

Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский государственный технический университет»  
Кафедра ИИТ

**Лабораторная работа №3**

По дисциплине: «Современные платформы программирования»

**Выполнил:**

студент 3 курса  
группы ПО-8  
Таразевич Н.А.

**Проверил:**

Крощенко А.А.

**Цель работы:** научиться создавать и использовать классы в программах на языке программирования Java.

**Задание 1:** реализовать класс прямоугольника, заданного длинами двух сторон. Предусмотреть возможность определения площади и периметра, а также 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:

```
package Task1;

public class Rectangle {

    private double
length; private
double width;

    public Rectangle() {
        this.length =
1; this.width =
1;
    }

    public Rectangle(double length, double width) throws Exception {
        if (length <= 0 || width <= 0)

            throw new Exception("Length and width must be
positive"); this.length = length;

        this.width =
width; }

    public double getLength()
    { return length;
    }

    public double getWidth()
    { return width;
    }

    public void setWidth(double width)
    { if (width >= 0)

        this.width =
width; }

    public void setLength(double length)
    { if (length >= 0)

        this.length =
length; }
```

```

    public double getArea() {
        return length *
            width;
    }

    public double getPerimeter() {
        return 2 * (length + width);
    }


    public boolean isSquare() {
        return length == width;
    }

    @Override
    public boolean equals(Object obj) {
        if (obj == null || getClass() != obj.getClass()) {
            return false;
        }

        Rectangle rect = (Rectangle) obj;

        return Double.compare(rect.length, length) == 0 &&
            Double.compare(rect.width, width) == 0;
    }

    @Override
    public String toString() {
        return "rectangle(" +
            "length = " + length +
            ", width = " + width + ")";
    }
}

```

## Входные данные:

```

package Task1;

public class RectangleTest {

    public static void main(String[] args) throws Exception {
        Rectangle rect1 = new Rectangle();

        Rectangle rect2 = new Rectangle(1, 1);

        if (rect1.equals(rect2))
            System.out.println("Rectangles are equal");
        else
            System.out.println("Rectangles are not equal");

        if (rect1.isSquare())
            System.out.print(rect1 + " is square;" +

```

```

        " its area = " + rect1.getArea() +
        "; its perimeter = " + rect1.getPerimeter());
    }
}

```

### Результат работы программы:

```

D:\SDK\JDK\bin\java.exe "-javaagent:D:\JetBrains\IntelliJ IDEA Community Edition 20
Rectangles are equal
rectangle(length = 1.0, width = 1.0) is square; its area = 1.0; its perimeter = 4.0
Process finished with exit code 0

```

**Задание 2:** реализовать автоматизированную систему 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:

```

package Task2;

import java.util.Date;
import java.util.HashMap;

class Book {
    private String udcCode;
    private String author;
    private String title;
    private int

```

```

    publicationYear; private
    int pageCount; private
    int volumesCount; private
    int copiesCount;

    private HashMap<String, Date> borrowedBooks;

    public Book(String udkNumber, String author, String title, int
        publicationYear, int pageCount, int volumesCount, int
        copiesCount) {

        this.udcCode =
            udkNumber; this.author
            = author; this.title =
            title;

        this.publicationYear =
            publicationYear; this.pageCount =
            pageCount; this.volumesCount =
            volumesCount; this.copiesCount =
            copiesCount; this.borrowedBooks = new
            HashMap<>();
    }

    public String getUdcCode() {
        return udcCode;
    }

    public void setUdcCode(String udcCode) {
        if (udcCode != null && !udcCode.isEmpty()){
            this.udcCode = udcCode;
        }
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        if (author != null && !author.isEmpty()){
            this.author = author;
        }
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        if (title != null && !title.isEmpty()){
            this.title = title;
        }
    }

    public int getPublicationYear() {
        return publicationYear;
    }

    public void setPublicationYear(int publicationYear) { if
        (publicationYear > 0){

