Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**Иркутский национальный исследовательский технический университет**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |
| Кафедра вычислительной техники |
| наименование кафедры |

|  |
| --- |
| **Отчет** |
| по лабораторной работе №4 по дисциплине «Разработка классов с использованием механизмов наследования, полиморфизма и инкапсуляции.» |
| наименование темы  Вариант №21 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  | АСУб-19-1 |  |  |  | Казаков С.Э. |
|  |  | шифр |  | подпись |  | И.О. Фамилия |
| Проверил |  |  |  |  |  | Маланова Т.В. |
|  |  |  |  | подпись |  | И.О. Фамилия |
| Работа защищена с оценкой | | | |  | | |

Иркутск 2020 г.

**Оглавление**

[1 Постановка задачи 3](#_Toc56894035)

[2 Описание структуры пользовательского меню 4](#_Toc56894036)

[3 Проектирование классов 5](#_Toc56894037)

[4 Таблица тестов 6](#_Toc56894038)

[5 Результаты тестирования 7](#_Toc56894039)

[6 Листинг исходного кода 14](#_Toc56894040)

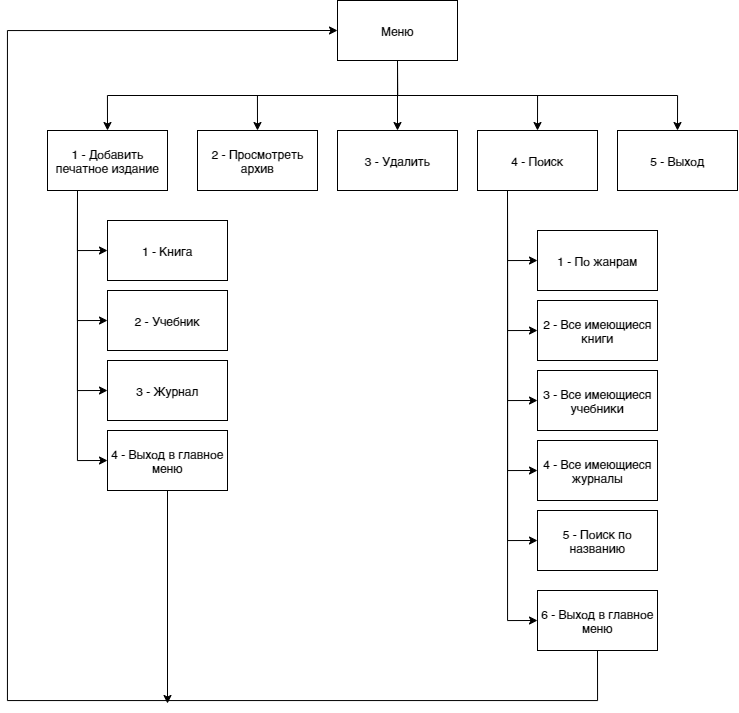
[Список использованных источников 30](#_Toc56894041)

1 Постановка задачи

В соответствии с индивидуальным заданием описать иерархию классов, для каждого класса описать поля и соответствующие методы доступа к ним. В зависимости от задания некоторые из этих классов (как минимум один) являются абстрактными и служат для выделения общих данных и поведения для других классов. Абстрактный класс должен содержать как минимум один абстрактный метод, реализация которого у его наследников должна различаться. Помимо этого, в общую часть задания входит разработка класса, группирующего объекты описанных в соответствии с заданием классов. Для первого варианта индивидуального задания это может быть класс «кафедра». Для выполнения задания необходимо создать некоторое количество объектов, добавить их в группу используя предусмотренные методы класса-«группы» и для каждого из них вызвать унаследованный метод.

Вариант 6 (21): журнал, книга, печатное издание, учебник.

