МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

по дисциплине «Технологии разработки программного обеспечения»

# на тему: «Развитие представлений о разработке программ. Объектно-ориентированной программирование в Java»

Выполнил: студент гр. ИТИ-21

Щербенко М.А.

Принял: преподаватель

Гуменников Е. Д.

Гомель 2023

**Цель работы:** ознакомиться с основными принципами объектно-ориентированного программирования и с их реализациями на языке Java.

**Практическое задание**

**Задание.**

1. Разработать UML-диаграмму иерархии классов, согласно варианта (таблица 1).

2. При наименовании компонентов руководствоваться соглашением о наименовании.

3. При описании иерархии использовать наследование и композицию.

4. На основе UML-диаграммы разработать иерархию классов на языке Java.

5. Весь код должен быть снабжен элементами документирования.

6. Разработанную иерархию поместить в .jar файл для дальнейшего использования в качестве библиотечных классов.

7. Создать консольное приложение для демонстрации работы созданных классов.

8. Составить отчет о проделанной работе.

Таблица 1

|  |  |
| --- | --- |
| **Вариант** | Условие задачи |
| 8 | 1.1 Создать иерархию для учета книг в библиотеках.  1.2 Библиотека содержит в себе название и коллекцию книг, а также методы добавления книги.  1.3 Класс книги содержит поля: название, автор, год написания. 1.4 Создать несколько объектов библиотек и добавить книги в каждую.  1.5 Вывести все книги одного автора из каждой библиотеки.  1.6 Вывести всю информация по книге(название, автор, год написания). |

**Ход работы**

**Порядок выполнения задания.**

На рисунке 1 показана UML-диаграмма задания.

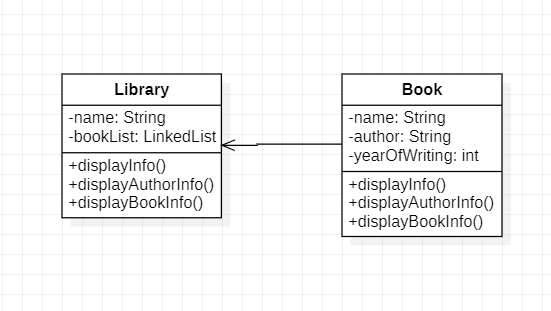


Рисунок 1 – UML-диаграмма поставленной задачи

На рисунке 2 показано меню основных функций программы.

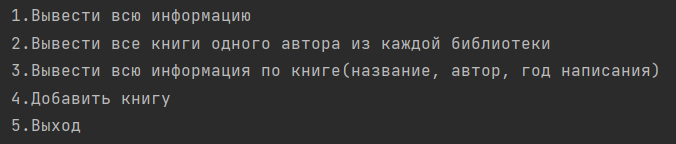


Рисунок 2 – Меню основных функций программы

На рисунке 3 показан результат выполнения пункта 1 меню.

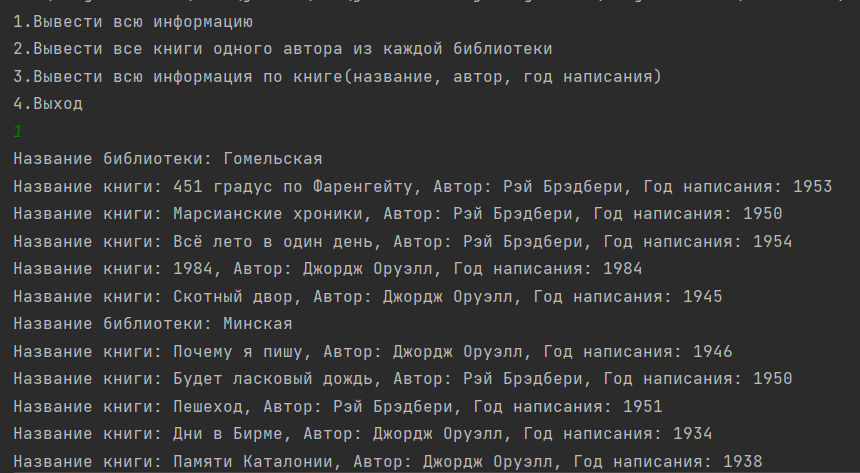


Рисунок 3 – Вывод всей информации

На рисунке 4 и 5 показано выполнение 2 пункта меню.

