МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №5

Специальность ПО5

Выполнил:
А.А. Игнатюк,
студент группы ПО-5

Проверил:
А.А. Крощенко,
ст. преп. кафедры ИИТ,
«__» _____ 2021 г.

Брест 2021

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

Вариант 5.

Задание 1.

Требования к выполнению:

• Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм.

5) interface Здание ← abstract class Общественное Здание ← class Театр.

Спецификация ввода:

-

Спецификация вывода:

<параметры функций System.out.println() (содержимое полей объектов)>

…

Структура проекта:

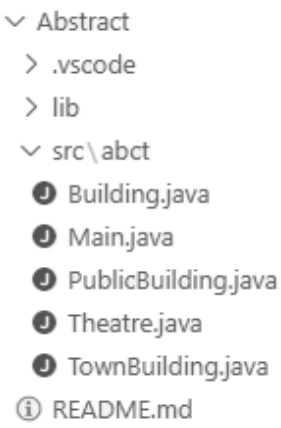

Рисунок 1.1 - Структура проекта.

Код программы:

Building.java

Java > Abstract > src > abct > Building.java > ...

```
1   package abct;
2
3   public interface Building {
4       public void f_create();
5
6       public void f_destroy();
7
8       public void f_open();
9
10      public void f_close();
11  }
12
```

Рисунок 1.2 - Содержимое файла Building.java.

Main.java

Java > Abstract > src > abct > Main.java > ...

```java
package abct;

public class Main {
    Run | Debug
    public final static void main(final String[] c_Args) throws Exception {
        Theatre v_Theater = new Theatre(new String("95 Hilltop Drive"), new String("20th Century Theatre"), 200, 34.99,
                new String("15:00 - 00:00"));

        System.out.println(new String("Name: ") + v_Theater.f_get_name());
        System.out.println(new String("Address: ") + v_Theater.f_get_address());
        System.out.println(new String("Visitors count: ") + v_Theater.f_get_visitors_count());
        System.out.println(new String("Visit price: ") + v_Theater.f_get_visit_price());
        System.out.println(new String("Work time: ") + v_Theater.f_get_work_time());

        System.out.println();
        v_Theater.f_show_performance();
    }
}

```

Рисунок 1.3 - Содержимое файла Main.java.

Theatre.java

Java > Abstract > src > abct > Theatre.java > ...

```java
package abct;

public final class Theatre extends PublicBuilding {
    public Theatre(final String c_Name, final String c_Address) {
        super(c_Name, c_Address);
    }

    public Theatre(final String c_Name, final String c_Address, final Integer c_VisitorsCount) {
        super(c_Name, c_Address, c_VisitorsCount);
    }

    public Theatre(final String c_Name, final String c_Address, final Integer c_VisitorsCount,
            final Double c_VisitPrice) {
        super(c_Name, c_Address, c_VisitorsCount, c_VisitPrice);
    }

    public Theatre(final String c_Name, final String c_Address, final Integer c_VisitorsCount,
            final Double c_VisitPrice, final String c_WorkTime) {
        super(c_Name, c_Address, c_VisitorsCount, c_VisitPrice, c_WorkTime);
    }

    public final void f_show_performance() {
        System.out.println("Very interesting performance...");
    }
}

```

Рисунок 1.4 - Содержимое файла Theatre.java.

PublicBuilding.java ×

Java > Abstract > src > abct > PublicBuilding.java > ...

```java
package abct;

public abstract class PublicBuilding extends TownBuilding {
    private Integer m_VisitorsCount = 0;
    private Double m_VisitPrice = 0.0;
    private String m_WorkTime = new String("00:00 - 00:00");

    public PublicBuilding(final String c_Name, final String c_Address) {
        super(c_Name, c_Address);
    }

    public PublicBuilding(final String c_Name, final String c_Address, final Integer c_VisitorsCount) {
        super(c_Name, c_Address);
        this.m_VisitorsCount = c_VisitorsCount;
    }

    public PublicBuilding(final String c_Name, final String c_Address, final Integer c_VisitorsCount,
            final Double c_VisitPrice) {
        super(c_Name, c_Address);
        this.m_VisitorsCount = c_VisitorsCount;
        this.m_VisitPrice = c_VisitPrice;
    }

    public PublicBuilding(final String c_Name, final String c_Address, final Integer c_VisitorsCount,
            final Double c_VisitPrice, final String c_WorkTime) {
        super(c_Name, c_Address);
        this.m_VisitorsCount = c_VisitorsCount;
        this.m_VisitPrice = c_VisitPrice;
        this.m_WorkTime = c_WorkTime;
    }

    public final String f_get_work_time() {
        return this.m_WorkTime;
    }

    public final void f_set_work_time(final String c_WorkTime) {
        this.m_WorkTime = c_WorkTime;
    }

    public final Integer f_get_visitors_count() {
        return this.m_VisitorsCount;
    }

    public final void f_set_visitors_count(final Integer c_VisitorsCount) {
        this.m_VisitorsCount = c_VisitorsCount;
    }

    public final Double f_get_visit_price() {
        return this.m_VisitPrice;
    }

    public final void f_set_visit_price(final Double c_VisitPrice) {
        this.m_VisitPrice = c_VisitPrice;
    }
}