2 Описание структуры пользовательского меню

  
Рис 1 – Структура пользовательского меню

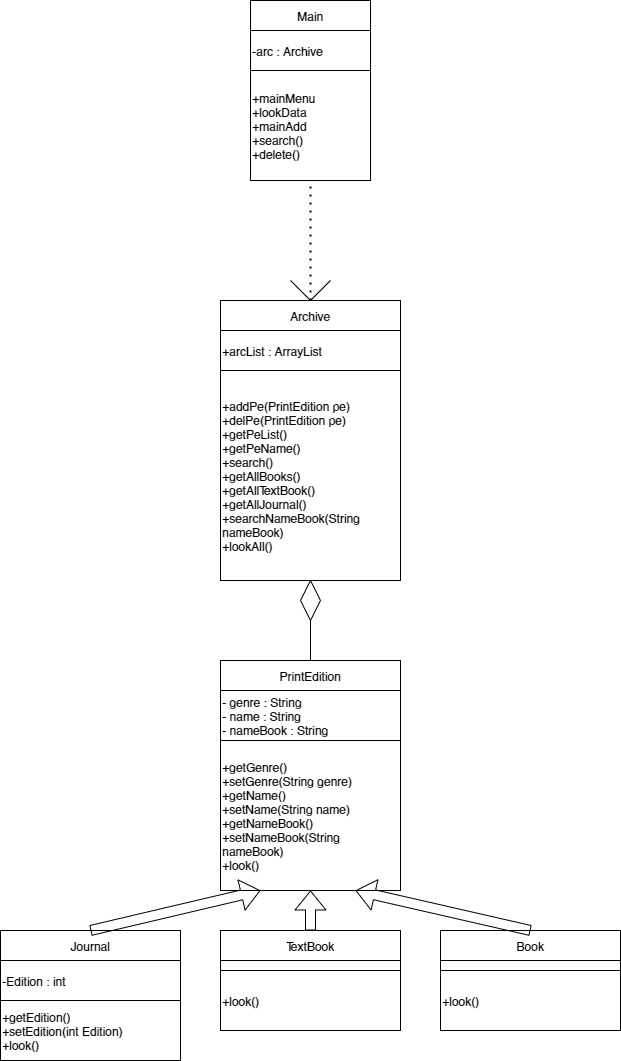
3 Проектирование классов

Рис 2 – Проектирование классов

4 Таблица тестов

Таблица 1 – Таблица тестов

|  |  |  |
| --- | --- | --- |
| № | Описание теста | Результат работы программы |
| 1 | Ввод в меню числа, не существующего пунктам меню | Сообщение об ошибке, повторение данного пункта меню |
| 2 | Ввод в главном меню «1» | Переход в меню «Добавить печатное издание» |
| 3 | Ввод в меню Добавить данные «1» | Предлагает ввести данные |
| 4 | Вводим данные | Возвращение в меню добавления данных. |
| 5 | Ввод в меню Добавить данные «2» | Предлагает ввести данные |
| 6 | Вводим данные | Возвращение в меню добавления данных. |
| 7 | Ввод в меню Добавить данные «3» | Предлагает ввести данные |
| 8 | Вводим данные | Возвращение в меню добавления данных. |
| 9 | Ввод в главном меню «2» | Вывод всех печатных изданий, Выход в главное меню |
| 10 | Ввод в главном меню «3» | Переход в меню «Удалить», выводятся все пронумерованные рукописи и при выборе цифры рукопись удаляется |
| 11 | Ввод в главном меню «4» | Переход в меню «Поиск» |
| 12 | Ввод в меню Поиск «1» | Предлагается ввести жанр |
| 13 | Ввод жанра | Вывод печатных изданий с таким же жанром |
| 14 | Ввод в меню Поиск «2» | Выводятся все книги, которые записаны |
| 15 | Ввод в меню Поиск «5» | Предлагается ввести название книги |
| 16 | Ввод в главном меню «5» | Выход из программы |

