

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №4

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 указанный класс, включив в него вспомогательный внутренний класс или классы.
- Реализовать 2-3 метода (на выбор).
- Продемонстрировать использование реализованных классов.

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.

Спецификация ввода:

-

Спецификация вывода:

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

...

Структура проекта:

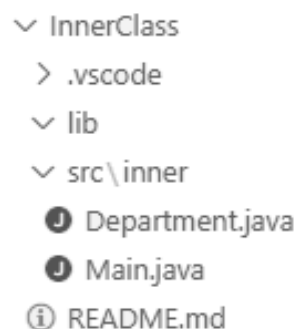


Рисунок 1.1 - Структура проекта.

Код программы:

1 Main.java X

Java > InnerClass > src > inner > 1 Main.java > ...

```
1 package inner;
2
3 import java.util.HashSet;
4 import java.util.Set;
5
6 public final class Main {
    Run | Debug
7     public final static void main(final String[] c_Args) throws Exception {
8         Department.Position.Employee v_Lilah = new Department.Position.Employee(21, "Lilah Boon"),
9         v_Ethan = new Department.Position.Employee(21, "Ethan Brand"),
10        v_Teri = new Department.Position.Employee(28, "Teri Parish"),
11        v_Morty = new Department.Position.Employee(19, "Morty Pond"),
12        v_Karyn = new Department.Position.Employee(24, "Karyn Scrivens"),
13        v_Jeffrey = new Department.Position.Employee(22, "Jeffrey Fairbairn"),
14        v_Zara = new Department.Position.Employee(29, "Zara Vernon"),
15        v_Jolene = new Department.Position.Employee(32, "Jolene Moon"),
16        v_Sharalyn = new Department.Position.Employee(25, "Sharalyn Lowe");
17
18        Department.Position v_Marketer = new Department.Position("Marketer"),
19        v_BusinessAnalyst = new Department.Position("Business Analyst"),
20        v_SalesManager = new Department.Position("Sales Manager");
21
22        v_Marketer.f_add_employee(v_Lilah);
23        v_Marketer.f_add_employee(v_Ethan);
24        v_Marketer.f_add_employee(v_Teri);
25
26        v_BusinessAnalyst.f_add_employee(v_Morty);
27        v_BusinessAnalyst.f_add_employee(v_Karyn);
28        v_BusinessAnalyst.f_add_employee(v_Jeffrey);
29
30        v_SalesManager.f_add_employee(v_Zara);
31        v_SalesManager.f_add_employee(v_Jolene);
32        v_SalesManager.f_add_employee(v_Sharalyn);
33
34        Set<Department.Position> v_SalesPositions = new HashSet<Department.Position>();
35        v_SalesPositions.add(v_Marketer);
36        v_SalesPositions.add(v_BusinessAnalyst);
37        v_SalesPositions.add(v_SalesManager);
38
39        Department v_Sales = new Department(new String("Sales Department"));
40        v_Sales.f_set_positions(v_SalesPositions);
41
42        System.out.println(v_Sales.f_get_name() + new String(":"));
43
44        for (final Department.Position c_Position : v_Sales.f_get_positions()) {
45            System.out.println(new String("\t") + c_Position.f_get_name() + new String(":"));
46
47            for (final Department.Position.Employee c_Employee : c_Position.f_get_employees()) {
48                System.out.println(new String("\t\t") + c_Employee.f_get_name());
49            }
50
51            System.out.println();
52        }
53
54        ///
55    }
```

Рисунок 1.2 - Содержимое файла Main.java.