```

Рисунок 1.5 - Содержимое файла PublicBuilding.java.

TownBuilding.java

Java > Abstract > src > abct > TownBuilding.java > ...

```java
package abct;

public abstract class TownBuilding implements Building {
    private Boolean m_IsCreated = Boolean.FALSE, m_IsOpen = Boolean.FALSE;
    private String m_Name = new String(), m_Address = new String();

    public TownBuilding(final String c_Name, final String c_Address) {
        this.m_Name = c_Name;
        this.m_Address = c_Address;
    }

    public final void f_set_name(final String c_Name) {
        this.m_Name = c_Name;
    }

    public final String f_get_name() {
        return this.m_Name;
    }

    public final void f_set_adress(final String c_Address) {
        this.m_Address = c_Address;
    }

    public final String f_get_address() {
        return this.m_Address;
    }

    public final void f_create() {
        this.m_IsCreated = Boolean.TRUE;
    }

    public final void f_destroy() {
        this.m_IsCreated = Boolean.FALSE;
    }

    public final Boolean f_is_created() {
        return this.m_IsCreated;
    }

    public final void f_open() {
        this.m_IsOpen = Boolean.TRUE;
    }

    public final void f_close() {
        this.m_IsOpen = Boolean.FALSE;
    }

    public final Boolean f_is_open() {
        return this.m_IsOpen;
    }
}

```

Рисунок 1.6 - Содержимое файла TownBuilding.java.

```
Very interesting performance...
PS C:\Users\User\Documents\Visual Studio Code>  c:; cd 'c:\Users\User\Documents\Visual
 Studio Code'; & 'c:\Users\User\.vscode\extensions\vscjava.vscode-java-debug-0.36.0\sc
ripts\launcher.bat' 'C:\Program Files\Eclipse Foundation\jdk-11.0.12.7-hotspot\bin\jav
a.exe' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspac
eStorage\35645f84fe309ce1d5ab2a9af314d21a\redhat.java\jdt_ws\Visual Studio Code_9125e1
93\bin' 'abct.Main'
Name: 95 Hilltop Drive
Address: 20th Century Theatre
Visitors count: 200
Visit price: 34.99
Work time: 15:00 - 00:00

Very interesting performance...
PS C:\Users\User\Documents\Visual Studio Code>
```

Рисунок 1.7 - Результат работы программы.

Задание 2.

Требования к выполнению:

• 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 его объектами подклассов. Объекты подклассов идентифицировать конструктором по имени или идентификационному номеру.

• Использовать объекты подклассов для моделирования реальных ситуаций и объектов.

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

Спецификация ввода:

-

Спецификация вывода:

<параметры функций System.out.println() (содержимое полей объектов)>

…

Структура проекта:

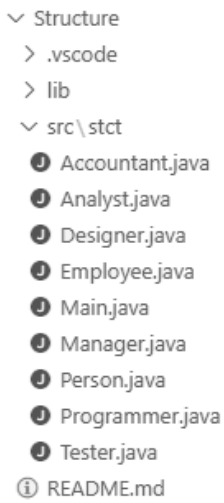

Рисунок 2.1 - Структура проекта.