5 Результаты тестирования

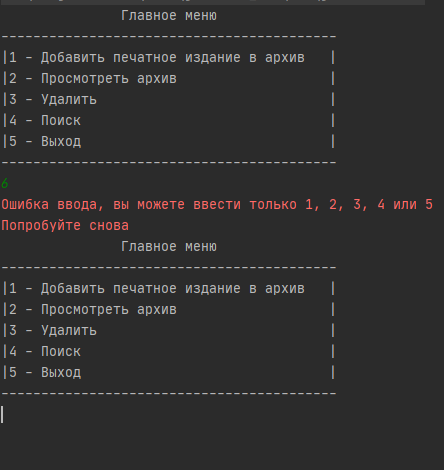


Рис 3 – Тест 1

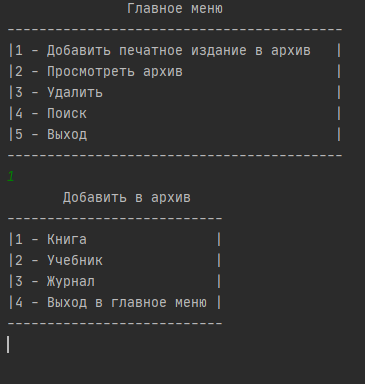


Рис 4 – Тест 2

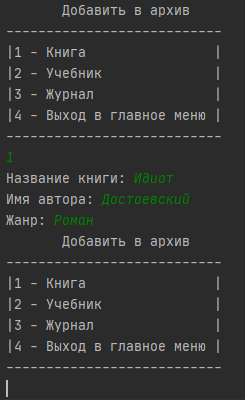


Рис 5 – Тест 3 и 4

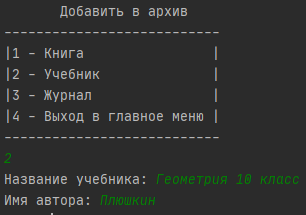


Рис 6 – Тест 5 и 6

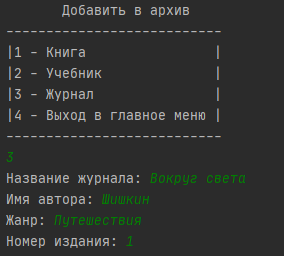


Рис 7 – Тест 7 и 8

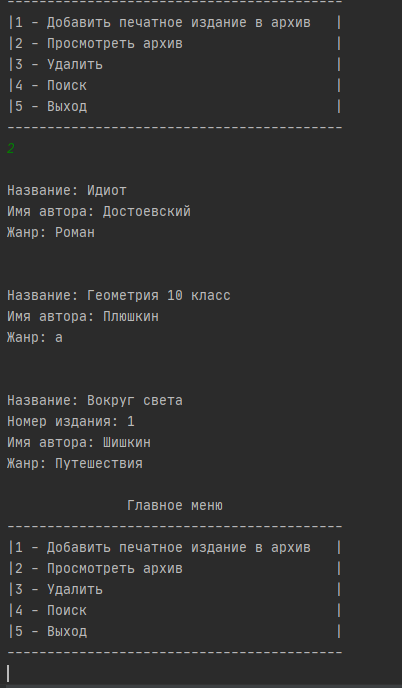


Рис 8 – Тест 9

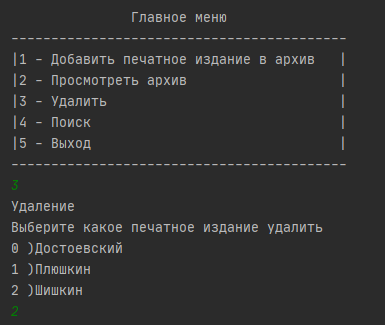


Рис 8 – Тест 10

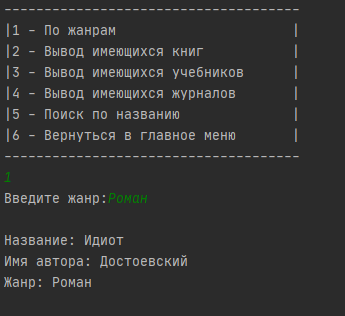


Рис 9 – Тест 11, 12 и 13

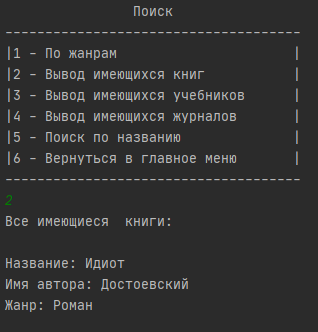


Рис 10 – Тест 14

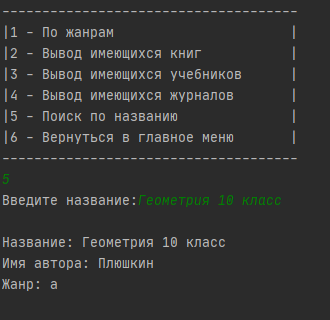


Рис 11 – Тест 15

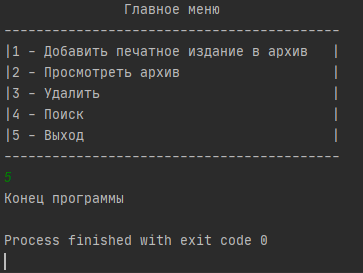


Рис 12 – Тест 16

5 Листинг исходного кода

1) Main

package com.company.Menu;  
  
