

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”
КАФЕДРА ИИТ

ОТЧЁТ
по лабораторной работе №5

Выполнил:
студент 3 курса
группы ПО-9
Оводок В.В.

Проверил:
Крощенко А.А.

Цель:приобрести практические навыки в области объектно-ориентированного проектирования

Вариант 5

Задание 1

Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов:

interface Здание ← abstract class Общественное Здание ← class Театр.

Выходные данные:

```
Здание открыто
Проходит репетиция
Здание закрыто
```

Краткий текст программы:

```
interface Building{
    void Open();
    void Close();
}
abstract class PublicBuilding implements Building{
    @Override
    public void Open() {
        System.out.println("Здание открыто");
    }
    @Override
    public void Close() {
        System.out.println("Здание закрыто");
    }
}
class Theater extends PublicBuilding{
    public void doEvent(){
        System.out.println("Проходит репетиция");
    }
}
public class Task1 {
    public static void main(String[] args) {
        Theater theater = new Theater();
        theater.Open();
        theater.doEvent();
        theater.Close();
    }
}
```

Задание 2

В следующих заданиях требуется создать суперкласс (абстрактный класс, интерфейс) и определить общие методы для данного класса. Создать подклассы, в которых добавить специфические свойства и методы. Часть

методов переопределить. Создать массив объектов суперкласса и заполнить объектами подклассов. Объекты подклассов идентифицировать конструктором по имени или идентификационному номеру. Использовать объекты подклассов для моделирования реальных ситуаций и объектов.

Создать абстрактный класс Работник фирмы и подклассы Менеджер, Аналитик, Программист, Тестировщик, Дизайнер, Бухгалтер. Реализовать логику начисления зарплаты.

Выходные данные:

```
Я Vadim менеджер, и я разделяю и властвую
Бухгалтер Vlad рассчитал зп сотрудника Vadim
Его зарплата = 1000.0
Я Vlad бухгалтер, и я устал заполнять бумаги
Бухгалтер Vlad рассчитал зп сотрудника Vlad
Его зарплата = 700.0
Я Dima программист, и я кроКОДИЛ, кроКОЖУ и буду кроКОДИТЬ
Бухгалтер Vlad рассчитал зп сотрудника Dima
Его зарплата = 900.0
Я Maxim тестировщик, и я... яая...эхх ладно....
Бухгалтер Vlad рассчитал зп сотрудника Maxim
Его зарплата = 700.0
Я Roma аналитик, и я хочу домой
Бухгалтер Vlad рассчитал зп сотрудника Roma
Его зарплата = 800.0
Я Mike дизайнер, и программист меня любит
Бухгалтер Vlad рассчитал зп сотрудника Mike
Его зарплата = 800.0
Менеджер Vadim награждает сотрудника Vlad
Менеджер Vadim награждает сотрудника Mike
Бухгалтер Vlad рассчитал зп сотрудника Vlad
Его зарплата = 800.0
Бухгалтер Vlad рассчитал зп сотрудника Mike
Его зарплата = 1000.0
```

Краткий текст программы:

```
public static void main(String[] args) {
```

```
    Manager manager = new Manager("Vadim", 1, 1000);
    Booker booker = new Booker("Vlad", 2, 700);
    Programmer programmer = new Programmer("Dima", 3, 900);
    Tester tester = new Tester("Maxim", 4, 700);
    Analyst analyst = new Analyst("Roma", 5, 800);
    Designer designer = new Designer("Mike", 6, 800);
```

```
    List<Employee> employees = new ArrayList<>();
```

```
    employees.add(manager);
    employees.add(booker);
    employees.add(programmer);
    employees.add(tester);
    employees.add(analyst);
    employees.add(designer);
```

```
    for (Employee employee:employees){
```

```

        employee.doWork();
        booker.calculateSalary(employee);
    }
    manager.awarding(booker, 100);
    manager.awarding(designer, 200);

    booker.calculateSalary(booker);
    booker.calculateSalary(designer);

}

```

Задание 3

Построить модель программной системы с применением отношений (обобщения, агрегации, ассоциации, реализации) между классами. Задать атрибуты и методы классов. Реализовать (если необходимо) дополнительные классы. Продемонстрировать работу разработанной системы.

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».

Заменить объявления суперклассов объявлениями абстрактных классов или интерфейсов, где это возможно.

Выходные данные:

```

Книга Заводной апельсин выдана читателю Вадим
Книга 1984 выдана читателю Влад
Книга 451 градус выдана читателю Дима
Книга 451 градус находится в прокате Вадим не может взять эту книгу
Книга Заводной апельсин возвращена читателем Вадим
Книга 1984 возвращена читателем Влад
Влад добавлен в черный список
Книга 451 градус возвращена читателем Дима
Читатель Влад в черном списке и не может взять книгу 451 градус
Книга 451 градус выдана читателю Вадим

```

Краткий текст программы:

```

public static void main(String[] args) {
    Library library = new Library();

    Book book1 = new Book("Заводной апельсин", "Энтони Берджес");
    Book book2 = new Book("1984", "Джордж Оруэлл");
    Book book3 = new Book("451 градус", "Рэй Бредбери");
}

```

```

Book book4 = new Book("О дивный новый мир", "Олдос Хаксли");

Catalog catalog = new Catalog();

catalog.addBook(book1);
catalog.addBook(book2);
catalog.addBook(book3);
catalog.addBook(book4);

Reader reader1 = new Reader("Вадим");
Reader reader2 = new Reader("Влад");
Reader reader3 = new Reader("Дима");

Order order1 = reader1.placeOrderForReadingRoom(book1);
Order order2 = reader2.placeOrderForHome(book2, LocalDate.of(2024, 3, 1));
Order order3 = reader3.placeOrderForHome(book3, LocalDate.of(2024, 5, 5));
Order order4 = reader1.placeOrderForReadingRoom(book3);

LibraryWorker libraryWorker = new LibraryWorker(library);

libraryWorker.addOrder(order1);
libraryWorker.addOrder(order2);
libraryWorker.addOrder(order3);
libraryWorker.addOrder(order4);

libraryWorker.issueBookToReader(order1);
libraryWorker.issueBookToReader(order2);
libraryWorker.issueBookToReader(order3);
libraryWorker.issueBookToReader(order4);

reader1.returnBook(book1, libraryWorker);
reader2.returnBook(book2, libraryWorker);
reader3.returnBook(book3, libraryWorker);

Order order5 = reader2.placeOrderForReadingRoom(book3);
libraryWorker.issueBookToReader(order5);
libraryWorker.issueBookToReader(order4);
}

```

Вывод: приобрел практические навыки в области объектно-ориентированного проектирования.