Код программы:

Accountant.java ×

Java > Structure > src > stct > Accountant.java > ...

```
1   package stct;
2
3   public final class Accountant extends Employee {
4       public Accountant(final Integer c_Age, final String c_Name, final Double c_MinimalSalary) {
5           super(c_Age, c_Name, c_MinimalSalary);
6       }
7
8       public Accountant(final Integer c_Age, final String c_Name, final Double c_MinimalSalary, final Double c_SalaryK) {
9           super(c_Age, c_Name, c_MinimalSalary, c_SalaryK);
10      }
11
12      public final void f_work() {
13          if (this.f_is_working()) {
14              System.out.println("The accountant " + this.f_get_name() + " is working...");
15              return;
16          }
17
18          System.out.println("Accountant " + this.f_get_name() + " does not work...");
19      }
20  }
21
```

Рисунок 2.2 - Содержимое файла Accountant.java.

Analyst.java ×

Java > Structure > src > stct > Analyst.java > ...

```
1   package stct;
2
3   public final class Analyst extends Employee {
4       public Analyst(final Integer c_Age, final String c_Name, final Double c_MinimalSalary) {
5           super(c_Age, c_Name, c_MinimalSalary);
6       }
7
8       public Analyst(final Integer c_Age, final String c_Name, final Double c_MinimalSalary, final Double c_SalaryK) {
9           super(c_Age, c_Name, c_MinimalSalary, c_SalaryK);
10      }
11
12      public final void f_work() {
13          if (this.f_is_working()) {
14              System.out.println("The analyst " + this.f_get_name() + " is working...");
15              return;
16          }
17
18          System.out.println("Analyst " + this.f_get_name() + " does not work...");
19      }
20  }
21
```

Рисунок 2.3 - Содержимое файла Analyst.java.

Designer.java

Java > Structure > src > stct > Designer.java > ...

```java
package stct;

public final class Designer extends Employee {
    public Designer(final Integer c_Age, final String c_Name, final Double c_MinimalSalary) {
        super(c_Age, c_Name, c_MinimalSalary);
    }

    public Designer(final Integer c_Age, final String c_Name, final Double c_MinimalSalary, final Double c_SalaryK) {
        super(c_Age, c_Name, c_MinimalSalary, c_SalaryK);
    }

    public final void f_work() {
        if (this.f_is_working()) {
            System.out.println("The designer " + this.f_get_name() + " is working...");
            return;
        }

        System.out.println("Designer " + this.f_get_name() + " does not work...");
    }
}

```

Рисунок 2.4 - Содержимое файла Designer.java.

Manager.java

Java > Structure > src > stct > Manager.java > ...

```java
package stct;

public final class Manager extends Employee {
    public Manager(final Integer c_Age, final String c_Name, final Double c_MinimalSalary) {
        super(c_Age, c_Name, c_MinimalSalary);
    }

    public Manager(final Integer c_Age, final String c_Name, final Double c_MinimalSalary, final Double c_SalaryK) {
        super(c_Age, c_Name, c_MinimalSalary, c_SalaryK);
    }

    public final void f_work() {
        if (this.f_is_working()) {
            System.out.println("The manager " + this.f_get_name() + " is working...");
            return;
        }

        System.out.println("Manager " + this.f_get_name() + " does not work...");
    }
}

```

Рисунок 2.5 - Содержимое файла Manager.java.

Employee.java ×

Java > Structure > src > stct > Employee.java > ...

```java
package stct;

public abstract class Employee extends Person {
    private Boolean m_IsWorking = Boolean.FALSE;
    private Double m_MinimalSalary = 0.0, m_SalaryK = 1.0, m_Salary = 0.0;

    public Employee(final Integer c_Age, final String c_Name, final Double c_MinimalSalary) {
        super(c_Age, c_Name);
        this.m_MinimalSalary = c_MinimalSalary;
        this.m_Salary = this.m_MinimalSalary * this.m_SalaryK;
    }

    public Employee(final Integer c_Age, final String c_Name, final Double c_MinimalSalary, final Double c_
        super(c_Age, c_Name);
        this.m_MinimalSalary = c_MinimalSalary;
        this.m_SalaryK = c_SalaryK;
        this.m_Salary = this.m_MinimalSalary * this.m_SalaryK;
    }

    public final void f_start_working() {
        this.m_IsWorking = Boolean.TRUE;
    }

    public final void f_stop_working() {
        this.m_IsWorking = Boolean.FALSE;
    }

    public final Boolean f_is_working() {
        return this.m_IsWorking;
    }

    public final void f_set_minimal_salary(final Double c_MinimalSalary) {
        this.m_MinimalSalary = c_MinimalSalary;
        this.m_Salary = this.m_MinimalSalary * this.m_SalaryK;
    }

    public final Double f_get_minimal_salary() {
        return this.m_MinimalSalary;
    }

    public final void f_set_salary_k(final Double c_SalaryK) {
        this.m_SalaryK = c_SalaryK;
        this.m_Salary = this.m_MinimalSalary * this.m_SalaryK;
    }

    public final Double f_get_salary_k() {
        return this.m_SalaryK;
    }

    public final Double f_get_salary() {
        return this.m_Salary;
    }

    public abstract void f_work();
}
```

Рисунок 2.6 - Содержимое файла Employee.java.

