МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Кафедра ИИТ

# Отчет по лабораторной работе №3

Выполнила:
Студентка группы
АС-50
Дряпко А. В.
Проверил:
Крощенко А.А.

Брест 2020

**Вариант 4**

**Задание 1.**

Реализовать простой класс.

Требования к выполнению

• Реализовать пользовательский класс по варианту.

• Создать другой класс с методом main, в котором будут находится примеры использования
пользовательского класса.

Для каждого класса

• Создать поля классов

• Создать методы классов

• Добавьте необходимые get и set методы (по необходимости)

• Укажите соответствующие модификаторы видимости

• Добавьте конструкторы

• Переопределить методы toString() и equals()

4) Прямоугольник, заданный длинами двух сторон – Предусмотреть возможность определения площади и периметра, а так же логические методы, определяющие, является ли прямоугольник квадратом и существует ли такой прямоугольник. Конструктор должен позволить создавать объекты с начальной инициализацией. Реализовать метод equals, выполняющий сравнение объектов данного типа.

**Код программы:**

**Class Main**

```
public class Main {

    public static void main(String[] args){

        Rectangle first =new Rectangle();
        Rectangle second =new Rectangle(2,2);
        Rectangle third =new Rectangle(-2,4);
        Rectangle fourth =new Rectangle(1,2);
        Rectangle fifth =new Rectangle(3,5);

        System.out.println("Площадь прямоугольника со сторонами "
+first.a+" и " + first.b+ ":");
        System.out.println(first.square());

        System.out.println("Периметр прямоугольника со сторонами "
+first.a+" и " + first.b+ ":");
        System.out.println(first.perimeter());


        System.out.println("Прямоугольник со сторонами " +second.a+" и " +
second.b+ " квадрат?");
        System.out.println(second.isItSquare());
        System.out.println("Прямоугольник со сторонами " +fourth.a+" и " +
fourth.b+ " квадрат?");
        System.out.println(fourth.isItSquare());


        System.out.println("Существует ли  прямоугольник со сторонами
"+third.a+" и " + third.b+ "?");
```

```
        System.out.println(third.checkForExistence());
        System.out.println("Существует ли  прямоугольник со сторонами
"+fourth.a+" и " + fourth.b+ "?");
        System.out.println(fourth.checkForExistence());


        System.out.println("Первый прямоугольник со сторонами " +first.a+"
и " + first.b+
                " равен второму прямоугольнику со сторонами "+second.a+" и
" + second.b);
        System.out.println(first.equals(second));
        System.out.println("Первый прямоугольник со сторонами " +first.a+"
и " + first.b+
                " равен второму прямоугольнику со сторонами "+fifth.a+" и "
+ fifth.b);
        System.out.println(first.equals(fifth));

    }

}
```

**Class Rectangle**

```
import java.util.Objects;

public class Rectangle {

    double a, b;

    public Rectangle(){
        this.a=3;
        this.b=5;
    }

    public Rectangle(double a, double b){
        this.a=a;
        this.b=b;

    }
    public double perimeter(){
        return 2*a+2*b;
    }
    public double square(){
        return a*b;
    }
    public boolean isItSquare(){
        if(a==b) return true;
        return false;
    }
    public boolean checkForExistence(){
        if(a<=0 || b<=0) return false;
        return true;
    }

    @Override
```

```
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Rectangle rectangle = (Rectangle) o;
        return Double.compare(rectangle.a, a) == 0 &&
                Double.compare(rectangle.b, b) == 0;
    }

    @Override
    public String toString() {
        return "Rectangle{" +
                "a=" + a +
                ", b=" + b +
                '}';
    }
}
```

```
Площадь прямоугольника со сторонами 3.0 и 5.0:
15.0
Периметр прямоугольника со сторонами 3.0 и 5.0:
16.0
Прямоугольник со сторонами 2.0 и 2.0 квадрат?
true
Прямоугольник со сторонами 1.0 и 2.0 квадрат?
false
Существует ли  прямоугольник со сторонами -2.0 и 4.0?
false
Существует ли  прямоугольник со сторонами 1.0 и 2.0?
true
Первый прямоугольник со сторонами 3.0 и 5.0 равен второму прямоугольнику со сторонами 2.0 и 2.0
false
Первый прямоугольник со сторонами 3.0 и 5.0 равен второму прямоугольнику со сторонами 3.0 и 5.0
true
```

**Задание 2.** Разработать автоматизированную систему на основе некоторой структуры данных, манипулирующей объектами пользовательского класса. Реализовать требуемые функции обработки данных

Требования к выполнению

• Задание посвящено написанию классов, решающих определенную задачу автоматизации;

• Данные для программы загружаются из файла (формат произволен). Файл создать и написать
вручную.

