

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

ОТЧЕТ
по лабораторной работе №3
по дисциплине СПП

Выполнила:
студ. гр.ПО-3
Гаврилкович Е.В.

Проверил:
Крощенко А.А.

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

Вариант 5

Задание 1

Реализовать простой класс.

Требования к выполнению

- Реализовать пользовательский класс по варианту.
- Создать другой класс с методом `main`, в котором будут находиться примеры использования пользовательского класса.

Для каждого класса

- Создать поля классов
- Создать методы классов
- Добавьте необходимые `get` и `set` методы (по необходимости)
- Укажите соответствующие модификаторы видимости
- Добавьте конструкторы
- Переопределить методы `toString()` и `equals()`

Множество целых чисел ограниченной мощности – Предусмотреть возможность объединения двух множеств, вывода на печать элементов множества, а так же метод, определяющий,

принадлежит ли указанное значение множеству. Класс должен содержать методы, позволяющие добавлять и удалять элемент в/из множества. Конструктор должен позволить создавать

объекты с начальной инициализацией. Мощность множества задается при создании объекта.

Реализацию множества осуществить на базе одномерного массива.

Реализовать метод `equals`, выполняющий сравнение объектов данного типа.

Реализация алгоритмов:

```
public class Array{

    private int[] array1;
    int number;
    public Array(int[] array, int number){
        this.array1 = array;
        this.number = number;
    }
    public int[] getArray(){
        return array1;
    }
}
```

```

}
public void print(){
    System.out.println(Arrays.toString(array1));
    System.out.println();
}
public boolean equals(Array array2){
    int[] newArray = array2.getArray();
    if(array1.length != newArray.length){
        return false;
    }
    Arrays.sort(newArray);
    Arrays.sort(array1);

    for(int i = 0; i < array1.length; i++){
        if(array1[i] != newArray[i])
            return false;
    }
    return true;
}
public boolean search(int number){
    for(int i = 0; i < array1.length; i++){
        if(array1[i] == number){
            return true;
        }
    }
    return false;
}
public void concatenation(Array array2){
    int[] newArray2 = array2.getArray();

    int[] newArray = new int[array1.length + newArray2.length];
    System.arraycopy(array1, 0, newArray, 0, array1.length);
    System.arraycopy(newArray2, 0, newArray, array1.length, newArray2.length);
    ;
    Arrays.sort(newArray);
    for(int i = 0 ; i < newArray.length; i++){
        if(i == newArray.length-1 || newArray[i] != newArray[i+1])
            System.out.print(newArray[i] + " ");
    }
}
public int[] removeElement(int index){
    if(index < 0 && index >= array1.length && array1.length != 0){
        System.out.println("Try again!");
        System.exit(0);
    }
    int[] newArray = new int[array1.length - 1];
    for(int i = 0, k = 0; i < array1.length; i++,k++){
        if(i == index){
            k--;
        }
        else {
            newArray[k] = array1[i];
        }
    }
}

```

```

        System.out.println(Arrays.toString(newArray));
        return newArray;
    }
    public int[] addElement(int index, int number){
        if(index < 0 && index >= array1.length){
            System.out.println("Try again!");
            System.exit(0);
        }
        int[] newArray = new int[array1.length + 1];
        for(int i = 0, k = 0; i < array1.length; i++,k++){
            if(k == index){
                newArray[index] = number;
                newArray[index+1] = array1[i];
                k++;
            }
            else{
                newArray[k] = array1[i];
            }
        }
        Array array2 = new Array(inputArray2, count);

//print
        System.out.println();
        System.out.println("First array: ");
        array1.print();
        System.out.println("Second array: ");
        array2.print();
        System.out.println();
//check equal

    }
}
System.out.println(Arrays.toString(newArray));
return newArray;
}
}

```

```

public class Main {

    public static void main(String[] args) {
        int [] inputArray1 ={0,1,3,4,55};
        int [] inputArray2 = {1,4,3,0};
        int count = 7;
        int index1= 3;
        int index2 = 2;
        Array array1 = new Array(inputArray1, count

        if (array1.equals(array2))
            System.out.println("Equal");
        else System.out.println("Not Equal");
        System.out.println();

//find number

```

```

System.out.println("Search in the first array: ");

if(array1.search(count))
    System.out.println("First array contains " + count);
else
    System.out.println("First array doesn't contains " + count);
System.out.println();

System.out.println("Search in the second array: ");
if(array2.search(count))
    System.out.println("Second array contains " + count);
else
    System.out.println("Second array doesn't contains " + count);
System.out.println();

//concatenation
System.out.println("Concatenation:");
array1.concatenation(array2);
System.out.println();
//remove element
System.out.println("Remove element:");
array1.removeElement(index1);
System.out.println();
//add element
System.out.println("Add element:");
array2.addElement(index2, count);
System.out.println();
}
}

```

Пример вывода:

First array:

[0, 1, 3, 4, 55]

Second array:

[1, 4, 3, 0]

Not Equal

Search in the first array:

First array doesn't contains 7

Search in the second array:

Second array doesn't contains 7

Concatenation:

0 1 3 4 55

Remove element:

[0, 1, 3, 55]

Add element:

[1, 4, 7, 3, 0]

Задание 2.

Разработать автоматизированную систему на основе некоторой структуры данных, манипулирующей объектами пользовательского класса.

