**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

Факультет инфокоммуникационных сетей и систем (ИКСС)

Лабораторная работа №3

по курсу

«Разработка Java-приложений управления телекоммуникациями»

Группа: ИКПИ-11

Выполнил студент: Хохлов Т. В.

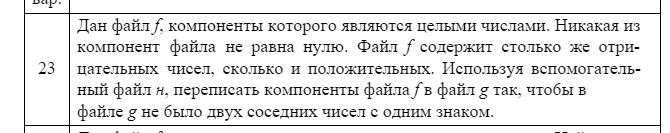
Принял преподаватель: Березин А.Ю.

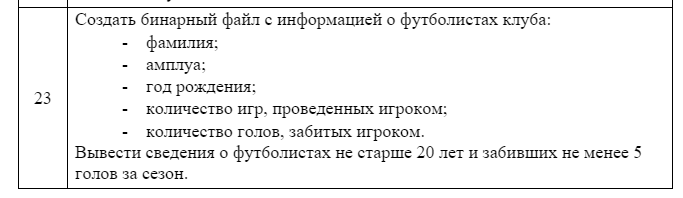
Место для подписи \_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2023 г.

# **Цель работы**





**Описание кода**

Функции**:**

**void menuOutput –** вывод меню выбора действий.

**void fileOutput –** вывод небинарного файла на экран.

**void footballFileOutput –** выводбинарного файла на экран.

**void loadingScreen –** вывод на экран анимированной загрузки…

**void removeFiles –** вспомогательная функция удаления созданных программой файлов.

**void task1/task2 –** функции выполняющие соответствующие задания.

**void task1/task2 –** вспомогательная функция для просмотра содержимого файлов с терминала.

Класс **football** был создан для более удобной и наглядной работы с бинарным файлом (задание 2).

# **Листинг**

## **Lab3.java**

package Java.lab3;

import java.io.BufferedInputStream;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.util.Date;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Scanner;

import java.util.concurrent.TimeUnit;

public class lab3 {

    public static void main(String[] args) throws InterruptedException, ParseException {

        Scanner systemIn = new Scanner(System.in);

        String pathF = "f.txt";

        String pathN = "n.txt";

        String pathG = "g.txt";

        String pathFootball = "football.txt";

        String pathLess20 = "less20.txt";

        while(true) {

            menuOutput();

            int choice = getInt(systemIn);

            switch (choice) {

                case 1:

                System.out.printf("\033[2J");

                System.out.println("Вывод файла F");

                fileOutput(pathF, systemIn);

                    break;

                case 2:

                System.out.printf("\033[2J");

                System.out.println("Вывод файла N");

                fileOutput(pathN, systemIn);

                    break;

                case 3:

                System.out.printf("\033[2J");

                System.out.println("Вывод файла G");

                fileOutput(pathG, systemIn);

                    break;

                case 4:

                System.out.printf("\033[2J");

                System.out.println("Вывод файла Football.txt");

                footballFileOutput(pathFootball, systemIn);

                    break;

                case 5:

                System.out.printf("\033[2J");

                System.out.println("Вывод файла less20.txt");

                footballFileOutput(pathLess20, systemIn);

                    break;

                case 6:

                loadingScreen();

                task1(pathF, pathN, pathG);

                    break;

                case 7:

                loadingScreen();

                task2(pathFootball, pathLess20);

                    break;

                case 8:

                removeFiles(pathN, pathG, pathLess20);

                    break;

                case 0:

                System.out.printf("Работа завершена!");

                    return;

                default:

                System.out.printf("\033[2J");

                System.out.println("Выберите существующее действие!");

                TimeUnit.SECONDS.sleep(1);

                    break;

            }

        }

    }

    static void menuOutput() {

        System.out.printf("\033[2J");

        System.out.println("================Выберите действие================");

        System.out.println("1. Вывести на экран файл F");

        System.out.println("2. Вывести на экран файл N");

        System.out.println("3. Вывести на экран файл G");

        System.out.println("4. Вывести на экран файл Football.txt");

        System.out.println("5. Вывести на экран файл less20.txt");

        System.out.println("6. Выполнить задание 1");

        System.out.println("7. Выполнить задание 2");

        System.out.println("8. Удалить вспомогательные файлы и файлы вывода");