import com.company.Archive;  
import com.company.Data.Book;  
import com.company.Data.Journal;  
import com.company.Data.TextBook;  
import com.company.Data.PrintEdition;  
  
import java.util.ArrayList;  
import java.util.InputMismatchException;  
import java.util.Scanner;  
  
public class Main {  
  
 private static Archive *arc* = new Archive();  
 private static Scanner *in* = new Scanner(System.*in*);  
 public static void main(String[] args) {  
 *mainMenu*();  
 }  
 public static void mainMenu(){  
 int a = 0;  
 do{  
 System.*out*.println(" Главное меню ");  
 System.*out*.println("------------------------------------------");  
 System.*out*.println("|1 - Добавить печатное издание в архив |");  
 System.*out*.println("|2 - Просмотреть архив |");  
 System.*out*.println("|3 - Удалить |");  
 System.*out*.println("|4 - Поиск |");  
 System.*out*.println("|5 - Выход |");  
 System.*out*.println("------------------------------------------");  
  
 try {  
 a = Integer.*parseInt*(*in*.nextLine());  
 if (a < 1 || a > 5) throw new IllegalArgumentException();  
 } catch (IllegalArgumentException e){  
 System.*err*.println("Ошибка ввода, вы можете ввести только 1, 2, 3, 4 или 5");  
 System.*err*.println("Попробуйте снова");  
 }  
  
 switch (a){  
 case 1:  
 *mainAdd*();  
 break;  
 case 2:  
 *lookData*();  
 break;  
 case 3:  
 *delete*();  
 break;  
 case 4:  
 *search*();  
 break;  
 }  
  
 }while(a != 5);  
 System.*out*.println("Конец программы");  
 }  
 public static void lookData(){  
 *arc*.lookAll();  
 }  
  
 public static void mainAdd(){  
 int s=0;  
 do {  
 System.*out*.println(" Добавить в архив ");  
 System.*out*.println("---------------------------");  
 System.*out*.println("|1 - Книга |");  
 System.*out*.println("|2 - Учебник |");  
 System.*out*.println("|3 - Журнал |");  
 System.*out*.println("|4 - Выход в главное меню |");  
 System.*out*.println("---------------------------");  
  
 try {  
 s = Integer.*parseInt*(*in*.nextLine());  
 if (s < 1 || s > 4) throw new IllegalArgumentException();  
 } catch (IllegalArgumentException e) {  
 System.*err*.println("Ошибка ввода, вы можете ввести только 1, 2, 3 или 4");  
 }  
 String genre, name, nameBook;  
 int edition;  
  
 switch (s){  
 case 1:  
 System.*out*.print("Название книги: ");  
 nameBook = *in*.nextLine();  
  
 System.*out*.print("Имя автора: ");  
 name = *in*.nextLine();  
  
 System.*out*.print("Жанр: ");  
 genre = *in*.nextLine();  
  
 Book book = new Book(genre, name, nameBook);  
 *arc*.addPe(book);  
 break;  
 case 2:  
 System.*out*.print("Название учебника: ");  
 nameBook = *in*.nextLine();  
  
 System.*out*.print("Имя автора: ");  
 name = *in*.nextLine();  
  
 System.*out*.print("Жанр: ");  
 genre = *in*.nextLine();  
  
 TextBook textbook = new TextBook(genre, name, nameBook);  
 *arc*.addPe(textbook);  
 break;  
 case 3:  
 System.*out*.print("Название журнала: ");  
 nameBook = *in*.nextLine();  
  
 System.*out*.print("Имя автора: ");  
 name = *in*.nextLine();  
  
 System.*out*.print("Жанр: ");  
 genre = *in*.nextLine();  
  