Main.java

Java > Structure > src > stct > Main.java > ...

```java
package stct;

import java.lang.Math;
import java.util.HashMap;
import java.util.Map;
import java.util.Vector;

public final class Main {
    public static enum Speciality {
        MANAGER(0), ANALYST(1), PROGRAMMER(2), TESTER(3), DESIGNER(4), ACCOUNTANT(5);

        private static Map<Integer, Speciality> m_Map = new HashMap<Integer, Speciality>();

        static {
            for (final Speciality c_Speciality : Speciality.values()) {
                m_Map.put(c_Speciality.m_Index, c_Speciality);
            }
        }

        private Integer m_Index = 0;

        private Speciality(final Integer c_Index) {
            this.m_Index = c_Index;
        }

        public final static Speciality f_value_of(final Integer c_Speciality) {
            return (Speciality) m_Map.get(c_Speciality);
        }

        public final Integer f_get_value() {
            return this.m_Index;
        }
    }

    public final static Integer f_random_int(final Integer c_Min, final Integer c_Max) {
        Double result = (Math.random() * (c_Max - c_Min) + c_Min);
        return result.intValue();
    }

    public final static Double f_random_double(final Double c_Min, final Double c_Max) {
        return Math.random() * (c_Max - c_Min) + c_Min;
    }

    Run | Debug
    public final static void main(final String[] c_Args) throws Exception {
        Vector<String> v_Names = new Vector<String>();
        v_Names.add(new String("Maya Rogerson"));
        v_Names.add(new String("Liana Roy"));
        v_Names.add(new String("Liza Thomson"));
        v_Names.add(new String("Phoebe Teel"));
        v_Names.add(new String("Erika Herbertson"));
        v_Names.add(new String("Vicky Nixon"));
        v_Names.add(new String("Donelle Joiner"));
        v_Names.add(new String("Brynlee Alden"));
        v_Names.add(new String("Mae Isaacson"));
        v_Names.add(new String("Ocean Sharp"));
        v_Names.add(new String("Phil Stephenson"));
        v_Names.add(new String("Tilly Ellington"));
        v_Names.add(new String("Richmal Statham"));
        v_Names.add(new String("Margo Gadsby"));
        v_Names.add(new String("Trudie George"));
        v_Names.add(new String("Kortney Abbott"));
        v_Names.add(new String("Amery Lyon"));
        v_Names.add(new String("Brooklyn Knight "));

```

Рисунок 2.7 - Содержимое файла Main.java.