        System.out.println("0. Выход");

    }

    static void fileOutput(String path, Scanner systemIn) throws InterruptedException {

        int[] list = null;

        try (BufferedReader file = new BufferedReader(new FileReader(path))){

            list = file.lines().mapToInt(Integer::parseInt).toArray();

            for (var i: list) {

                System.out.print(i + " ");

            }

            pressEnterToContinue(systemIn);

        } catch (FileNotFoundException e) {

            System.out.println("Данного файла не существует.");

            TimeUnit.SECONDS.sleep(1);

            return;

        } catch (IOException e) {

            e.getMessage();

        }

    }

    static void footballFileOutput(String path, Scanner systemIn) throws InterruptedException {

        try (BufferedReader br = new BufferedReader(new FileReader(path))) {

            String line;

            while ((line = br.readLine()) != null) {

                System.out.println(line);

            }

            pressEnterToContinue(systemIn);

        } catch (FileNotFoundException e) {

            System.out.println("Данного файла не существует.");

            TimeUnit.SECONDS.sleep(1);

            return;

        } catch (IOException e) {

            e.getMessage();

        }

    }

    static void task1(String pathF, String pathN, String pathG) throws InterruptedException {

        int[] list = null;

        try (BufferedReader file = new BufferedReader(new FileReader(pathF))){

            list = file.lines().mapToInt(Integer::parseInt).toArray();

        } catch (FileNotFoundException e) {

            System.out.println("Файла F не существует! Проверьте задание!");

            TimeUnit.SECONDS.sleep(1);

            return;

        } catch (IOException e) {

            e.getMessage();

        }

        try (BufferedWriter out = new BufferedWriter(new FileWriter(pathN))) {

            for (var i: list) {

                if (i > 0) {

                out.append(String.valueOf(i));

                out.append("\n");

                }

            }

        } catch (IOException e) {

            e.getMessage();

        }

        int[] listPositive = null;

        try (BufferedReader file = new BufferedReader(new FileReader(pathN))){

            listPositive = file.lines().mapToInt(Integer::parseInt).toArray();

        } catch (IOException e) {

            e.getMessage();

        }

        try (BufferedWriter out = new BufferedWriter(new FileWriter(pathG))) {

            int count = 0;

            for (var i: list) {

                if (i < 0) {

                out.append(String.valueOf(i));

                out.append("\n");

                out.append(String.valueOf(listPositive[count]));

                out.append("\n");

                count++;

                }

            }

        } catch (IOException e) {

            e.getMessage();

        }

    }

    static void task2 (String pathFootball, String pathLess20) throws InterruptedException {

        ArrayList<football> footballList = new ArrayList<football>();

        try (DataInputStream file = new DataInputStream(new BufferedInputStream(new FileInputStream(pathFootball)))) {

            byte[] buffer = file.readAllBytes();

            String str = new String(buffer);

            String lines[] = str.split("\\r?\\n");

            String copy[] = new String[lines.length];

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

                copy[i] = lines[i];

            while (lines[0]!=null) {

                Date d = new SimpleDateFormat("dd/MM/yyyy").parse(lines[2]);

                football kicker = new football(lines[0], lines[1], d, Integer.parseInt(lines[3]), Integer.parseInt(lines[4]));

                footballList.add(kicker);

                for (int i = 0; i < 5; ++i) {

                    lines[i] = null;

                }

                while (lines[0] == null) {

                    for (int i = 1; i < lines.length; ++i) {

                        lines[i-1] = lines[i];

                    }

                }

                // for (football i: footballList) {

                //     i.printAll();

                // }

                DataOutputStream out = new DataOutputStream(new FileOutputStream(pathLess20));

                String newLine = System.getProperty("line.separator");

                for (football i: footballList) {

                    if (i.getAge() < 20) {

                        out.write(i.getSurname().getBytes());

                        out.writeBytes(newLine);

                        out.write(i.getPos().getBytes());

                        out.writeBytes(newLine);

                        out.write(i.getBirthDate().getBytes());

                        out.writeBytes(newLine);

                        out.write(i.getGamesPlayed().getBytes());

                        out.writeBytes(newLine);

                        out.write(i.getGoalsMade().getBytes());

                        out.writeBytes(newLine);

                    }

                }

                out.close();