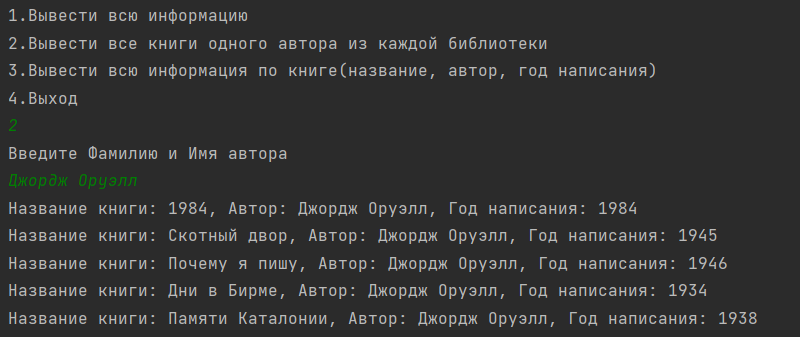


Рисунок 4 – Выполнение 2 пункта меню

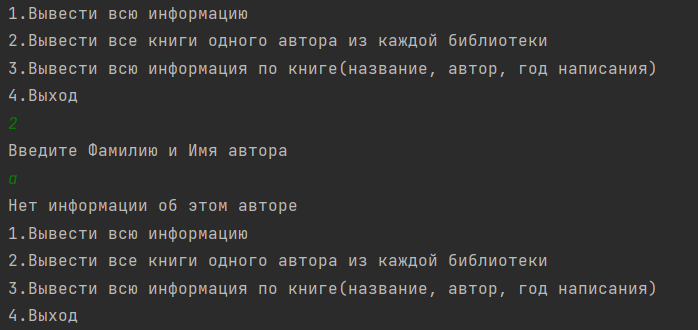


Рисунок 5 – Выполнение 2 пункта меню

На рисунках 6 и 7 показано выполнение 3 функции программы.

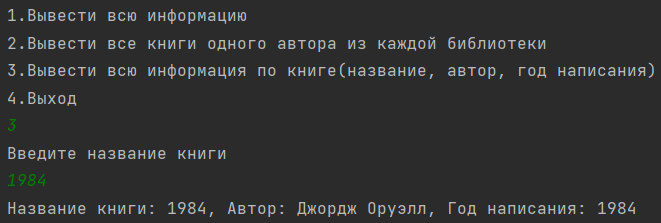


Рисунок 6 – Вывод информации о книге

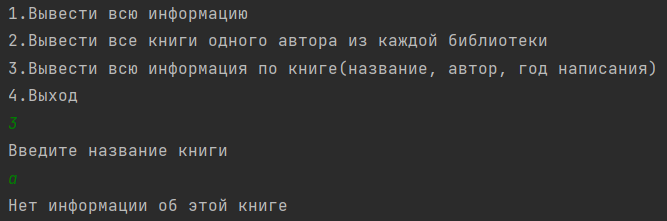


Рисунок 7 – Выполнение 3 пункта меню

На рисунках 8 показано выполнение 4 функции программы.