Реализовать

требуемые функции обработки данных

Требования к выполнению

- **Задание посвящено написанию классов, решающих определенную задачу автоматизации;**
- **Данные для программы загружаются из файла (формат произволен). Файл создать и написать вручную.**

Составить программу, которая содержит текущую информацию о книгах в библиотеке. Сведения о книгах (Book) содержат:

- **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;**

Реализация алгоритма:

```
package com.company;
```

```
import java.io.BufferedReader;  
import java.io.FileReader;  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.Scanner;
```

```
public class Main {  
    public static void main(String[] args) throws IOException {  
        printMenu();  
        ArrayList <Book> books = readInfo("./file.txt");
```

```

Scanner sc = new Scanner(System.in);
while (true) {
    String ch = sc.nextLine();
    switch (ch) {
        case "1":
            AllBooks(books);
            break;
        case "2":

            System.out.println("Enter the year:");
            int year = Integer.parseInt(sc.nextLine());
            BooksOlderThanN(books, year);
            break;
        case "3":
            BooksTakenForReading(books);
            break;
        case "4":
            BooksTakenForReadingWithReaderInformation(books);
            break;
        case "0":
            return;
        default:
            System.out.println("Enter again:");
            continue;
    }
    break;
}

private static void AllBooks (ArrayList<Book> books) {
    for (Book book: books) {
        System.out.println(book.allBooks());
    }
}

private static void BooksTakenForReading (ArrayList<Book> books) {
    for (Book book: books) {
        if (book.isTaken()) {
            System.out.println(book.allBooks());
        }
    }
}

private static void BooksOlderThanN (ArrayList<Book> books, int n) {
    for (Book book: books) {
        if (book.getYear() > n) {
            System.out.println(book.allBooks());
        }
    }
}

private static void BooksTakenForReadingWithReaderInformation (ArrayList<Book> books) {
    for (Book book: books) {
        if (book.isTaken()) {
            System.out.println(book.toString());
        }
    }
}

```

```

}

private static void printMenu() {
    System.out.println("information about all books - 1");
    System.out.println("Books older than n years - 2");
    System.out.println("Books taken for reading - 3");
    System.out.println("Books taken for reading(and reader information) - 4")
    ;
    System.out.println("Exit - 0");
}
private static ArrayList<Book> readInfo(String fileName) throws
    IOException {
    FileReader fr = new FileReader(fileName);
    BufferedReader br = new BufferedReader(fr);
    String sCurrentLine;
    ArrayList<Book> books = new ArrayList<Book>();
    while (((sCurrentLine = br.readLine()) != null) &&
        !sCurrentLine.equals("")) {
        String author = sCurrentLine;
        String title = br.readLine();
        int year = Integer.parseInt(br.readLine());
        int numberOfBooks = Integer.parseInt(br.readLine());
        int numberOfPages = Integer.parseInt(br.readLine());
        String readerFIO = br.readLine();
        String date = br.readLine();
        boolean isTaken = Boolean.parseBoolean(br.readLine());
        Book book = new Book(author, title, year, numberOfBooks, numberOfPages, readerFIO, date,
isTaken);
        books.add(book);
    }
    return books;
}
}

```

```

package com.company;

```

```

class Book{
    private String author;
    private String title;
    private int year;
    private int numberOfBooks;
    private int numberOfPages;
    private String readerFIO;
    private String dateOfDelivery;
    private boolean isTaken;
    public Book(String author, String title, int year, int numberOfBooks, int numberOfPages,String
readerFIO, String date, boolean isTaken){
        this.author = author;
        this.title= title;
        this.year = year;
        this.numberOfBooks = numberOfBooks;
        this.numberOfPages = numberOfPages;
        this.readerFIO = readerFIO;
    }
}

```



```

        this.dateOfDelivery = date;
        this.isTaken = isTaken;
    }
    public int getYear(){
        return year;
    }
    public boolean isTaken() {
        return isTaken;
    }

    public String allBooks(){
        return "Author: " + author + '\n' +
            "Title: " + title + '\n' +
            "Year: " + year + '\n' +
            "number of books: " + numberOfBooks + '\n' +
            "number Of Pages: " + numberOfPages + '\n';
    }
    @Override
    public String toString() {
        String str_taken;
        if (isTaken) str_taken = "Book is taken";
        else str_taken = "Book is not taken";
        return "Author: " + author + '\n' +
            "Title: " + title + '\n' +
            "Year: " + year + '\n' +
            "number of books: " + numberOfBooks + '\n' +
            "number Of Pages: " + numberOfPages + '\n' +
            "reader FIO: " + readerFIO + '\n' +
            "Date of delivery: " + dateOfDelivery + '\n' +
            "Is book Taken: " + str_taken + '\n';
    }
}

```

Пример вывода:

information about all books - 1
 Books older than n years - 2
 Books taken for reading - 3
 Books taken for reading(and reader information) - 4
 Exit - 0

Enter again:

2

Enter the year:

1800

Author: Gogol N.V.

Title: Dead Souls

Year: 1842

number of books: 9

number Of Pages: 300

Author: Bulgakov M.V.

Title: Master and Margarita

Year: 1967

number of books: 11

number Of Pages: 400

Author: Jerom D.S.

Title: over abyss in the rye

Year: 1951

number of books: 5

number Of Pages: 200

Author: Tolstoy L.N.

Title: Anna Karenina

Year: 1877

number of books: 7

number Of Pages: 367

Вывод: Научились создавать и использовать классы в программах на языке программирования Java