**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №1

По дисциплине: **«Технологии разработки программного обеспечения»**

на тему: **Развитие представлений о разработке программ. Объектно-ориентированное программирование в *Java***

Выполнил: студент группы ИТП-21

Шинкарук В.Г.

Проверил:

Малиновский И.Л.

Гомель 2024

**Цель работы**: изучить особенности разработки в объектно-ориентированном языке программирования, используя механизмы наследования и композиции. Развить представление о разработке программ на *Java*.

**Задание:**

1. Разработать *UML*-диаграмму иерархии классов, согласно варианта.

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.

**Ход выполнения и результаты работы:**

Текстовый вариант задания для варианта 23 представлен на рисунке 1.

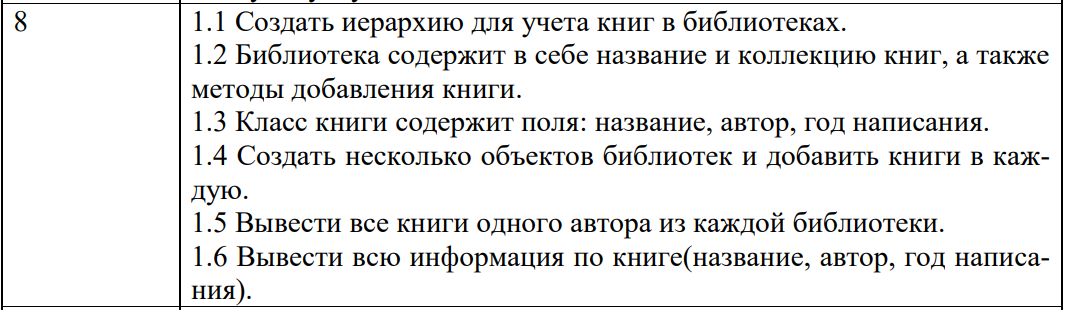


Рисунок 1 – Задание варианта 23

На рисунке 2 представлено меню консольного приложения программы.

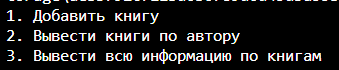


Рисунок 2 – Меню программы

На рисунке 3 представлено выполнение добавления книги.

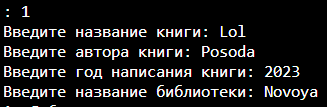


Рисунок 3 – Выполнение пункта меню 1

На рисунке 4 представлено выполнение вывода книги по автору.

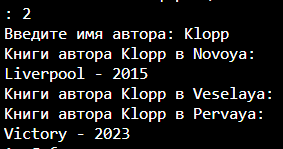


Рисунок 4 – Выполнение пункта меню 2

На рисунке 5 представлен вывод всей информации по книгам.

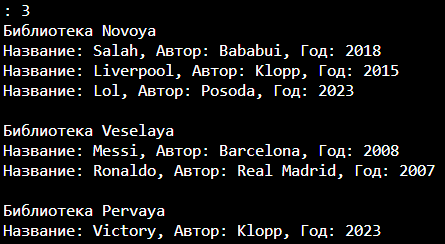


Рисунок 5 – Выполнение пункта меню 3

Текст класса *Library.java* представлен в приложении А.

Текст класса *Book.java* представлен в приложении Б.

Текст класса *Menu.java* представлен в приложении В.

Текст класса *Main.java* представлен в приложении Г.

Текст класса *LibraryActions.java* представлен в приложении Д.

На рисунке 6 представлена *UML*-диаграмма иерархии классов.

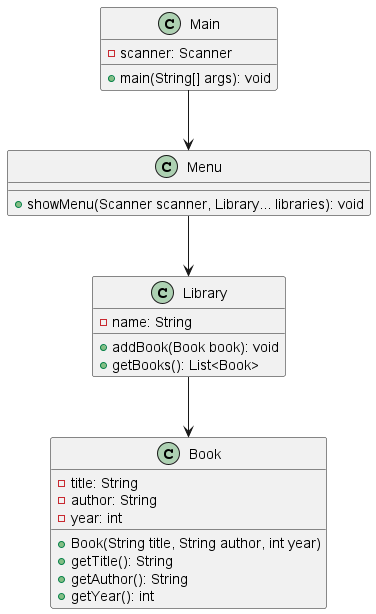


Рисунок 6 – *UML*-диаграмма иерархии классов

**Вывод:** в ходе выполнения лабораторной работы изучены особенности разработки в объектно-ориентированном языке программирования, используя механизмы наследования и композиции. В результате выполнения лабораторной работы разработана программа для учёта книг в библиотеках.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Текст *Library.java***

package Lab1;

import java.util.ArrayList;

import java.util.List;

class Library {

private String libraryName;

private List<Book> books;

public Library(String libraryName) {

this.libraryName = libraryName;

this.books = new ArrayList<>();