Продолжение рисунка 1.2.

```

56 Department.Position.Employee v_Brand = new Department.Position.Employee(28, "Brand Ash"),
57     v_Ariella = new Department.Position.Employee(20, "Ariella Evered"),
58     v_Kenneth = new Department.Position.Employee(20, "Kenneth Stafford"),
59     v_Blaze = new Department.Position.Employee(24, "Blaze Wilton"),
60     v_Camryn = new Department.Position.Employee(33, "Camryn Kingston"),
61     v_Ford = new Department.Position.Employee(26, "Ford Lamb"),
62     v_Dorinda = new Department.Position.Employee(18, "Dorinda Constable"),
63     v_Francis = new Department.Position.Employee(36, "Francis Bray"),
64     v_Peace = new Department.Position.Employee(31, "Peace Morris");
65
66 Department.Position v_Administrator = new Department.Position("Administrator"),
67     v_Programmer = new Department.Position("Programmer"),
68     v_SecuritySpecialist = new Department.Position("Security Specialist");
69
70 v_Administrator.f_add_employee(v_Brand);
71 v_Administrator.f_add_employee(v_Ariella);
72 v_Administrator.f_add_employee(v_Kenneth);
73
74 v_Programmer.f_add_employee(v_Blaze);
75 v_Programmer.f_add_employee(v_Camryn);
76 v_Programmer.f_add_employee(v_Ford);
77
78 v_SecuritySpecialist.f_add_employee(v_Dorinda);
79 v_SecuritySpecialist.f_add_employee(v_Francis);
80 v_SecuritySpecialist.f_add_employee(v_Peace);
81
82 Department v_IT = new Department(new String("IT Department"));
83 v_IT.f_add_position(v_Administrator);
84 v_IT.f_add_position(v_Programmer);
85 v_IT.f_add_position(v_SecuritySpecialist);
86
87 System.out.println(v_IT.f_get_name() + new String(":"));
88
89 for (final Department.Position c_Position : v_IT.f_get_positions()) {
90     System.out.println(new String("\t") + c_Position.f_get_name() + new String(":"));
91
92     for (final Department.Position.Employee c_Employee : c_Position.f_get_employees()) {
93         System.out.println(new String("\t\t") + c_Employee.f_get_name());
94     }
95
96     System.out.println();
97 }
98 }
99 }
100

```

Department.java X

Java > InnerClass > src > inner > Department.java > ...

```
1  package inner;
2
3  import java.util.HashSet;
4  import java.util.Set;
5
6  public final class Department {
7      private String m_Name = new String();
8      private Set<Department.Position> m_Positions = new HashSet<Department.Position>();
9
10     public Department() {
11     }
12
13     public Department(final String c_Name) {
14         this.m_Name = c_Name;
15     }
16
17     public Department(final String c_Name, final Set<Department.Position> c_Positions) {
18         this.m_Name = c_Name;
19         this.m_Positions = c_Positions;
20     }
21
22     public final void f_set_name(final String c_Name) {
23         this.m_Name = c_Name;
24     }
25
26     public final String f_get_name() {
27         return this.m_Name;
28     }
29
30     public final void f_set_positions(final Set<Department.Position> c_Positions) {
31         this.m_Positions = c_Positions;
32     }
33
34     public final Set<Department.Position> f_get_positions() {
35         return this.m_Positions;
36     }
37
38     public final void f_add_position(final Department.Position c_Position) {
39         this.m_Positions.add(c_Position);
40     }
41
42     public final void f_remove_position(final Department.Position c_Position) {
43         this.m_Positions.remove(c_Position);
44     }
45
46     public final static class Position {
47         private String m_Name = new String();
48         private Set<Position.Employee> m_Employees = new HashSet<Position.Employee>();
49
50         public Position() {
51         }
52
53         public Position(final String c_Name) {
54             this.m_Name = c_Name;
55         }
56
57         public Position(final String c_Name, final Set<Position.Employee> c_Employees) {
58             this.m_Name = c_Name;
59             this.m_Employees = c_Employees;
60         }
61
62         public final void f_set_name(final String c_Name) {
63             this.m_Name = c_Name;
64         }
65
66         public final String f_get_name() {
67             return this.m_Name;
68         }
69     }
```

Рисунок 1.3 - Содержимое файла Department.java.

Продолжение рисунка 1.3.

```
70     public final void f_set_employees(final Set<Position.Employee> c_Employees) {
71         |     this.m_Employees = c_Employees;
72     }
73
74     public final Set<Position.Employee> f_get_employees() {
75         |     return this.m_Employees;
76     }
77
78     public final void f_add_employee(final Position.Employee c_Employee) {
79         |     this.m_Employees.add(c_Employee);
80     }
81
82     public final void f_remove_employee(final Position.Employee c_Employee) {
83         |     this.m_Employees.remove(c_Employee);
84     }
85
86     public final static class Employee {
87         |     private Integer m_Age;
88         |     private String m_Name = new String();
89
90         |     public Employee(final Integer c_Age, final String c_Name) {
91         |         |     this.m_Age = c_Age;
92         |         |     this.m_Name = c_Name;
93         |     }
94
95         |     public final void f_set_age(final Integer c_Age) {
96         |         |     this.m_Age = c_Age;
97         |     }
98
99         |     public final Integer f_get_age() {
100        |         |     return this.m_Age;
101        |     }
102
103        |     public final void f_set_name(final String c_Name) {
104        |         |     this.m_Name = c_Name;
105        |     }
106
107        |     public final String f_get_name() {
108        |         |     return this.m_Name;
109        |     }
110    }
111 }
112 }
113
```

```
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' 'inner.Main'
```

Sales Department:

Business Analyst:

Morty Pond
Jeffrey Fairbairn
Karyn Scrivens

Marketer:

Teri Parish
Lilah Boon
Ethan Brand

Sales Manager:

Zara Vernon
Jolene Moon
Sharalyn Lowe

IT Department:

Administrator:

Brand Ash
Kenneth Stafford
Ariella Evered

Security Specialist:

Peace Morris
Francis Bray
Dorinda Constable

Programmer:

Camryn Kingston
Ford Lamb
Blaze Wilton

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

Рисунок 1.4 - Результат выполнения программы.