4) Автоматизированная система в библиотеке. Составить программу, которая содержит текущую информацию о книгах в библиотеке. Сведения о книгах (Book) содержат:

• номер УДК;
• Фамилию и инициалы автора;
• Название;
• Год издания;
• Количество экземпляров в библиотеке;
• Количество страниц;
• Количество томов;
• ФИО читателя, взявшего книгу (при наличии);
• Срок сдачи книги (если была взята).

Программа должна обеспечивать:

• Формирование общего списка книг;
• Формирование списка книг, старше n лет;
• Формирование списка книг, взятых на чтение;
• Формирование списка книг, взятых на чтение с выводом личной информации о читателях;
• Формирование списка книг, которые задержаны читателем дольше указанного срока.

**Код программы:**
**Class Main**

```
import java.io.*;
import java.time.LocalDate;
import java.util.ArrayList;

public class Main {

    private static ArrayList<Library>list1, list2,list3, list4, list5;

    public static void main(String[] args){
        File file = new File("boooks.txt");
        list1 = new ArrayList<>();
        list2 = new ArrayList<>();
        list3 = new ArrayList<>();
        list4 = new ArrayList<>();
        list5 = new ArrayList<>();
        System.out.println("Список книг");
        fillListBook(list1,file);
        outInfoAboutBook(list1);
        System.out.println("Книги старше 10 лет");
        fillListBookOld(list2,file,10);
        outInfoAboutBook(list2);
        System.out.println("Книги, взятые на чтение");
        fillListSomebodyRead(list3, file);
        outInfoAboutBook(list3);
        System.out.println("Книги, взятые на чтение, с выводом личной
информации");
        fillListSomebodyReadWithInfo(list4, file);
        outInfoAllInfo(list4);
        System.out.println("Книги, которые задержаны читетелем дольше
указанного срока");
        LocalDate dateNow=  LocalDate.parse("2020-05-05");
        fillListOverdue(list5, file, dateNow);
        outInfoAllInfo(list5);
    }



    public static void outInfoAboutBook(ArrayList<Library> list){
        for(Library library:list){
            System.out.println(library.getBook());
        }
    }
```

```
    public static void outInfoAllInfo(ArrayList<Library> list){
        for(Library library:list){
            System.out.println(library.getAllInfo());
        }
    }
    public static void fillListBook(ArrayList<Library> list, File file){
        try {
            FileReader fr = new FileReader(file);
            BufferedReader reader = new BufferedReader(fr);
            String line = reader.readLine();
            Library book;
            while(line!=null){
                String[] words = line.split("\t");
                book = new
Library(words[0],words[1],words[2],words[3],words[4],
                        words[5],words[6]);
                list.add(book);
                line = reader.readLine();
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }


    public static void fillListBookOld(ArrayList<Library> list, File file,
int old){
        try {
            FileReader fr = new FileReader(file);
            BufferedReader reader = new BufferedReader(fr);
            String line = reader.readLine();
            Library book;
            while(line!=null){
                String[] words = line.split("\t");
                int year = 2020 - Integer.parseInt(words[3]);
                if(year>old) {
                    book = new Library(words[0], words[1], words[2],
words[3], words[4],
                            words[5], words[6]);
                    list.add(book);
                }
                    line = reader.readLine();

            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

```
    public static void fillListSomebodyRead (ArrayList<Library> list, File
file){
        try {
            FileReader fr = new FileReader(file);
            BufferedReader reader = new BufferedReader(fr);
            String line = reader.readLine();
            Library book;
            while(line!=null){
                String[] words = line.split("\t");
                if(!words[7].equals("-")) {
                    book = new Library(words[0], words[1], words[2],
words[3], words[4],
                            words[5], words[6]);
                    list.add(book);
                }
                line = reader.readLine();

