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

ОТЧЕТ
по лабораторной работе №4
по дисциплине СПП

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

Проверил:
Крощенко А.А.

Брест 2020

Цель работы: приобрести практические навыки в области объектно-ориентированного проектирования.

Вариант 5

Задание 1
приобрести практические навыки в области объектно-ориентированного проектирования

Реализация алгоритмов:

```
package com.company;

import java.util.ArrayList;

public class Department{
  private String name;
  private int count;
  private class Position{
     String title;
     int salary;
     @Override
     public String toString() {
        return "Position: " + title +
              "\nSalary: " + salary + "$\n" ;
     }
  }
  private class Employee{
     String name;
     String surname;
     int age;
     String employeePosition;
     int employeeSalary;
     @Override
     public String toString() {
        return "Name " + name +
              "\nSurname: " + surname +
              "\nAge: " + age +
              "\nPosition: " + employeePosition +
              "\nSalary: " + employeeSalary + "\n";
     }

  }
  ArrayList<Position> positions = new ArrayList<>();
  public void AddPosition (String _title, int _salary) {
     Position pos = new Position();
     pos.title = _title;
     pos.salary = _salary;
     positions.add(pos);
  }
  public void show () {
     System.out.println("All positions: ");
     System.out.println();
     for (Position pos : positions) {
```

```
            System.out.println(pos.toString());
        }
        System.out.println("All employees: ");
        System.out.println();
        for (Employee employee : employees) {
            System.out.println(employee.toString());
        }
    }
    ArrayList<Employee> employees = new ArrayList<>();
    public void AddEmployee (String _name, String _surname, int _age, String _employeePosition) {
        for (Position pos : positions){
            if((pos.title).equals(_employeePosition)){
                Employee employee = new Employee();
                employee.name = _name;
                employee.surname = _surname;
                employee.age = _age;
                employee.employeePosition = _employeePosition;
                employee.employeeSalary = pos.salary;
                employees.add(employee);
            }
        }
    }
}

public class Main {

    public static void main(String[] args) {
        Department department = new Department();
        department.AddPosition("programmer", 2000);
        department.AddPosition("HR", 1000);
        department.AddEmployee("Katya", "Gavrilkovich", 23, "programmer");
        department.AddEmployee("Ivan", "Petrov", 30, "HR");
        department.AddEmployee("Ivan", "Serggev", 30, "Manager");
        department.AddEmployee("Dmitriy", "Petrov", 30, "HR");
        department.show();
    }
}
```

Пример вывода:

All positions:

Position: programmer
Salary: 2000$

Position: HR
Salary: 1000$

All employees:

Name Katya
Surname: Gavrilkovich
Age: 23

Position: programmer
Salary: 2000

Name Ivan
Surname: Petrov
Age: 30
Position: HR
Salary: 1000

Name Dmitriy
Surname: Petrov
Age: 30
Position: HR
Salary: 1000

Задание 2.
Создать класс Абзац, используя класс Слово.

Реализация алгоритма:

```
public class Main {
  public static void main(String[] args) {
     Word word1 = new Word("My");
     Word word2 = new Word("name");
     Word word3 = new Word("is");
     Word word4 = new Word("Katya");

     Paragraph paragraph1 = new Paragraph();
     paragraph1.addWord(word1);
     paragraph1.addWord(word2);
     paragraph1.addWord(word3);
     paragraph1.addWord(word4);
     paragraph1.printAll();
  }

}
class Word{
  private String word;
  public Word(String word){
     this.word = word;
  }
  public String getWord(){
     return word;
  }

}
class Paragraph{
  ArrayList<Word> words = new ArrayList<>();
  public void addWord(Word word){
     words.add(word);
```

```
  }
  public void printAll(){
     for(int i = 0 ; i < words.size(); i++)
        System.out.print(words.get(i).getWord() + " ");
  }
}
```

Пример вывода:

Задание 3
Система Библиотека. Читатель оформляет Заказ на Книгу. Система осуществляет поиск в Каталоге. Библиотекарь выдает Читателю Книгу на абонемент или в читальный зал. При невозвращении Книги Читателем он может быть занесен Администратором в «черный список».