Продолжение рисунка 2.7.

```
65          final Integer c_NumberOfWorkers = v_Names.size(), c_NumberOfSpecialties = 6, c_MinAge = 18, c_MaxAge = 65;
66
67          final Double c_MinimalSalary = 100.0, c_MinSalaryK = 1.0, c_MaxSalaryK = 2.0, c_ManagerMaxMinSalary = 1000.0,
68                  c_AnalystMaxMinSalary = 800.0, c_ProgrammerMaxMinSalary = 900.0, c_TesterMaxMinSalary = 500.0,
69                  c_DesignerMaxMinSalary = 700.0, c_AccountantMaxMinSalary = 450.0;
70
71          Vector<Employee> v_Employees = new Vector<Employee>();
72
73          for (Integer v_I = 0; v_I < c_NumberOfWorkers; ++v_I) {
74              switch (Speciality.f_value_of(v_I % c_NumberOfSpecialties)) {
75              case MANAGER:
76                  v_Employees.add(new Manager(f_random_int(c_MinAge, c_MaxAge), v_Names.elementAt(v_I),
77                          f_random_double(c_MinimalSalary, c_ManagerMaxMinSalary),
78                          f_random_double(c_MinSalaryK, c_MaxSalaryK)));
79                  break;
80              case ANALYST:
81                  v_Employees.add(new Analyst(f_random_int(c_MinAge, c_MaxAge), v_Names.elementAt(v_I),
82                          f_random_double(c_MinimalSalary, c_AnalystMaxMinSalary),
83                          f_random_double(c_MinSalaryK, c_MaxSalaryK)));
84                  break;
85              case PROGRAMMER:
86                  v_Employees.add(new Programmer(f_random_int(c_MinAge, c_MaxAge), v_Names.elementAt(v_I),
87                          f_random_double(c_MinimalSalary, c_ProgrammerMaxMinSalary),
88                          f_random_double(c_MinSalaryK, c_MaxSalaryK)));
89                  break;
90              case TESTER:
91                  v_Employees.add(new Tester(f_random_int(c_MinAge, c_MaxAge), v_Names.elementAt(v_I),
92                          f_random_double(c_MinimalSalary, c_TesterMaxMinSalary),
93                          f_random_double(c_MinSalaryK, c_MaxSalaryK)));
94                  break;
95              case DESIGNER:
96                  v_Employees.add(new Designer(f_random_int(c_MinAge, c_MaxAge), v_Names.elementAt(v_I),
97                          f_random_double(c_MinimalSalary, c_DesignerMaxMinSalary),
98                          f_random_double(c_MinSalaryK, c_MaxSalaryK)));
99                  break;
100             case ACCOUNTANT:
101                 v_Employees.add(new Accountant(f_random_int(c_MinAge, c_MaxAge), v_Names.elementAt(v_I),
102                         f_random_double(c_MinimalSalary, c_AccountantMaxMinSalary),
103                         f_random_double(c_MinSalaryK, c_MaxSalaryK)));
104                 break;
105             }
106
107             v_Employees.elementAt(v_I).f_start_working();
108         }
109
110         for (Integer v_I = 0; v_I < c_NumberOfWorkers; ++v_I) {
111             final Employee c_Employee = v_Employees.elementAt(v_I);
112
113             c_Employee.f_work();
114
115             System.out.println("Salary = " + c_Employee.f_get_minimal_salary() + " * " + c_Employee.f_get_salary_k()
116                     + " = " + c_Employee.f_get_salary() + '\n');
117
118         }
119     }
120 }
121
```

Person.java

Java > Structure > src > stct > Person.java > ...

```java
package stct;

public abstract class Person {
    private Integer m_Age = 0;
    private String m_Name = new String();

    public Person(final Integer c_Age, final String c_Name) {
        this.m_Age = c_Age;
        this.m_Name = c_Name;
    }

    public final void f_set_age(final Integer c_Age) {
        this.m_Age = c_Age;
    }

    public final Integer f_get_age() {
        return this.m_Age;
    }

    public final void f_set_name(final String c_Name) {
        this.m_Name = c_Name;
    }

    public final String f_get_name() {
        return this.m_Name;
    }
}

```

Рисунок 2.8 - Содержимое файла Person.java.

Programmer.java

Java > Structure > src > stct > Programmer.java > ...

```java
package stct;

public final class Programmer extends Employee {
    public Programmer(final Integer c_Age, final String c_Name, final Double c_MinimalSalary) {
        super(c_Age, c_Name, c_MinimalSalary);
    }

    public Programmer(final Integer c_Age, final String c_Name, final Double c_MinimalSalary, final Double c_SalaryK) {
        super(c_Age, c_Name, c_MinimalSalary, c_SalaryK);
    }

    public final void f_work() {
        if (this.f_is_working()) {
            System.out.println("The programmer " + this.f_get_name() + " is working...");
            return;
        }

        System.out.println("Programmer " + this.f_get_name() + " does not work...");
    }
}

```

Рисунок 2.9 - Содержимое файла Programmer.java.

Tester.java

Java > Structure > src > stct > Tester.java > ...

```java
package stct;

public final class Tester extends Employee {
    public Tester(final Integer c_Age, final String c_Name, final Double c_MinimalSalary) {
        super(c_Age, c_Name, c_MinimalSalary);
    }

    public Tester(final Integer c_Age, final String c_Name, final Double c_MinimalSalary, final Double c_SalaryK) {
        super(c_Age, c_Name, c_MinimalSalary, c_SalaryK);
    }

    public final void f_work() {
        if (this.f_is_working()) {
            System.out.println("The tester " + this.f_get_name() + " is working...");
            return;
        }

        System.out.println("Tester " + this.f_get_name() + " does not work...");
    }
}

