МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2

Специальность ПО-5(о)

Выполнил

Д. С. Бриштен,

студент группы ПО-5

Проверил

А. А. Крощенко

ст. преп. каф. ИИТ

Брест 2021

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

Вариант 4

Задание 1. Реализовать простой класс. 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.

Выполнение:

Код программы

public class One{

    public static void main(String[] args){

        Rectangle object = new Rectangle();

        System.out.println("Test? - " + object.Test());

        Rectangle rect = new Rectangle(4, 4);

        System.out.println(rect.toString());

        System.out.println("Area: " + rect.Area());

        System.out.println("Perimeter: " + rect.Perimeter());

        System.out.println("Square? - " + rect.Square());

        Rectangle rect2 = new Rectangle(3, 4);

        System.out.println(rect2.toString());

        System.out.println("rect = rect2? - " + rect.Equals(rect2));

    }

}

class Rectangle{

    private Integer a, b;

    Rectangle(){

        this.a = 0;

        this.b = 0;

    }

    Rectangle(int A, int B){

        this.a = A;

        this.b = B;

    }

    int Area(){

        return this.a \* this.b;

    }

    int Perimeter(){

        return (this.a + this.b) \* 2;

    }

    boolean Square(){

        return this.a == this.b;

    }

    boolean Test(){

        return this.a != 0 && this.b != 0;

    }

    boolean Equals(Object obj){

        return this.equals(obj);

    }

    public boolean equals(Object obj){

        if (this == obj)

            return true;

        if (obj == null || getClass() != obj.getClass())

            return false;

        Rectangle rectangle = (Rectangle) obj;

        if (a != null ? !a.equals(rectangle.a) : rectangle.a != null)

            return false;

        if (b != null ? !b.equals(rectangle.b) : rectangle.b != null)

            return false;

        return true;

    }

    public String toString(){

        return "\nRectangle with a = " + this.a + " and b = " + this.b;

    }

}

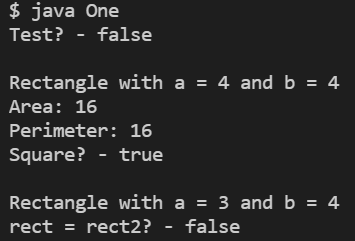
Спецификация вывода:

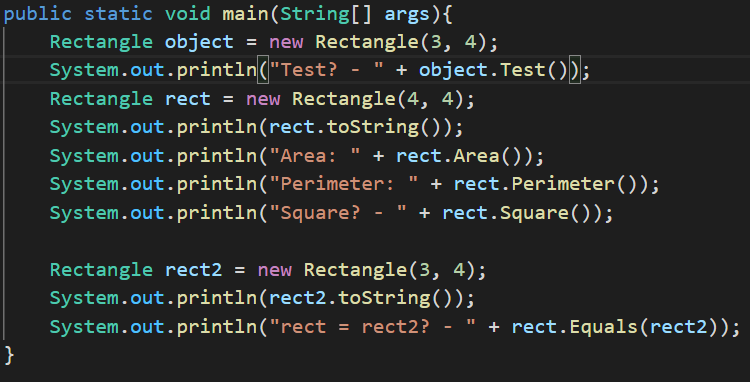
<Строка <1-я сторона прямоугольника><2-я>>

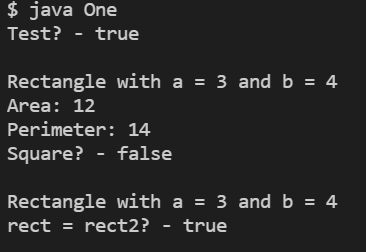
Пример:

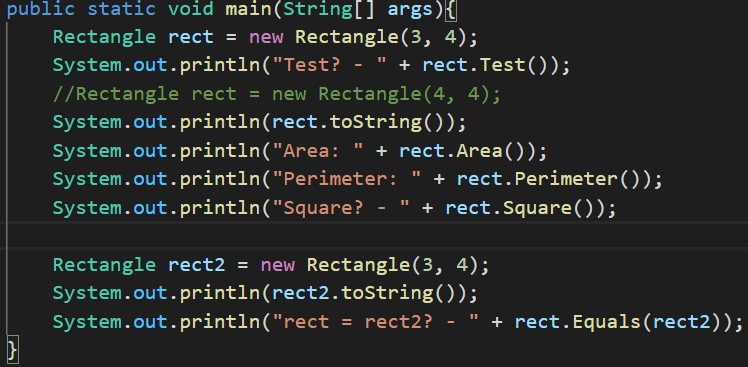


Рисунки с результатами работы программы









Задание 2. 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.

Выполнение:

Код программы

import java.io.\*;

import java.util.\*;

import java.util.stream.Collectors;

public class Library {

    private Book[] books;

    Map<String, Integer> data = Map.of(

            "author", 0,

            "title", 1,

            "publishYear", 2,

            "copies", 3,

            "pages", 4,

            "volumes", 5,

            "number", 6,

            "fio", 7,

            "period", 8

        );

    public Library(int size) {

        books = new Book[size];

    }

    public static void main(String[] args) {

        List<String> sentences = Input\_File("file.txt");

        int numberOfBooks = sentences.size();

        String[][] array = new String[numberOfBooks][];

        int i = 0;

        for (String str : sentences) {

            array[i++] = str.split("\\s");

        }

        Library lb = new Library(numberOfBooks);

        lb.testLibrary(array);

    }

    void ListOfBooks(Book[] Books) {

        for (Book book : Books) {

            System.out.print(book.getTitle() + " | ");