```

```

        this.publicationYear = publicationYear; }
    }

    public int getPagesCount() {
        return pagesCount;
    }

    public void setPagesCount(int pagesCount) { if
        (pagesCount > 0){
            this.pagesCount = pagesCount; }
    }

    public int getVolumesCount() {
        return volumesCount;
    }

    public void setVolumesCount(int volumesCount) { if
        (volumesCount > 0){
            this.volumesCount = volumesCount; }
    }

    public int getCopiesCount() {

        return copiesCount;
    }

    public void setCopiesCount(int copiesCount) {
        if (copiesCount > 0){
            this.copiesCount = copiesCount;
        }
    }

    @Override
    public String toString() {
        StringBuilder readers = new StringBuilder();
        if (!borrowedBooks.isEmpty()){
            readers.append(", readers( ");
            borrowedBooks.forEach((readerName, deadline) -> {
                readers.append(readerName).append(",
").append(deadline).append(";");
            });
            readers.append(")");
        }

        return "book(" +
            "udkNumber = " + udcCode +
            ", author = " + author +
            ", title = " + title +
            ", publicationYear = " + publicationYear +
            ", pageCount = " + pageCount +

```

```

        ", volumesCount = " + volumesCount +
        ", copiesCount = " + copiesCount +
        readers + ")\n";
    }

    public Book borrowBook(String reader, Date deadline){
        if (reader != null && deadline != null && copiesCount >= 1){
            copiesCount--;

            borrowedBooks.put(reader, deadline);
            return this;
        }
        else return null;
    }

    public void returnBook(String reader){
        borrowedBooks.remove(reader);
    }

    public HashMap<String, Date> getBorrowedBooks(){
        return borrowedBooks;
    }
}

```

**Вывод:** я научился создавать и использовать классы в программах на языке программирования Java.





**Задание 2:** Утилита `nl` выводит переданный файл в стандартный вывод или в другой файл, выполняя нумерацию его строк. Если файл не задан или задан как `-`, читает стандартный ввод.

Формат использования: `nl [-i] [-l] [-n] входной_файл [выходной_файл]`

- `-i` ЧИСЛО Задаёт шаг увеличения номеров строк
- `-l` 1/0 Задаёт флаг нумерации пустых строк
- `-n` ФОРМАТ Использовать заданный формат для номеров строк.

`ln` – номер выровнен по левому краю, без начальных нулей

`rn` – номер выровнен по правому краю, без начальных нулей

`rz` – номер выровнен по правому краю с начальными нулями

Пример использования: `nl -i 2 -l 0 -n ln in.txt`

Обрабатывает файл `in.txt`, выводит результат в стандартный вывод, инкремент счетчика равен двум (`-i 2`), пустые строки не нумеруются.

### Код программы:

```
import java.io.File;

import java.io.FileOutputStream;
import java.io.PrintStream;
import java.util.*;

public class Task2 {

    private static final int optionPattern = 0;
    private static final int optionValue = 1;

    public static void main(String[] args) throws Exception {
        if (args.length == 0 || !args[0].equals("nl")) {
            System.err.println("Utility name - nl");
            System.exit(1);
        }

        Map<String, ArrayList<String>> options = new HashMap<>();
        options.put("-i", new ArrayList<>(Arrays.asList("^([1-9][0-9])*", "1")));
        options.put("-l", new ArrayList<>(Arrays.asList("[10]", "1")));
        options.put("-n", new ArrayList<>(Arrays.asList("ln|rn|rz", "ln")));

        ArrayList<String> params = new ArrayList<>();
```

```

for (int i = 1; i < args.length; i++){
    if (args[i].startsWith("-") && args[i].length() >
        1){ if (params.isEmpty()

                &&
                options.containsKey(args[i])
                && i < args.length + 1

                && args[i+1].matches(
                    options.get(args[i]).get(optionPattern)
                ))

        {
            options.put(args[i], new
                ArrayList<>( Arrays.asList(

                    options.get(args[i]).get(optionPattern)
                ), args[++i])

            )
        };

    }

    else {

        System.out.println("Wrong option " +
            args[i]); System.exit(1);

    }

}

else {
    params.add(args[i]
    );
}

}

if (params.isEmpty() || params.size() > 2){
    System.out.println("Wrong params
        count"); System.exit(1);
}

if (params.size() == 2

        && params.getFirst().equals(params.getLast())){
    System.out.println("Input file can't be output
        file"); System.exit(1);
}

Scanner
scanner; File
inFile;