  
 System.*out*.print("Номер издания: ");  
 try {  
 edition = *in*.nextInt();  
 } catch (InputMismatchException e){  
 System.*err*.println("Номер издания не может быть словом");  
 edition = 0;  
 }  
 *in*.nextLine();  
 Journal journal = new Journal(edition, genre, name, nameBook);  
 *arc*.addPe(journal);  
 break;  
 }  
 }while (s!=4);  
 }  
  
 public static void search(){  
  
 int b = 0;  
 do {  
 System.*out*.println(" Поиск ");  
 System.*out*.println("-------------------------------------");  
 System.*out*.println("|1 - По жанрам |");  
 System.*out*.println("|2 - Вывод имеющихся книг |");  
 System.*out*.println("|3 - Вывод имеющихся учебников |");  
 System.*out*.println("|4 - Вывод имеющихся журналов |");  
 System.*out*.println("|5 - Поиск по названию |");  
 System.*out*.println("|6 - Вернуться в главное меню |");  
 System.*out*.println("-------------------------------------");  
  
 try {  
 b = Integer.*parseInt*(*in*.nextLine());  
 if (b < 1 || b > 6) throw new IllegalArgumentException();  
 } catch (IllegalArgumentException e) {  
 System.*err*.println("Ошибка ввода, вы можете ввести только 1, 2, 3, 4, 5 или 6");  
 }  
  
 switch (b) {  
 case 1:  
 System.*out*.print("Введите жанр:");  
 String h = *in*.nextLine();  
 ArrayList<PrintEdition> found = *arc*.search(h);  
 if (found.isEmpty()) {  
 System.*err*.println("Печатных изданий с данным жанром не найдено");  
 }  
 else {  
 for (PrintEdition a: found) {  
 a.look();  
 }  
 }  
 break;  
  
 case 2:  
 System.*out*.println("Все имеющиеся книги:");  
 ArrayList<Book> find = *arc*.getAllBooks();  
 if (find.isEmpty()){  
 System.*err*.println("Ни одной книги не найдено");  
 } else {  
 for (PrintEdition r : find) {  
 r.look();  
 }  
 }  
 break;  
 case 3:  
 System.*out*.println("Все имеющиеся учебники:");  
 ArrayList<TextBook> finde = *arc*.getAllTextBook();  
 if (finde.isEmpty()){  
 System.*err*.println("Ни одного учебника не найдено");  
 } else {  
 for (PrintEdition r : finde) {  
 r.look();  
 }  
 }  
 break;  
 case 4:  
 System.*out*.println("Все имеющиеся журналы:");  
 ArrayList<Journal> fin = *arc*.getAllJournal();  
 if (fin.isEmpty()){  
 System.*err*.println("Ни одного учебника не найдено");  
 } else {  
 for (PrintEdition r : fin) {  
 r.look();  
 }  
 }  
 break;  
 case 5:  
 System.*out*.print("Введите название:");  
 String q = *in*.nextLine();  
 ArrayList<PrintEdition> f = *arc*.searchNameBook(q);  
 if (f.isEmpty()){  
 System.*err*.println("Печатных изданий с данным названием не найдено");  
 } else {  
 for (PrintEdition a : f) {  
 a.look();  
 }  
 }  
 break;  
 }  
 }while (b != 6);  
 }  
  
 public static void delete(){  
  
 int n;  
 String[] PeNames = *arc*.getPeName();  
 do {  
 System.*out*.println("Удаление");  
 System.*out*.println("Выберите какое печатное издание удалить");  
  
 for (int i = 0; i < PeNames.length; i++) {  
 System.*out*.println(i + " )" + PeNames[i]);  
 }  
 n = Integer.*parseInt*(*in*.nextLine());  
  
 }while(n < 0 || n > PeNames.length);  
  
 *arc*.delPe(*arc*.getPeList().get(n));  
 }  
}

2) PrintEdition

package com.company.Data;  
  
public abstract class PrintEdition {  
 private String genre;  
 private String name;  
 private String nameBook;  
  
  
 public PrintEdition(String genre, String name, String nameBook){  
 this.genre = genre;  
 this.name = name;  
 this.nameBook = nameBook;  
 }  
  
 public String getGenre(){  
 return genre;  
 }  
 public void setGenre(String genre){  
 this.genre = genre;  
 }  
  
 public String getName(){ return name; }  
 public void setName(String name){  
 this.name = name;  
 }  
  
 public String getNameBook(){  
 return nameBook;  
 }  
 public void setNameBook(String nameBook){  
 this.nameBook = nameBook;  
 }  
 public abstract void look();  
}