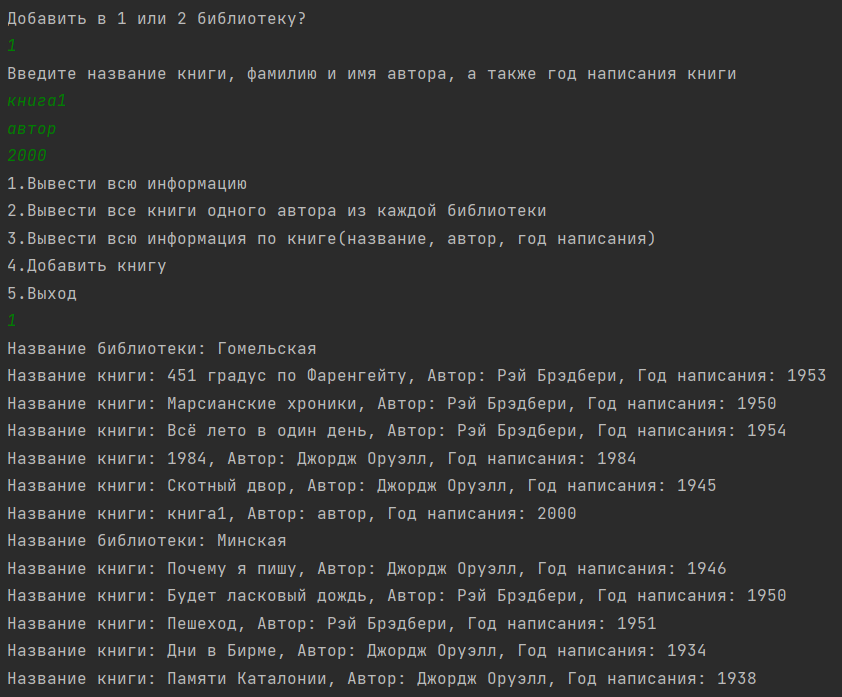


Рисунок 8 – Выполнение 4 пункта меню

**Вывод:** в процессе работы ознакомились с основными принципами объектно-ориентированного программирования, а также с созданием классов и объектов на языке Java.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Текст программы**

**Класс Main:**

import java.util.\*;  
  
public class Main {  
 public Main() {  
 }  
  
 public static void main(String[] args) {  
 LinkedList<Library> libraryList = new LinkedList();  
 LinkedList<Book> bookListFirst = new LinkedList();  
  
 bookListFirst.add(new Book("451 градус по Фаренгейту", "Рэй Брэдбери", 1953));  
 bookListFirst.add(new Book("Марсианские хроники", "Рэй Брэдбери", 1950));  
 bookListFirst.add(new Book("Всё лето в один день", "Рэй Брэдбери", 1954));  
 bookListFirst.add(new Book("1984", "Джордж Оруэлл", 1984));  
 bookListFirst.add(new Book("Скотный двор", "Джордж Оруэлл", 1945));  
 libraryList.add(new Library("Гомельская", bookListFirst));  
  
 LinkedList<Book> bookListSecond = new LinkedList();  
 bookListSecond.add(new Book("Почему я пишу", "Джордж Оруэлл", 1946));  
 bookListSecond.add(new Book("Будет ласковый дождь", "Рэй Брэдбери", 1950));  
 bookListSecond.add(new Book("Пешеход", "Рэй Брэдбери", 1951));  
 bookListSecond.add(new Book("Дни в Бирме", "Джордж Оруэлл", 1934));  
 bookListSecond.add(new Book("Памяти Каталонии", "Джордж Оруэлл", 1938));  
 libraryList.add(new Library("Минская", bookListSecond));  
  
 Scanner in = new Scanner(System.*in*);  
  
  
  
 int k = 0;  
 while(k==0) {  
  
  
 int authorFind = 0;  
 int bookFind = 0;  
 System.*out*.print("1.Вывести всю информацию\n");  
 System.*out*.print("2.Вывести все книги одного автора из каждой библиотеки\n");  
 System.*out*.print("3.Вывести всю информация по книге(название, автор, год написания)\n");  
 System.*out*.print("4.Добавить книгу\n");  
 System.*out*.print("5.Выход\n");  
 int num = in.nextInt();  
 switch (num) {  
 case 1:  
 Iterator var1 = libraryList.iterator();  
  
 while (var1.hasNext()) {  
 Library library = (Library) var1.next();  
 library.displayInfo();  
 }  
 break;  
 case 2:  
 System.*out*.print("Введите Фамилию и Имя автора\n");  
 Scanner st1 = new Scanner(System.*in*);  
 String authorName = st1.nextLine();  
 Iterator var2 = libraryList.iterator();  
  