```

```

if (!params.getFirst().equals("-")) {
    inFile = new
    File(params.getFirst()); if
    (!inFile.isFile()){

        System.err.println("Input file doesn't
        exist"); System.exit(1);

    }

    scanner = new
    Scanner(inFile); }
else {

    scanner = new
    Scanner(System.in); }

PrintStream outputStream;
FileOutputStream
fileOutputStream; if
(params.size() == 2){

    File outFile = new
    File(params.getLast()); if
    (!outFile.isFile()){

        if (!outFile.createNewFile()){
            System.err.println("Failed to create
            output
            fi
            le"); System.exit(1);

        }

    }

    fileOutputStream = new
    FileOutputStream(params.getLast()); outputStream = new
    PrintStream(fileOutputStream);

}

else {

    outputStream = new
    PrintStream(System.out); }

int step = Integer.parseInt(options

                                .get("-
i").get(optionValue)); boolean countEmptyLines =
"1".equals(options

                                .get("-l"))

```

```

                                .get(optionValue));

String alignType = options.get("-
n").get(optionValue); String line;
for (int lineNumber = 1;
    scanner.hasNextLine();) { line =
    scanner.nextLine();

    if
        (line.equals("\
q")) break;

    if (countEmptyLines ||
        !line.isEmpty()) { switch
        (alignType) {

            case "ln":

                line = lineNumber + " -- " +
                line; break;

            case "rn":

                line = line + " -- " +
                lineNumber; break;

            case "rz":

                line = line + " -- "

                    + String.format("%4s",
                        lineNumber) .replace(' ',
                        '0');

                br
                eak; }

            lineNumber +=
            step; }

        outputStream.println(line
        ne); }

    }

}

```

### **Входные данные:**

Файл inFile.txt

```
1 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2 Curabitur vulputate ante sapien, non aliquet risus laoreet ut.
3 Proin molestie quam tortor, consequat rhoncus neque rhoncus eu.
4 Aenean vel lorem in nunc cursus ultrices.
5 Ut rhoncus rutrum fermentum. Ut nec dolor urna.
6 In non congue orci.
7
8 Nulla luctus malesuada erat, ac faucibus felis ornare nec.
9 Integer ac pellentesque odio.
10 Vivamus lacinia cursus leo, non volutpat nibh condimentum in.
11
12 Pellentesque a nisi est. Suspendisse lobortis elit a libero lacinia,
13 quis iaculis nulla pharetra.
```

Файл run.bat

```
1 java -jar spp_lab2.jar nl -i 2 -l 0 -n rz inFile.txt outFile.txt
```

**Результат работы программы:**

Полученный файл outFile.txt

```
1 Lorem ipsum dolor sit amet, consectetur adipiscing elit. -- 0001
2 Curabitur vulputate ante sapien, non aliquet risus laoreet ut. -- 0003
3 Proin molestie quam tortor, consequat rhoncus neque rhoncus eu. -- 0005
4 Aenean vel lorem in nunc cursus ultrices. -- 0007
5 Ut rhoncus rutrum fermentum. Ut nec dolor urna. -- 0009
6 In non congue orci. -- 0011
7
8 Nulla luctus malesuada erat, ac faucibus felis ornare nec. -- 0013
9 Integer ac pellentesque odio. -- 0015
10 Vivamus lacinia cursus leo, non volutpat nibh condimentum in. -- 0017
11
12 Pellentesque a nisi est. Suspendisse lobortis elit a libero lacinia, -- 0019
13 quis iaculis nulla pharetra. -- 0021
```

**Вывод:** были приобретены базовые навыки работы с файловой системой в Java.