1, deleteRow - 1);

System.out.println("Массив после удаления строки: \n");

ShowArray(array2);

System.out.println();

}

// Метод для создания массива

static int[][] CreateArray(int rows, int columns) {

// Временный массив

int[][] tempArr = new int[rows][columns];

for(int i = 0; i < rows; i++) {

for(int j = 0; j < columns; j++) {

tempArr[i][j] = (int)(Math.random() \* 10);

}

}

return tempArr;

}

// Метод для отображения массива

static void ShowArray(int[][] arr) {

for(int[] rowItem : arr) {

System.out.println(Arrays.toString(rowItem));

}

}

// Метод для удаления строки (создание нового массива, исключающего указанную строку)

static int[][] DeleteRow(int[][] arr, int rowNumber) {

int rowSize = arr[0].length;

int columnSize = arr.length ;

int[][] tempArr = new int[rowSize - 1][columnSize];

for(int i = 0, count = 0; i < rowSize; i++) {

// Если текущая строка является той, которую необходимо удалить из исходного массива,

// то пропускаем итерацию и не копируем её в новый массив

if(i == rowNumber)

continue;

for(int j = 0; j < columnSize; j++) {

tempArr[count][j] = arr[i][j];

}

count++;

}

return tempArr;

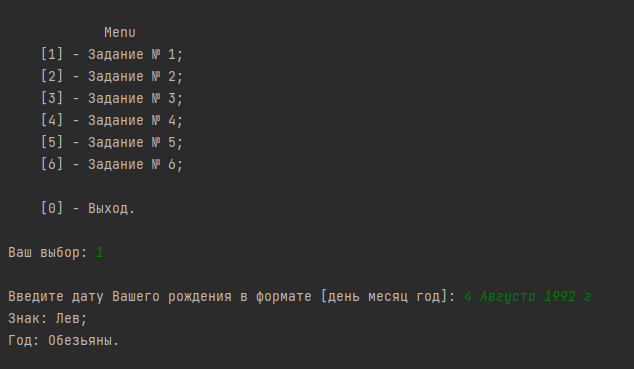
}

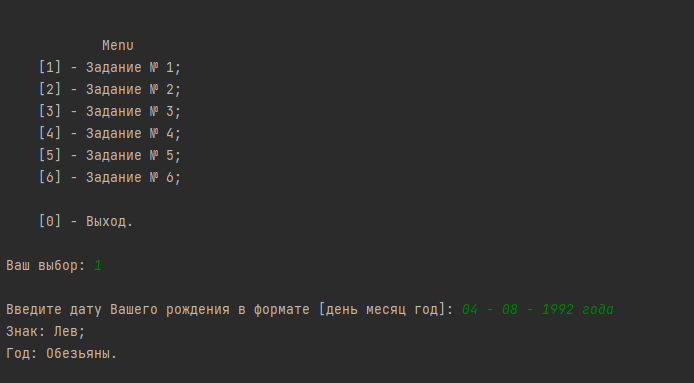
}

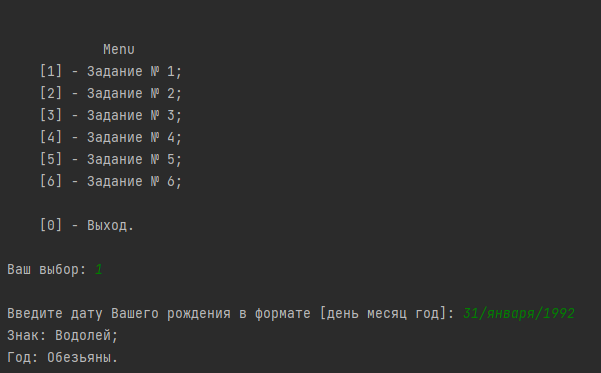
**Работа приложения:**

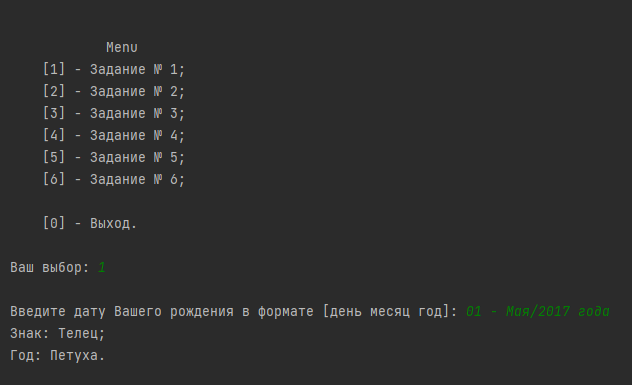


*Рис.2 – Общий главного меню проекта.*

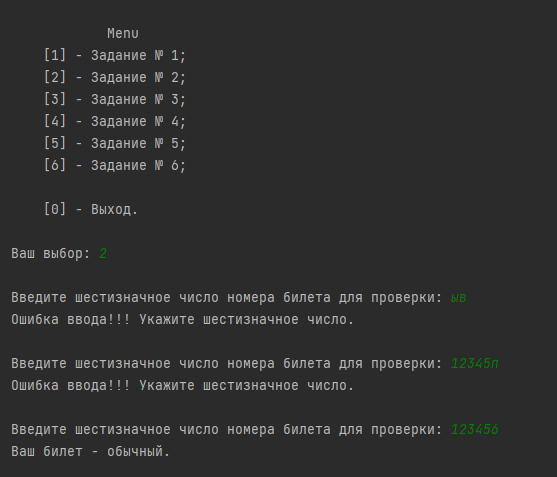


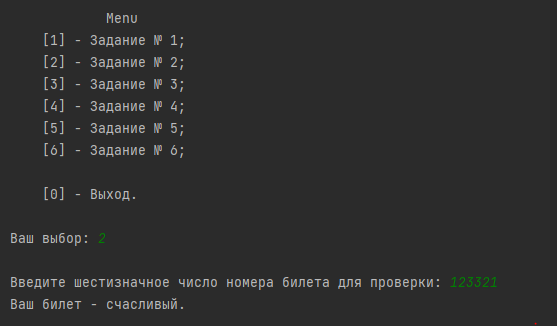


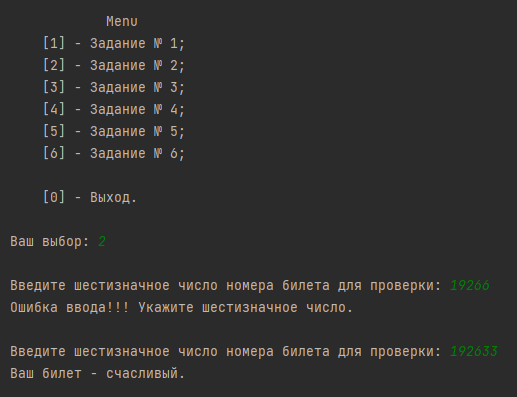




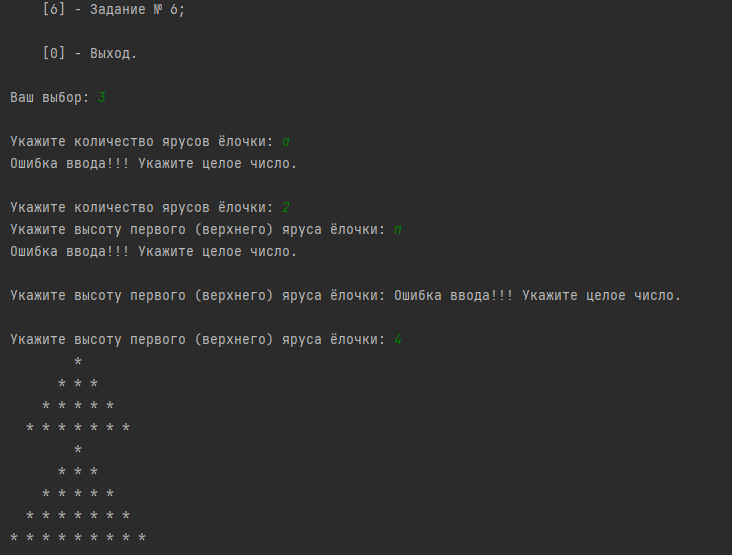
*Рис.3-6 – Работа программы с решением задания №1.*