3) Book

package com.company.Data;  
  
public class Book extends PrintEdition {  
 public Book(String genre, String name, String nameBook) {  
 super(genre, name, nameBook);  
 }  
 public void look(){  
 System.*out*.println();  
 System.*out*.println("Название: " + getNameBook());  
 System.*out*.println("Имя автора: " + getName());  
 System.*out*.println("Жанр: " + getGenre());  
 System.*out*.println();  
 }  
}

4) Journal

package com.company.Data;  
  
public class Journal extends PrintEdition {  
  
 private int Edition;  
 public Journal(int edition, String genre, String name, String nameBook) {  
 super(genre, name, nameBook);  
 this.Edition = edition;  
 }  
  
 public int getEdition() { return Edition; }  
 public void setEdition (int Edition) { this.Edition = Edition; }  
  
 public void look(){  
 System.*out*.println();  
 System.*out*.println("Название: " + getNameBook());  
 System.*out*.println("Номер издания: " + getEdition());  
 System.*out*.println("Имя автора: " + getName());  
 System.*out*.println("Жанр: " + getGenre());  
 System.*out*.println();  
 }  
}

5) TextBook

package com.company.Data;  
  
public class TextBook extends PrintEdition {  
 public TextBook(String genre, String name, String nameBook) {  
 super(genre, name, nameBook);  
 }  
 public void look(){  
 System.*out*.println();  
 System.*out*.println("Название: " + getNameBook());  
 System.*out*.println("Имя автора: " + getName());  
 System.*out*.println("Жанр: " + getGenre());  
 System.*out*.println();  
 }  
}

6)Library

package com.company;  
  
import com.company.Data.Book;  
import com.company.Data.Journal;  
import com.company.Data.PrintEdition;  
import com.company.Data.TextBook;  
  
import java.util.ArrayList;  
  
public class Archive {  
 ArrayList<PrintEdition> ArcList = new ArrayList<>();  
   
 public void addPe(PrintEdition pe){  
 ArcList.add(pe);  
 }  
   
 public void delPe(PrintEdition pe){  
 ArcList.remove(pe);  
 }  
   
 public ArrayList<PrintEdition> getPeList(){  
 return ArcList;  
 }  
   
 public String[] getPeName(){  
 String[] names = new String[ArcList.size()];  
 for (int i = 0; i < ArcList.size(); i++){  
 names[i] = ArcList.get(i).getName();  
 }  
 return names;  
 }  
   
 public ArrayList<PrintEdition> search(String genre){  
 ArrayList<PrintEdition> answer = new ArrayList<>();  
 for (PrintEdition a:ArcList) {  
 if (a.getGenre().contains(genre)) {  
 answer.add(a);  
 }  
 }  
 return answer;  
 }  
   
 public ArrayList<Book> getAllBooks(){  
 ArrayList<Book> answer = new ArrayList<>();  
 for (PrintEdition a:ArcList) {  
 if (a instanceof Book) {  
 answer.add((Book) a);  
 }  
 }  
 return answer;  
 }  
   
 public ArrayList<Journal> getAllJournal(){  
 ArrayList<Journal> answer = new ArrayList<>();  
 for (PrintEdition a:ArcList) {  
 if (a instanceof Journal) {  
 answer.add((Journal) a);  
 }  
 }  
 return answer;  
 }  
   
 public ArrayList<TextBook> getAllTextBook(){  
 ArrayList<TextBook> answer = new ArrayList<>();  
 for (PrintEdition a:ArcList) {  
 if (a instanceof TextBook) {  
 answer.add((TextBook) a);  
 }  
 }  
 return answer;  
 }  
   
 public ArrayList<PrintEdition> searchNameBook(String nameBook) {  
 ArrayList<PrintEdition> answer = new ArrayList<>();  
 for (PrintEdition a : ArcList) {  
 if (a.getNameBook().contains(nameBook)) {  
 answer.add(a);  
 }  
 }  
 return answer;  
 }  
   
 public void lookAll(){  
 for (PrintEdition a:ArcList) {  
 a.look();  
 }  
 }  
}

Список использованных источников

1. Методические указания по выполнению лабораторных работ / сост.: В.Л. Аршинский. – Иркутск : Изд-во ИРНИТУ, 2017. – 24 c.
2. Файн Я. – Программирование на Java для детей, родителей, дедушек и бабушек, 2011