            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }



    public static void fillListSomebodyReadWithInfo(ArrayList<Library>
list, File file){
        try {

            FileReader fr = new FileReader(file);
            BufferedReader reader = new BufferedReader(fr);
            String line = reader.readLine();
            Library book;
            while(line!=null){
                String[] words = line.split("\t");
                if(!words[7].equals("-")) {
                    book = new Library(words[0], words[1], words[2],
words[3], words[4],
                            words[5], words[6], words[7], words[8]);
                    list.add(book);
                }
                line = reader.readLine();

            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

```
    public static void fillListOverdue(ArrayList<Library> list, File file,
LocalDate dateNow){
        try {
            FileReader fr = new FileReader(file);
            BufferedReader reader = new BufferedReader(fr);
            String line = reader.readLine();
            Library book;
            while(line!=null){
                String[] words = line.split("\t");
                if(!words[7].equals("-")) {
                    LocalDate dateSdachi = LocalDate.parse(words[8]);

                    if ((dateNow.compareTo(dateSdachi))>0) {
                        book = new Library(words[0], words[1], words[2],
words[3], words[4],
                                words[5], words[6], words[7], words[8]);
                        list.add(book);
                    }
                    line = reader.readLine();
                }
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

}
```

**Class Library**

```
public class Library {
    private String numberUdk;
    private String SurnameInit;
    private String title;
    private String year;
    private String numberExampleInLibrary;
    private String numberPages;
    private String numberVolume;
    private String fio;
    private String timeToPass;

    public Library(String numberUdk, String surnameInit, String title,
String year, String numberExsampleInLibrary, String numberPages, String
numberVolume) {
        this.numberUdk = numberUdk;
        SurnameInit = surnameInit;
        this.title = title;
        this.year = year;
        this.numberExampleInLibrary = numberExsampleInLibrary;
        this.numberPages = numberPages;
        this.numberVolume = numberVolume;

    }
```

```
    public Library(String numberUdk, String surnameInit, String title,
String year, String numberExampleInLibrary, String numberPages, String
numberVolume, String fio, String timeToPass) {
        this.numberUdk = numberUdk;
        SurnameInit = surnameInit;
        this.title = title;
        this.year = year;
        this.numberExampleInLibrary = numberExampleInLibrary;
        this.numberPages = numberPages;
        this.numberVolume = numberVolume;
        this.fio = fio;
        this.timeToPass = timeToPass;
    }



    public String getBook() {
        return
                "numberUdk='" + numberUdk + '\'' +
                ", SurnameInit='" + SurnameInit + '\'' +
                ", title='" + title + '\'' +
                ", year='" + year + '\'' +
                ", numberExampleInLibrary='" + numberExampleInLibrary +
'\'' +
                ", numberPages='" + numberPages + '\'' +
                ", numberVolume='" + numberVolume + '\'' ;
    }


    public String getAllInfo() {
        return
                "numberUdk='" + numberUdk + '\'' +
                ", SurnameInit='" + SurnameInit + '\'' +
                ", title='" + title + '\'' +
                ", year='" + year + '\'' +
                ", numberExampleInLibrary='" + numberExampleInLibrary +
'\'' +
                ", numberPages='" + numberPages + '\'' +
                ", numberVolume='" + numberVolume + '\'' +
                ", fio='" + fio + '\'' +
                ", timeToPass='" + timeToPass + '\''
                ;
    }

    public String getNumberUdk() {
        return numberUdk;
    }

    public String getSurnameInit() {
        return SurnameInit;
    }

    public String getTitle() {
```

```
        return title;
    }

    public String getYear() {
        return year;
    }

    public String getNumberExampleInLibrary() {
        return numberExampleInLibrary;
    }

    public String getNumberPages() {
        return numberPages;
    }

    public String getNumberVolume() {
        return numberVolume;
    }

    public String getFio() {
        return fio;
    }

    public String getTimeToPass() {
        return timeToPass;
    }
}
```