```

Рисунок 2.10 - Содержимое файла Tester.java.

```
PS C:\Users\User\Documents\Visual Studio Code>  c:; cd 'c:\Users\User\Documents\Visual Studio Code'; & 'c:\Users\User\.vscode\extensions\vscjava.vscode-java-debug-0.36.0\scripts\launcher.bat' 'C:\Program Files\Eclipse Foundation\jdk-11.0.12.7-hotspot\bin\java.exe' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\35645f84fe309ce1d5ab2a9af314d21a\redhat.java\jdt_ws\Visual Studio Code_9125e193\bin' 'stct.Main'
The manager Maya Rogerson is working...
Salary = 278.73813904898634 * 1.239875754599145 = 345.6006604889233

The analyst Liana Roy is working...
Salary = 636.8961703501296 * 1.136619468335396 = 723.9085865282142

The programmer Liza Thomson is working...
Salary = 671.6527807291727 * 1.0657438108626898 = 715.809794110831

The tester Phoebe Teel is working...
Salary = 101.25640413169523 * 1.941950550749827 = 196.63492977049262

The designer Erika Herbertson is working...
Salary = 132.97692494907187 * 1.9135516018727796 = 254.45820774841286

The accountant Vicky Nixon is working...
Salary = 397.3918142365974 * 1.138676916404023 = 452.500885639129

The manager Donelle Joiner is working...
Salary = 273.74351272569936 * 1.0762553593903728 = 294.6179226693806

The analyst Brynlee Alden is working...
Salary = 686.0160143412346 * 1.7031666070986937 = 1168.3995675609294

The programmer Mae Isaacson is working...
Salary = 851.9958847497701 * 1.6905391178780391 = 1440.3323714405958

The tester Ocean Sharp is working...
Salary = 233.46739650890683 * 1.7237250742183696 = 402.43360537488496

The designer Phil Stephenson is working...
Salary = 594.243816180892 * 1.1339315248859094 = 673.831796636021

The accountant Tilly Ellington is working...
Salary = 320.26183875236427 * 1.877460198223711 = 601.278855267504

The manager Richmal Statham is working...
Salary = 868.7804477414567 * 1.1204485980747139 = 973.4238347066373

The analyst Margo Gadsby is working...
Salary = 102.75963048498951 * 1.1116102736916993 = 114.22866096787708

The programmer Trudie George is working...
Salary = 623.7326039989367 * 1.7684701059072923 = 1103.0524642518308

The tester Kortney Abbott is working...
Salary = 320.5771968966012 * 1.257427459504071 = 403.1025702686296

The designer Amery Lyon is working...
Salary = 379.39546486290854 * 1.3853807879948455 = 437.84504274777077

PS C:\Users\User\Documents\Visual Studio Code>
```

Рисунок 2.11 - Результат работы программы.

Задание 3.

Требования к выполнению:

• В задании 3 ЛР №4, где возможно, заменить объявления суперклассов объявлениями абстрактных классов или интерфейсов.

Спецификация ввода:

-

Спецификация вывода:

<параметры функций System.out.println() (содержимое полей объектов)>

…

Структура проекта:

Рисунок 3.1 - Структура проекта.

Код программы:

```
MINGW64:/c/Users/User/Desktop/ssp_po5

User@PC MINGW64 ~/Desktop/ssp_po5 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Administrator.java
        modified:   reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Librarian.java
        modified:   reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Library.java
        modified:   reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Main.java
        modified:   reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Person.java
        modified:   reports/Ignatyuk/Lab4/src/Task3/Model/src/model/WorkingPerson.java

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Building.java
        reports/Ignatyuk/Lab4/src/Task3/Model/src/model/PublicBuilding.java
        reports/Ignatyuk/Lab4/src/Task3/Model/src/model/TownBuilding.java

no changes added to commit (use "git add" and/or "git commit -a")

User@PC MINGW64 ~/Desktop/ssp_po5 (main)
$
```

Рисунок 3.2 - Изменения относительно 4 работы.

