

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ
Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №4

Выполнил

Литвинко В.А.

студент группы АС-50

Проверил

Крощенко А. А.

ст. преп. кафедры ИИТ

Брест 2020

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

ВАРИАНТ 5

Задание 1

5) Создать класс Department (отдел фирмы) с внутренним классом, с помощью объектов которого можно хранить информацию обо всех должностях отдела и обо всех сотрудниках, когда-либо занимавших конкретную должность.

Код программы:

```
package com.company;

public class Main {

    public static void main(String[] args) {
        Department department = new Department();

        System.out.println("Все должности:\n");
        department.showFunctions();

        department.addWorker(1, "Селиванов П.Ю.", 1);
        department.addWorker(2, "Круглова Ю.Я.", 2);
        department.addWorker(3, "Косой М.М.", 3);
        department.addWorker(4, "Белова А.Е.", 4);
        department.addWorker(5, "Горький И.И.", 4);

        department.dismissWorker(4);

        System.out.println("Все работники:\n");

        department.showWorkers();
    }
}
```

```
package com.company;

import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;

public class Department {

    class Function{
        private int id;
        private String name;

        public Function(int id, String name){
            this.id = id;
            this.name = name;
        }

        public String toString(){
            return "id="+id+", должность="+name;
        }
    }
}
```

```

class Worker{
    private int id;
    private String fio;
    private Function function;
    private LocalDate dateWorkingStart;
    private LocalDate dateWorkingEnd = null;

    public Worker(int id, String fio, Function function){
        this.id = id;
        this.fio = fio;
        this.function = function;
        this.dateWorkingStart = LocalDate.now();
    }

    public String toString(){
        StringBuilder builder = new StringBuilder();
        builder.append("id: "+id+", фιο: "+fio+", должность: "+function.name+", статус: ");
        if(dateWorkingEnd==null) {
            builder.append("работает+", " дата трудоустройства: "
+dateWorkingStart.format(DateTimeFormatter.ofPattern("dd/MM/yyyy")));
        }
        else
            builder.append("уволен+", " дата трудоустройства: "
+dateWorkingStart.format(DateTimeFormatter.ofPattern("dd/MM/yyyy"))+"", " дата
увольнения: "
+dateWorkingEnd.format(DateTimeFormatter.ofPattern("dd/MM/yyyy")));

        return builder.toString();
    }
}

ArrayList<Function> functions = new ArrayList<Function>();
ArrayList<Worker> workers = new ArrayList<Worker>();

public Department(){
    functions.add(new Function(1, "начальник"));
    functions.add(new Function(2, "заместитель начальника"));
    functions.add(new Function(3, "бухгалтер"));
    functions.add(new Function(4, "работник"));
}

public void showFunctions(){
    for(int i=0; i<functions.size(); i++)
        System.out.println(functions.get(i).toString());
}

public void showWorkers(){
    for(int i=0; i<workers.size(); i++)
        System.out.println(workers.get(i).toString());
}

public void addWorker(int workerId, String fio, int functionId){
    Function function = null;
    for(int i=0; i<functions.size(); i++)
        if(functions.get(i).id==functionId)
        {
            function = functions.get(i);
            break;
        }
}

```

```

        if(function==null)
            return;

        Worker worker = new Worker(workerId, fio, function);
        workers.add(worker);
    }

    public void dismissWorker(int workerId){
        for(int i=0; i<workers.size(); i++)
            if(workers.get(i).id==workerId)
            {
                workers.get(i).dateWorkingEnd = LocalDate.now();
                return;
            }
    }
}

```

Вывод:

```

id=1, должность=начальник
id=2, должность=заместитель начальника
id=3, должность=бухгалтер
id=4, должность=работник
Все работники:

id: 1, фео: Селиванов П.Ю., должность: начальник, статус: работает, дата трудоустройства: 01/12/2020
id: 2, фео: Круглова Ю.Я., должность: заместитель начальника, статус: работает, дата трудоустройства: 01/12/2020
id: 3, фео: Косой М.М., должность: бухгалтер, статус: работает, дата трудоустройства: 01/12/2020
id: 4, фео: Белова А.Е., должность: работник, статус: уволен, дата трудоустройства: 01/12/2020, дата увольнения: 01/12/2020
id: 5, фео: Горький И.И., должность: работник, статус: работает, дата трудоустройства: 01/12/2020

```

Задание 2

5) Реализовать агрегирование. При создании класса агрегируемый класс объявляется как атрибут (локальная переменная, параметр метода). Включить в каждый класс 2-3 метода на выбор. Продемонстрировать использование разработанных классов.