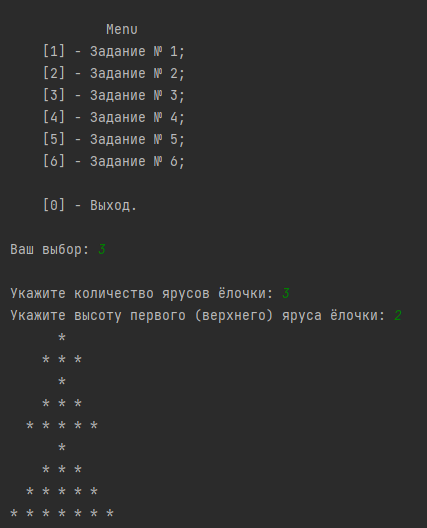


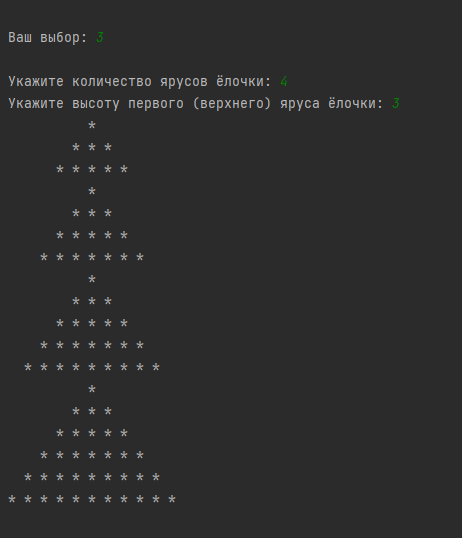




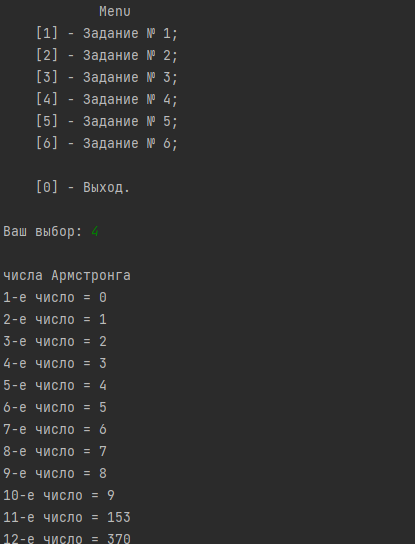
*Рис.7-9 – Работа программы с решением задания №2.*

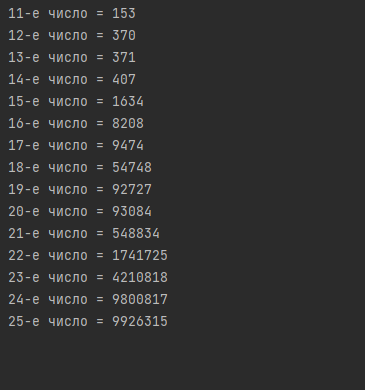




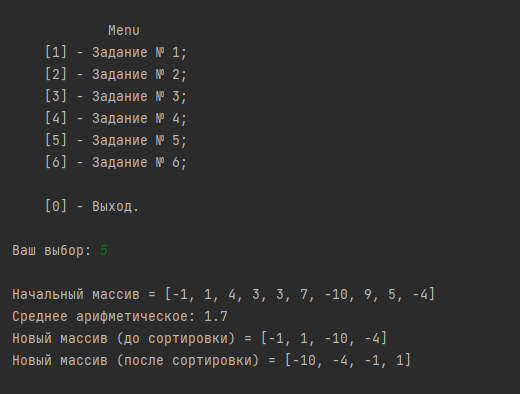


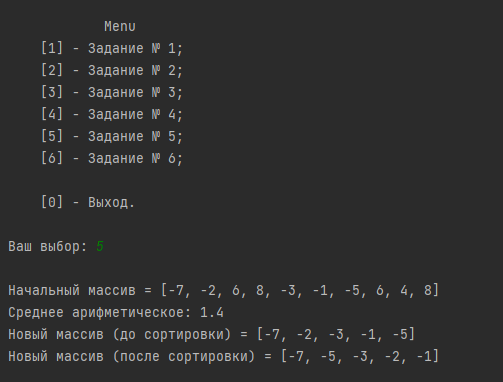
*Рис.10-12 – Работа программы с решением задания №3.*