Продолжение рисунка 3.2.

```
MINGW64:/c/Users/User/Desktop/ssp_po5
User@PC MINGW64 ~/Desktop/ssp_po5 (main)
$ git diff
warning: LF will be replaced by CRLF in reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Main.java.
The file will have its original line endings in your working directory
diff --git a/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Administrator.java b/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Administrator.java
index f03ff83..7c190b2 100644
--- a/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Administrator.java
+++ b/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Administrator.java
@@ -1,7 +1,7 @@
 package model;

 public final class Administrator extends WorkingPerson {
-    private Library m_Library = new Library();
+    private Library m_Library = null;

     public Administrator(final Integer c_Age, final String c_Name, final Library c_Library) {
         super(c_Age, c_Name);
diff --git a/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Librarian.java b/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Librarian.java
index 97a7d92..ed0f965 100644
--- a/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Librarian.java
+++ b/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Librarian.java
@@ -1,7 +1,7 @@
 package model;

 public final class Librarian extends WorkingPerson {
-    private Library m_Library = new Library();
+    private Library m_Library = null;

     public Librarian(final Integer c_Age, final String c_Name, final Library c_Library) {
         super(c_Age, c_Name);
diff --git a/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Library.java b/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Library.java
index 68cfe35..382dea5 100644
--- a/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Library.java
+++ b/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Library.java
@@ -3,7 +3,7 @@ package model;
 import java.util.HashSet;
 import java.util.Set;

-public final class Library {
+public final class Library extends PublicBuilding {
     private Catalog m_Catalog = new Catalog();
     private Set<Administrator> m_Administrators = new HashSet<Administrator>();
     private Set<Librarian> m_Librarians = new HashSet<Librarian>();
@@ -12,35 +12,57 @@ public final class Library {

     private String c_Deadline = new String("11/10/2021");

-    public Library() {
+    public Library(final String c_Name, final String c_Address) {
+        super(c_Name, c_Address);
     }

-    public Library(final Catalog c_Catalog) {
+    public Library(final String c_Name, final String c_Address, final Integer c_VisitorsCount) {
+        super(c_Name, c_Address, c_VisitorsCount);
+    }
+
+    public Library(final String c_Name, final String c_Address, final Integer c_VisitorsCount,
+            final Double c_VisitPrice) {
+        super(c_Name, c_Address, c_VisitorsCount, c_VisitPrice);
+    }
+
+    public Library(final String c_Name, final String c_Address, final Integer c_VisitorsCount,
+            final Double c_VisitPrice, final String c_WorkTime) {
+        super(c_Name, c_Address, c_VisitorsCount, c_VisitPrice, c_WorkTime);
+    }
+
+    public Library(final String c_Name, final String c_Address, final Catalog c_Catalog) {
+        super(c_Name, c_Address);
         this.m_Catalog = c_Catalog;
     }

-    public Library(final Catalog c_Catalog, final Set<Administrator> c_Administrators) {
+    public Library(final String c_Name, final String c_Address, final Catalog c_Catalog,
+            final Set<Administrator> c_Administrators) {
+        super(c_Name, c_Address);
         this.m_Catalog = c_Catalog;
         this.m_Administrators = c_Administrators;
     }

-    public Library(final Catalog c_Catalog, final Set<Administrator> c_Administrators,
-            final Set<Librarian> c_Librarians) {
+    public Library(final String c_Name, final String c_Address, final Catalog c_Catalog,
+            final Set<Administrator> c_Administrators, final Set<Librarian> c_Librarians) {
+        super(c_Name, c_Address);
         this.m_Catalog = c_Catalog;
         this.m_Administrators = c_Administrators;
         this.m_Librarians = c_Librarians;
     }

-    public Library(final Catalog c_Catalog, final Set<Administrator> c_Administrators,
-            final Set<Librarian> c_Librarians, final Set<Reader> c_Readers) {
+    public Library(final String c_Name, final String c_Address, final Catalog c_Catalog,
+            final Set<Administrator> c_Administrators, final Set<Librarian> c_Librarians, final Set<Reader> c_Readers) {
+        super(c_Name, c_Address);
         this.m_Catalog = c_Catalog;
         this.m_Administrators = c_Administrators;
         this.m_Librarians = c_Librarians;
         this.m_Readers = c_Readers;
     }

-    public Library(final Catalog c_Catalog, final Set<Administrator> c_Administrators,
-            final Set<Librarian> c_Librarians, final Set<Reader> c_Readers, final Set<Reader> c_BlackList) {
+    public Library(final String c_Name, final String c_Address, final Catalog c_Catalog,
+            final Set<Administrator> c_Administrators, final Set<Librarian> c_Librarians, final Set<Reader> c_Readers,
+            final Set<Reader> c_BlackList) {
+        super(c_Name, c_Address);
         this.m_Catalog = c_Catalog;
         this.m_Administrators = c_Administrators;
         this.m_Librarians = c_Librarians;
```