 while (var2.hasNext()) {  
 Library library = (Library) var2.next();  
 library.displayAuthorInfo(authorName, authorFind);  
 }  
 if(authorFind==0)  
 System.*out*.print("Нет информации об этом авторе\n");  
 break;  
 case 3:  
 System.*out*.print("Введите название книги\n");  
 Scanner st2 = new Scanner(System.*in*);  
 String bookName = st2.nextLine();  
 Iterator var3 = libraryList.iterator();  
  
 while (var3.hasNext()) {  
 Library library = (Library) var3.next();  
 library.displayBookInfo(bookName, bookFind);  
 }  
 if(bookFind==0)  
 System.*out*.print("Нет информации об этой книге\n");  
 break;  
 case 4:  
 System.*out*.print("Добавить в 1 или 2 библиотеку?\n");  
 Scanner st3 = new Scanner(System.*in*);  
 int numberOfLibrary = in.nextInt();  
 System.*out*.print("Введите название книги, фамилию и имя автора, а также год написания книги\n");  
 Scanner st4 = new Scanner(System.*in*);  
 String bookNameForAdd = st4.nextLine();  
 String authorNameForAdd = st4.nextLine();  
 int yearForAdd = st4.nextInt();  
 switch (numberOfLibrary){  
 case 1:  
 libraryList.removeFirst();  
 bookListFirst.add(new Book(bookNameForAdd,authorNameForAdd, yearForAdd));  
 libraryList.addFirst(new Library("Гомельская", bookListFirst));  
 break;  
 case 2:  
 libraryList.removeLast();  
 bookListSecond.add(new Book(bookNameForAdd,authorNameForAdd, yearForAdd));  
 libraryList.add(new Library("Минская", bookListSecond));  
 break;  
 default:  
 System.*out*.println("Проверьте правильность данных");  
 }  
 break;  
 case 5:  
 k++;  
 break;  
 default:  
 System.*out*.println("Проверьте правильность данных");  
 }  
 }  
  
  
 }  
}

**Класс Library:**

import java.util.Iterator;  
import java.util.LinkedList;  
  
class Library {  
 String name;  
 LinkedList<Book> bookList = new LinkedList();  
  
 Library(String name, LinkedList<Book> bookList) {  
 this.name = name;  
 this.bookList = bookList;  
 }  
  
 void displayInfo() {  
 System.out.printf("Название библиотеки: %s\n", this.name);  
 Iterator var1 = this.bookList.iterator();  
  
 while (var1.hasNext()) {  
 Book bookList = (Book) var1.next();  
 bookList.displayInfo();  
 }  
 }  
  
 void displayAuthorInfo(String authorName, int authorFind) {  
 Iterator var4 = this.bookList.iterator();  
  
 while (var4.hasNext()) {  
 Book bookList = (Book) var4.next();  
 bookList.displayAuthorInfo(authorName, authorFind);  
 }  
 }  
 void displayBookInfo(String bookName, int bookFind) {  
 Iterator var6 = this.bookList.iterator();  
  
 while (var6.hasNext()) {  
 Book bookList = (Book) var6.next();  
 bookList.displayBookInfo(bookName, bookFind);  
 }  
 }  
}

**Класс Book:**

class Book {  
 String name;  
 String author;  
 int yearOfWriting;  
  
 Book(String name, String author, int yearOfWriting) {  
 this.name = name;  
 this.author = author;  
 this.yearOfWriting = yearOfWriting;  
 }  
  
 void displayInfo() {  
 System.out.printf("Название книги: %s, Автор: %s, Год написания: %d\n", this.name, this.author, this.yearOfWriting);  
 }  
 void displayAuthorInfo(String authorName, int authorFind) {  
 if(authorName.equals(this.author)) {  
 System.out.printf("Название книги: %s, Автор: %s, Год написания: %d\n", this.name, this.author, this.yearOfWriting);  
 authorFind++;  
 }  
 }  
 void displayBookInfo(String bookName, int bookFind) {  
 if(bookName.equals(this.name)) {  
 System.out.printf("Название книги: %s, Автор: %s, Год написания: %d\n", this.name, this.author, this.yearOfWriting);  
 bookFind++;  
 }  
 }  
}