Реализация алгоритмов:

```
package com.company;
import java.util.*;
import java.text.*;

public class Main {

  public static void main(String[] args) {
     String[] bookNames = { "Master and Margarita", "Dead Souls", "Anna Karenina" };
          Library library = new Library();
//create books
     Book[] books = { new Book(bookNames[0], "Bulgakov", 3),
          new Book(bookNames[1], "Gogol", 5),
          new Book(bookNames[2], "Tolstoy", 1) };
//create readers
     Reader[] readers = { new Reader("Ivan Ivanov", "24.05.2019"),
          new Reader("Ivan Petrov", "04.12.2019"),
          new Reader("Ivan Sergeev", "01.12.2019") ,
          new Reader("Ivan Semenov", "05.12.2019")};
//add books in arrayList

     for (Book book : books) {
        library.addBook(book);
     }
//add readers in arrayList
     for (Reader reader : readers) {
        library.addReader(reader);
     }
     library.showAll();
     System.out.println("\nTry to add Bulgakov to Ivan Ivanov\n");
     library.createOrder(readers[0], books[0]);
     System.out.println("\nTry to add Gogol to Ivan Ivanov\n");
     library.createOrder(readers[0], books[1]);

     System.out.println("\nTry to add Tolstoi to Ivan Petrov\n");
```

```
        library.createOrder(readers[1], books[2]);

        System.out.println("\nTry to add Tolstoi to Ivan Sergeev\n");

        library.createOrder(readers[2], books[2]);
        System.out.println("\nTry to add Gogol to Ivan Sergeev\n");
        library.createOrder(readers[2], books[1]);
        System.out.println("\nTry to add Tolstoi to Ivan Semenov\n");
        library.createOrder(readers[3], books[2]);
        for(Reader reader: readers){
            System.out.println("Reader name: " + reader.getName()
                    + "\nDate: " + reader.getDate() + "\n");
        }
        library.showBlackList();
    }
    }
        interface Show{
            void showAll();
        }
        class Library implements Show{
            private ArrayList<Book> books = new ArrayList<>();
            private ArrayList<Reader> readers = new ArrayList<>();
            private ArrayList<Reader> blackList = new ArrayList<>();
            private ArrayList<Order> orders = new ArrayList<>();
            public void addBook(Book book) {
                books.add(book);
            }
            public void createOrder(Reader reader, Book book) {
                if(reader.checkDate(new Date(), reader.getDate()) > 20){
                    blackList.add(reader);
                    return;
                }
                for(Book currentBook : books){
                    if(currentBook.getId() == book.getId() && currentBook.getNumber() > 0
                    ){
                        Order order = new Order(book.getId(), reader.getId(), book.getTitle());
                        orders.add(order);

                        removeBook(currentBook);
                        reader.addBook(currentBook);
    //removeOrder(order);
                    }
                }

            }
            public void showBlackList(){
                System.out.println("\nBlack list\n");
                for(Reader reader: blackList){
                    System.out.println("\nReader name: " + reader.getName()
                            + "\nDate: " + reader.getDate());
                }
            }
            public void showAll(){
                System.out.println("\nAll books in the library:\n");
```

```
        for(Book book: books){
            System.out.println("\nBooks title: " + book.getTitle()
                    + "\nAuthor: " + book.getAuthor()
                    + "\nCount: " + book.getNumber() + "\n");
        }
    }
    public void removeBook(Book book){
        book.setNumber(book.getNumber()-1);
    }
    public void removeOrder(Order order){
        orders.remove(order);
    }
    public void addReader(Reader reader){
        readers.add(reader);
    }
}
class Book {
    private int id;
    private String title;
    private String author;
    private int number;
    private static int booksCount = 1;
    public Book(String title, String author, int number) {
        id = booksCount++;

        setTitle(title);
        setAuthor(author);
        setNumber(number);
    }
    public int getId() {
        return id;
    }
    public int getNumber() {
        return number;
    }
    public void setNumber(int number) {
        this.number = number;
    }
    public String getAuthor() {
        return author;
    }
    public void setAuthor(String author) {
        this.author = author;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
}
class Reader implements Show{
    private int id;
    private String name;
```

```
    private Date bookDate = null;
    private ArrayList<Book>books = new ArrayList<>();
    private static int readersCount = 1;
    public Reader(String name, String date) {
        id = readersCount++;
        setName(name);
        setDate(date);
    }

    public int getId() {
        return id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void setDate(String date){
        SimpleDateFormat format = new SimpleDateFormat("dd.MM.yyyy");
        try{
            bookDate = format.parse(date);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    public Date getDate(){
        return bookDate;
    }
    public long checkDate(Date date, Date bookDate){
        long difference = date.getTime() - bookDate.getTime();
        long days = difference / (24 * 60 * 60 * 1000);
        return days;
    }
    public void addBook(Book book){
        books.add(book);
    }
    public void showAll(){
        for(Book book: books){
            if(books.size() == 0) {System.out.println("There is not any books");}
            else {
                System.out.println("Book name: " + book.getTitle() + "\n" +
                        "Author: " + book.getAuthor() + "\n");
            }
        }
    }
}
class Order {
    private int id;
    private int bookId;
    private int readerId;
    private String bookTitle;

    private static int ordersCount = 1;
```

```
    public Order(int bookId, int readerId, String bookTitle) {
        id = ordersCount++;
        setBookId(bookId);
        setReaderId(readerId);
        setBookTitle(bookTitle);
    }
    public int getReaderId() {
        return readerId;
    }
    public void setReaderId(int readerId) {
        this.readerId = readerId;
    }
    public int getBookId() {
        return bookId;
    }
    public void setBookId(int bookId) {
        this.bookId = bookId;
    }
    public void setBookTitle(String bookTitle){
        this.bookTitle = bookTitle;
    }
    public String getBookTitle(){
        return bookTitle;
    }
}
```