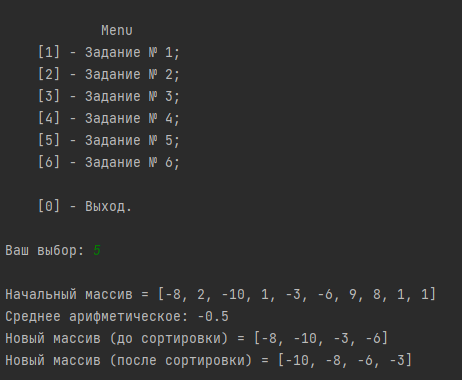




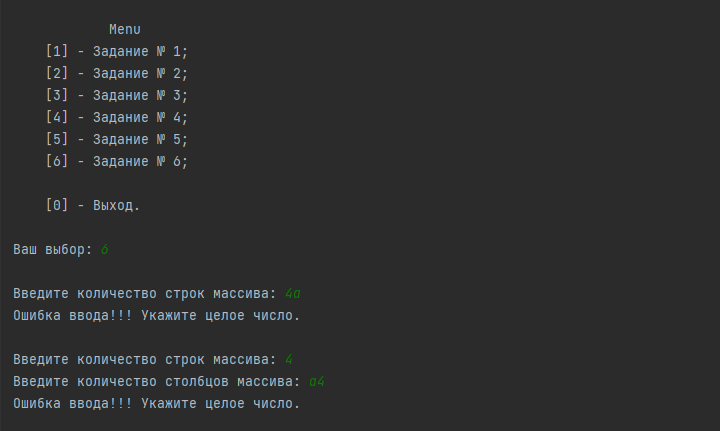
*Рис.13-14 – Работа программы с решением задания №4.*

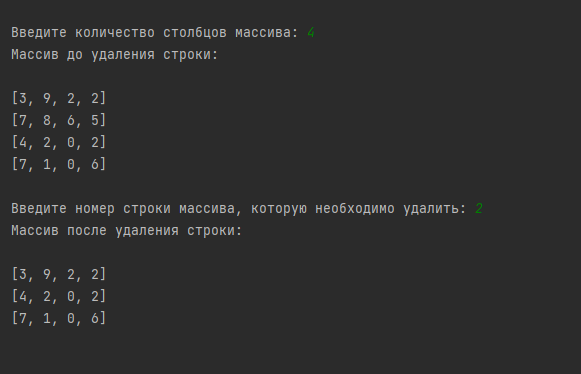


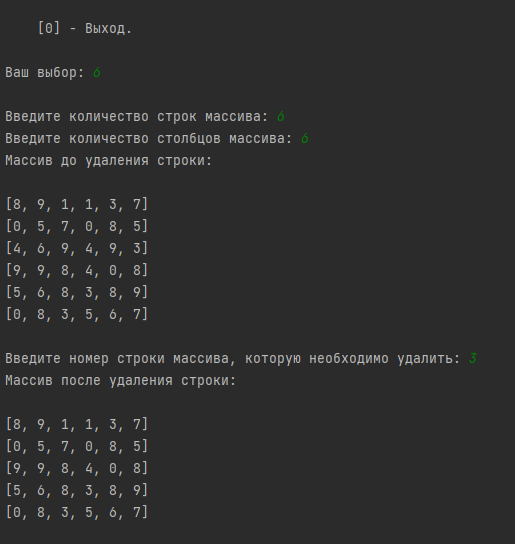




*Рис.15-17 – Работа программы с решением задания №5.*







*Рис.18-20 – Работа программы с решением задания №6.*