        }

    }

    void OlderBooks(Book[] Books, int year) {

        for (Book book : Books) {

            int pubYear = book.getYear();

            if (pubYear > year) {

                System.out.print(book.getTitle() + " | ");

            }

        }

    }

    void TakeBooks(Book[] Books) {

        for (Book book : Books) {

            String Read = book.getRead();

            if (!Read.equals("null")) {

                System.out.print(book.getTitle() + " | ");

            }

        }

    }

    void TakeBooks\_WithFIO(Book[] Books) {

        for (Book book : Books) {

            String Read = book.getRead();

            String FIO = book.getFIO();

            if (!Read.equals("null")) {

                System.out.print(book.getTitle() + ": " + FIO + " | ");

            }

        }

    }

    void Delay(Book[] Books, int BookWithReader) {

        for (Book book : Books) {

            String Read = book.getRead();

            if (!Read.equals("null") && Integer.parseInt(Read) > BookWithReader) {

                System.out.print(book.getTitle() + " | ");

            }

        }

    }

    void testLibrary(String[][] arr) {

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

            books[i] = new Book(arr[i][data.get("author")], arr[i][data.get("title")],

            arr[i][data.get("publishYear")], arr[i][data.get("copies")], arr[i][data.get("pages")],

            arr[i][data.get("volumes")], arr[i][data.get("number")], arr[i][data.get("fio")],

            arr[i][data.get("period")]);

            System.out.println(books[i].toString());

        }

        System.out.println("\n======== List of books ========");

        ListOfBooks(books);

        System.out.println("\n======== Old books(2<) ========");

        OlderBooks(books, 2);

        System.out.println("\n======== Taken books ========");

        TakeBooks(books);

        System.out.println("\n======== Taken books[FIO] ========");

        TakeBooks\_WithFIO(books);

        System.out.println("\n======== Seized books ========");

        Delay(books, 2);

    }

    public static List<String> Input\_File(String fileName) {

        List<String> sentences = new ArrayList<>();

        try(Scanner file = new Scanner(new File(fileName)))

        {

            while (file.hasNextLine()) {

                sentences.add(file.nextLine());

            }

            file.close();

        }

        catch(IOException ex){

            System.out.println(ex.getMessage());

        }

        return sentences.stream().map(String::trim).collect(Collectors.toList());

    }

}

class Book {

    private String author, title, publishYear, copies, pages, volumes, number, fio, period;

    Book() {

        this("Braun", "Point", "2001", "2", "103", "3", "323.32", "Brishten", "23");

    }

    Book(String bookAuthor, String bookName, String year, String numcopy,

        String numpage, String numvol, String UDK, String FIO, String Period) {

        this.author = bookAuthor;

        this.title = bookName;

        this.publishYear = year;

        this.copies = numcopy;

        this.pages = numpage;

        this.volumes = numvol;

        this.number = UDK;

        this.fio = FIO;

        this.period = Period;

    }

    @Override

    public String toString() {

        return "Book [[author=" + this.author + ", title=" + this.title + ", publishYear=" + this.publishYear +

        "]\n      [copies=" + this.copies + ", pages=" + this.pages  + ", volumes=" + this.volumes +

        "]\n      [number=" + this.number + ", fio=" + this.fio + ", period=" + this.period + "]]";

    }

    public String getTitle() {

        return this.title;

    }

    public Integer getYear() {

        return Integer.parseInt(this.publishYear);

    }

    public String getRead() {

        return this.period;

    }

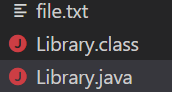
    public String getFIO() {

        return this.fio;

    }

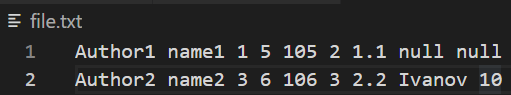
}

Имеем следующие файлы в каталоге



Спецификация ввода:

Файл чтения

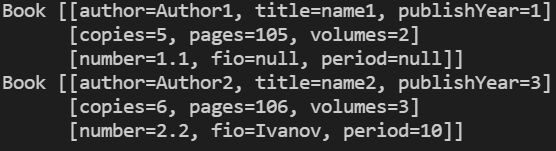


sentences (файловый ввод) = <1 элемент списка (1 строка файла)> <2 (2 строка)>...

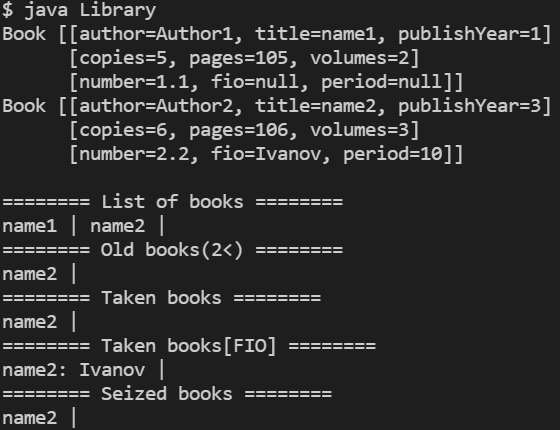
Спецификация вывода:

На примере переопределённого метода toString(): Имеем массив объектов типа Book - **books[]**, к каждому объекту которого мы обращаемся и выводим соответствующие ему свойства в формате <1 свойство> <2> <3>…

Пример:



Рисунки с результатами работы программы



Вывод: Получил начальные навыки в работе с классами и объектами в языке программирования Java.