            }

        } catch (FileNotFoundException e) {

            System.out.println("Файла football.txt не существует! Проверьте задание!");

            TimeUnit.SECONDS.sleep(1);

            return;

        } catch (IOException e) {

            e.getMessage();

        } catch (NullPointerException e) {

            e.getMessage();

        } catch (ParseException e) {

            e.getMessage();

        }

    }

    static void removeFiles(String pathN, String pathG, String pathLess20) throws InterruptedException {

        System.out.printf("\033[2J");

        File N = new File(pathN);

        File G = new File(pathG);

        File Less20 = new File(pathLess20);

        if(!N.exists()&&!G.exists()&&!Less20.exists()){System.out.println("Файлов нет. Удалять нечего..."); TimeUnit.SECONDS.sleep(1); return;}

        if (N.exists()) {N.delete(); System.out.println("Файл N удален!");}

        if (G.exists()) {G.delete(); System.out.println("Файл G удален!");}

        if (Less20.exists()) {Less20.delete(); System.out.println("Файл less20.txt удален!");}

        TimeUnit.SECONDS.sleep(1);

    }

    static int getInt(Scanner systemIn) throws InterruptedException {

        if (systemIn.hasNextInt()) {

            int nextInt = systemIn.nextInt();

            systemIn.nextLine();

            return nextInt;

        } else {

            System.out.printf("\033[2J");

            System.out.println("Ошибка ввода! Введите число.");

            TimeUnit.SECONDS.sleep(1);

            systemIn.nextLine();

        }

        return 7;

    }

    static void pressEnterToContinue(Scanner systemIn)

    {

        System.out.println("\nНажмите Enter чтобы продолжить...");

        try {

            System.in.read();

            systemIn.nextLine();

        } catch(Exception e) {}

    }

    static void loadingScreen() throws InterruptedException {

        System.out.printf("\033[2J");

        for (int i = 0; i < 6; i++) {

            System.out.print("\\ Загрузка \\\r");

            TimeUnit.MILLISECONDS.sleep(100);

            System.out.print("| Загрузка |\r");

            TimeUnit.MILLISECONDS.sleep(100);

            System.out.print("/ Загрузка /\r");

            TimeUnit.MILLISECONDS.sleep(100);

            System.out.print("- Загрузка -\r");

            TimeUnit.MILLISECONDS.sleep(100);

        }

    }

}

## **football.java**

package Java.lab3;

import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

public class football {

    String surname;

    String pos;

    Date birthDate;

    int gamesPlayed;

    int goalsMade;

    football() {

        this.surname = "";

        this.pos = "";

        this.birthDate = null;

        this.gamesPlayed = 0;

        this.goalsMade = 0;

    }

    football(String surname, String pos, Date birthDate, int gamesPlayed, int goalsMade) {

        this.surname = surname;

        this.pos = pos;

        this.birthDate = birthDate;

        this.gamesPlayed = gamesPlayed;

        this.goalsMade = goalsMade;

    }

    void printAll() {

        System.out.println(this.surname);

        System.out.println(this.pos);

        System.out.println(this.birthDate);

        System.out.println(this.gamesPlayed);

        System.out.println(this.goalsMade);

    }

    String getSurname() {

        return this.surname;

    }

    String getPos () {

        return this.pos;

    }

    String getBirthDate() {

        return this.birthDate.toString();

    }

    String getGamesPlayed() {

        String gamesPlayedCopy = this.gamesPlayed + "";

        return gamesPlayedCopy;

    }

    String getGoalsMade() {

        String goalsMadeCopy = this.goalsMade + "";

        return goalsMadeCopy;

    }

    int getAge () {

        int year = this.birthDate.getYear() + 1900;

        int age = 2023 - year;

        return age;

    }

}

# **Содержимое входных файлов**

**f.txt:**

**-3243**

**-2**

**3**

**4**

**-6**

**-3**

**5**

**1**

**Football.txt:**

**Дунаев**

**Нападающий**

**31/01/2014**

**12**

**21**

**Хохлов**

**Защитник**

**12/07/2003**

**2**

**3**

# **Выводы**

В результате выполнения лабораторной работы была написана программа на языке Java, которая включает в себя несколько задач; изучен принцип работы классов для работы с файлами.