Пример вывода:

All books in the library:

Books title: Master and Margarita
Author: Bulgakov
Count: 3

Books title: Dead Souls
Author: Gogol
Count: 5

Books title: Anna Karenina
Author: Tolstoy
Count: 1

Try to add Bulgakov to Ivan Ivanov

Try to add Gogol to Ivan Ivanov

Try to add Tolstoi to Ivan Petrov

Try to add Tolstoi to Ivan Sergeev

Try to add Gogol to Ivan Sergeev

Try to add Tolstoi to Ivan Semenov

Reader name: Ivan Ivanov
Date: Fri May 24 00:00:00 MSK 2019

Reader name: Ivan Petrov
Date: Wed Dec 04 00:00:00 MSK 2019

Reader name: Ivan Sergeev
Date: Sun Dec 01 00:00:00 MSK 2019

Reader name: Ivan Semenov
Date: Thu Dec 05 00:00:00 MSK 2019

Black list

Reader name: Ivan Ivanov
Date: Fri May 24 00:00:00 MSK 2019

Reader name: Ivan Ivanov
Date: Fri May 24 00:00:00 MSK 2019

Reader name: Ivan Petrov
Date: Wed Dec 04 00:00:00 MSK 2019

Reader name: Ivan Sergeev
Date: Sun Dec 01 00:00:00 MSK 2019

Reader name: Ivan Sergeev
Date: Sun Dec 01 00:00:00 MSK 2019

Reader name: Ivan Semenov
Date: Thu Dec 05 00:00:00 MSK 2019

Process finished with exit code 0

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