Создать класс Абзац, используя класс Слово.

Код программы:

```

package com.company;

import java.util.ArrayList;

public class Indent {

    ArrayList<Word> words = new ArrayList<Word>();
    int margin = 5;

    public void setMargin(int margin){
        this.margin = margin;
    }

    public boolean addWord(String word){
        Word temp = new Word(word);
        if(!temp.checkIsCorrect())
            return false;
    }
}

```

```

        words.add(temp);
        return true;
    }

    public void print() {
        if (words.size() == 0)
            System.out.println("Пустой абзац");
        else
        {
            for (int i = 0; i < margin; i++)
                System.out.print(" ");
            for (int i = 0; i < words.size(); i++) {
                words.get(i).print();
                System.out.print(" ");
            }
            System.out.println();
        }
    }
}

```

```

package com.company;

public class Main {

    public static void main(String[] args) {
        Indent indent = new Indent();
        System.out.println(indent.addWord("Получился"));
        System.out.println(indent.addWord("это слово не вставится"));
        System.out.println(indent.addWord("тестовый"));
        System.out.println(indent.addWord("абзац"));
        indent.print();
    }
}

```

```

package com.company;

public class Word {

    String word;

    public Word(String word) {
        this.word = word;
    }

    public void setWord(String word) {
        this.word = word;
    }

    public void print() {
        System.out.print(word);
    }

    public boolean checkIsCorrect() {
        String[] temp = word.split(" ");
        if (temp.length == 1)
            return true;
        else
            return false;
    }
}

```

Вывод:

```
C:\Users\Влад (.jdk\openjdk-15.0.1\bin\java.  
true  
false  
5 true  
2 true  
Получился тестовый абзац
```

Задание 3

5) 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».

Код программы:

```
package com.company;  
  
public class Main {  
  
    public static void main(String[] args) {  
        Admin administrator = new Admin(0, "Иванов И.И.");  
  
        Catalog catalog = new Catalog();  
  
        Book book1 = new Book(5, "Сказка о рыбаке и рыбке", "Пушкин");  
        Book book2 = new Book(10, "Война и мир", "Толстой");  
        Book book3 = new Book(12, "Горе от ума", "Грибоедов");  
        catalog.addBook(book1);  
        catalog.addBook(book2);  
        catalog.addBook(book3);  
  
        Librarian librarian = new Librarian(1, "Светлый А.Е.", catalog);  
  
        catalog.show();  
  
        Reader reader1 = new Reader(987, "Вертьянова М.И");  
  
        Order order1 = reader1.makeOrder(34, 10, 0);  
        Order order2 = reader1.makeOrder(11, 12, 1);  
  
        Reader reader2 = new Reader(584, "Горшков А.М.");  
  
        Order order3 = reader1.makeOrder(15, 5, 0);  
  
        System.out.println("\n\n\n");  
  
        if(!librarian.giveBookToReader(reader1, order1))  
            System.out.println("order1 не удалось выдать reader1");  
  
        if(!librarian.giveBookToReader(reader1, order2))  
            System.out.println("order2 не удалось выдать reader1");  
  
        if(!librarian.giveBookToReader(reader2, order3))  
            System.out.println("order3 не удалось выдать reader2");  
    }  
}
```

```

        librarian.takeReturnedBook(reader1, book2);

        administrator.addToBlacklist(reader2);

        Order order4 = reader2.makeOrder(357, 10, 1);

        if(!librarian.giveBookToReader(reader2, order4))
            System.out.println("order4 не удалось выдать reader2");

        System.out.println("\n\n\n");

        catalog.show();

        System.out.println("\n\n\nReader1:");

        reader1.show();

        System.out.println("\n\n\nReader2:");

        reader2.show();
    }
}

```

Вывод:

```

Каталог:
id: 5; name: Сказка о рыбаке и рыбке; author: Пушкин
id: 10; name: Война и мир; author: Толстой
id: 12; name: Горе от ума; author: Грибоедов

order4 не удалось выдать reader2

Каталог:
id: 10; name: Война и мир; author: Толстой

Reader1:
id: 987; fio: Вертьянова М.И; in blacklist: false
Взятые книги:
id: 12; name: Горе от ума; author: Грибоедов; тип выдачи: читальный зал

Reader2:
id: 584; fio: Горшков А.М.; in blacklist: true
Взятые книги:
id: 5; name: Сказка о рыбаке и рыбке; author: Пушкин; тип выдачи: абонемент

```

Вывод: приобрел практические навыки в области объектно-ориентированного проектирования