}

public String getName() {

return libraryName;

}

public List<Book> getBooks() {

return books;

}

public void addBook(Book book) {

books.add(book);

}

public List<Book> getBooksByAuthor(String author) {

List<Book> booksByAuthor = new ArrayList<>();

for (Book book : books) {

if (book.getAuthor().equals(author)) {

booksByAuthor.add(book);

}

}

return booksByAuthor;

}

}

**ПРИЛОЖЕНИЕ Б**

(обязательное)

**Текст *Book.java***

package Lab1;

class Book {

private String title;

private String author;

private int year;

public Book(String title, String author, int year) {

this.title = title;

this.author = author;

this.year = year;

}

public String getTitle() {

return title;

}

public String getAuthor() {

return author;

}

public int getYear() {

return year;

}

}

**ПРИЛОЖЕНИЕ В**

(обязательное)

**Текст *Menu.java***

package Lab1;

import java.util.Scanner;

public class Menu {

public static void showMenu(Scanner scanner, Library... libraries) {

int choice;

do {

System.out.println("1. Добавить книгу");

System.out.println("2. Вывести книги по автору");

System.out.println("3. Вывести всю информацию по книгам");

System.out.print(": ");

choice = scanner.nextInt();

scanner.nextLine();

switch (choice) {

case 1:

LibraryActions.addBook(scanner, libraries);

break;

case 2:

LibraryActions.printBooksByAuthor(scanner, libraries);

break;

case 3:

LibraryActions.printAllBooks(libraries);

break;

default:

System.out.println("Неверный выбор");

break;

}

} while (true);

}

}

**ПРИЛОЖЕНИЕ Г**

(обязательное)

**Текст *Main.java***

package Lab1;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Library library1 = new Library("Novoya");

Library library2 = new Library("Veselaya");

Library library3 = new Library("Pervaya");

library1.addBook(new Book("Salah", "Bababui", 2018));

library1.addBook(new Book("Liverpool", "Klopp", 2015));

library2.addBook(new Book("Messi", "Barcelona", 2008));

library2.addBook(new Book("Ronaldo", "Real Madrid", 2007));

library3.addBook(new Book("Victory", "Klopp", 2023));

Menu.showMenu(scanner, library1, library2, library3);

}

}

**ПРИЛОЖЕНИЕ Д**

(обязательное)

**Текст *LibraryActions.java***

package Lab1;

import java.util.Scanner;

public class LibraryActions {

public static void addBook(Scanner scanner, Library... libraries) {

System.out.print("Введите название книги: ");

String title = scanner.nextLine();

System.out.print("Введите автора книги: ");

String author = scanner.nextLine();

System.out.print("Введите год написания книги: ");

int year = scanner.nextInt();

scanner.nextLine();

System.out.print("Введите название библиотеки: ");

String libraryName = scanner.nextLine();

for (Library library : libraries) {

if (library.getName().equals(libraryName)) {

library.addBook(new Book(title, author, year));

}

}

}

public static void printBooksByAuthor(Scanner scanner, Library... libraries) {

System.out.print("Введите имя автора: ");

String authorToSearch = scanner.nextLine();

for (Library library : libraries) {

System.out.println("Книги автора " + authorToSearch + " в " + library.getName() + ":");

for (Book book : library.getBooksByAuthor(authorToSearch)) {

System.out.println(book.getTitle() + " - " + book.getYear());

}

}

}

public static void printAllBooks(Library... libraries) {

for (Library library : libraries) {

System.out.println("Библиотека " + library.getName());

for (Book book : library.getBooks()) {

System.out.println("Название: " + book.getTitle() + ", Автор: " + book.getAuthor() + ", Год: " + book.getYear());

}

System.out.println();

}

}

}