Продолжение рисунка 3.2.

```
@@ -48,9 +70,10 @@ public final class Library {
         this.m_BlackList = c_BlackList;
     }

-    public Library(final Catalog c_Catalog, final Set<Administrator> c_Administrators,
-            final Set<Librarian> c_Librarians, final Set<Reader> c_Readers, final Set<Reader> c_BlackList,
-            final Set<Order> c_Orders) {
+    public Library(final String c_Name, final String c_Address, final Catalog c_Catalog,
+            final Set<Administrator> c_Administrators, final Set<Librarian> c_Librarians, final Set<Reader> c_Readers,
+            final Set<Reader> c_BlackList, final Set<Order> c_Orders) {
+        super(c_Name, c_Address);
         this.m_Catalog = c_Catalog;
         this.m_Administrators = c_Administrators;
         this.m_Librarians = c_Librarians;
diff --git a/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Main.java b/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Main.java
index 06e757d..f7ee848 100644
--- a/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Main.java
+++ b/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Main.java
@@ -9,7 +9,8 @@ public final class Main {
         v_Catalog.f_add_book(new Book(2005, "Haunted", "Chuck Palahniuk"));
         v_Catalog.f_add_book(new Book(2002, "The Lovely Bones", "Alice Sebold"));

-        Library v_Library = new Library();
+        Library v_Library = new Library(new String("Alpha Park District"),
+                new String("Alpha Park Public Library District"), 350, 12.33, new String("08:00 - 17:00"));
         v_Library.f_set_catalog(v_Catalog);

         final Librarian c_Librarian = new Librarian(32, "Mayson Falconer", v_Library);
@@ -23,6 +24,13 @@ public final class Main {
         c_Librarian.f_work();
         c_Administrator.f_work();

+        System.out.println(new String("Name: ") + v_Library.f_get_name());
+        System.out.println(new String("Address: ") + v_Library.f_get_address());
+        System.out.println(new String("Visitors count: ") + v_Library.f_get_visitors_count());
+        System.out.println(new String("Visit price: ") + v_Library.f_get_visit_price());
+        System.out.println(new String("Work time: ") + v_Library.f_get_work_time());
+        System.out.println();
+
         if (v_Reader.f_request_book(v_Library, new Book(2005, "Haunted", "Chuck Palahniuk"))) {
             System.out.println("Book successfully taken");
         } else {
diff --git a/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Person.java b/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Person.java
index 496f5e5..962b6e4 100644
--- a/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Person.java
+++ b/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/Person.java
@@ -1,6 +1,6 @@
 package model;

-public class Person {
+public abstract class Person {
     private Integer m_Age = 0;
     private String m_Name = new String();

diff --git a/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/WorkingPerson.java b/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/WorkingPerson.java
index 1ebd99c..44f662b 100644
--- a/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/WorkingPerson.java
+++ b/reports/Ignatyuk/Lab4/src/Task3/Model/src/model/WorkingPerson.java
@@ -1,6 +1,6 @@
 package model;

-public class WorkingPerson extends Person implements Worker {
+public abstract class WorkingPerson extends Person implements Worker {
     private Boolean m_IsWorking = Boolean.FALSE;

     public WorkingPerson(final Integer c_Age, final String c_Name) {
(END)
```

Новые файлы идентичны одноименным файлам из задания 2. Все остальные исправления показаны через git diff.

```
PS C:\Users\User\Documents\Visual Studio Code>  & 'c:\Users\User\.vscode\extensions\vscjava.vscode-java-debug-0.36
.0\scripts\launcher.bat' 'C:\Program Files\Eclipse Foundation\jdk-11.0.12.7-hotspot\bin\java.exe' '-Dfile.encoding
=UTF-8' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\35645f84fe309ce1d5ab2a9af314d21a\redhat.ja
va\jdt_ws\Visual Studio Code_9125e193\bin' 'model.Main'
Name: Alpha Park District
Address: Alpha Park Public Library District
Visitors count: 350
Visit price: 12.33
Work time: 08:00 - 17:00

Book successfully taken
PS C:\Users\User\Documents\Visual Studio Code>
```

Рисунок 3.3 - Результат работы программы.

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