Задание 2.

Требования к выполнению:

- 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.

5) Создать класс Абзац, используя класс Слово.

Спецификация ввода:

-

Спецификация вывода:

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

...

Структура проекта:

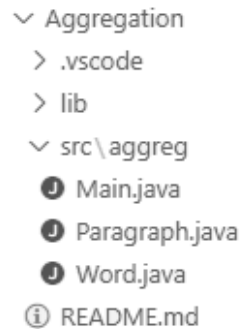


Рисунок 2.1 - Структура проекта.

Код программы:

1 Main.java X

Java > Aggregation > src > aggreg > 1 Main.java > ...

```
1  package aggreg;
2
3  import java.util.Vector;
4
5  public final class Main {
6      Run | Debug
7      public final static void main(final String[] c_Args) {
8          Vector<Word> v_Words = new Vector<Word>();
9          v_Words.add(new Word(new String("Hello")));
10         v_Words.add(new Word(new String("world")));
11
12         Paragraph v_Paragraph = new Paragraph(v_Words);
13
14         for (final Word c_Word : v_Paragraph.f_get_words()) {
15             System.out.print(c_Word.f_get_data() + new String(" "));
16         }
17
18         v_Paragraph.f_add_word(new Word(new String("we")));
19         v_Paragraph.f_add_word(new Word(new String("are")));
20         v_Paragraph.f_add_word(new Word(new String("here")));
21
22         System.out.println();
23
24         for (final Word c_Word : v_Paragraph.f_get_words()) {
25             System.out.print(c_Word.f_get_data() + new String(" "));
26         }
27     }
28
29 }
```

Рисунок 2.2 - Содержимое файла Main.java.

Paragraph.java X

Java > Aggregation > src > aggreg > Paragraph.java > ...

```
1  package aggreg;
2
3  import java.util.Vector;
4
5  public final class Paragraph {
6      private Vector<Word> m_Words = new Vector<Word>();
7
8      public Paragraph() {
9      }
10
11     public Paragraph(final Vector<Word> c_Words) {
12         this.m_Words = c_Words;
13     }
14
15     public final void f_set_words(final Vector<Word> c_Words) {
16         this.m_Words = c_Words;
17     }
18
19     public final Vector<Word> f_get_words() {
20         return this.m_Words;
21     }
22
23     public final void f_add_word(final Word c_Word) {
24         this.m_Words.add(c_Word);
25     }
26 }
27
28
```

Рисунок 2.3 - Содержимое файла Paragraph.java.

Word.java X

Java > Aggregation > src > aggreg > Word.java > ...

```
1  package aggreg;
2
3  public final class Word {
4      private String m_Data = new String();
5
6      public Word(final String c_Data) {
7          this.m_Data = c_Data;
8      }
9
10     public final void f_set_data(final String c_Data) {
11         this.m_Data = c_Data;
12     }
13
14     public final String f_get_data() {
15         return this.m_Data;
16     }
17 }
18
```

Рисунок 2.4 - Содержимое файла Word.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' 'aggreg.Main'
Hello world
Hello world we are here
PS C:\Users\User\Documents\Visual Studio Code>

```

Рисунок 2.5 - Результат выполнения программы.

Задание 3.

Требования к выполнению:

- 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.

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”.

Спецификация ввода:

-

Спецификация вывода:

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

...

Структура проекта:



Рисунок 3.1 - Структура проекта.

Код программы:

1 Main.java X

Java > Model > src > model > 1 Main.java

```
1 package model;
2
3 public final class Main {
    Run | Debug
4     public final static void main(final String[] c_Args) throws Exception {
5         Catalog v_Catalog = new Catalog();
6         v_Catalog.f_add_book(new Book(2006, "The Road", "Cormac McCarthy"));
7         v_Catalog.f_add_book(new Book(1985, "Perfume: The Story of a Murderer", "Patrick Süskind"));
8         v_Catalog.f_add_book(new Book(2003, "We Need to Talk About Kevin", "Lionel Shriver"));
9         v_Catalog.f_add_book(new Book(2005, "Haunted", "Chuck Palahniuk"));
10        v_Catalog.f_add_book(new Book(2002, "The Lovely Bones", "Alice Sebold"));
11
12        Library v_Library = new Library();
13        v_Library.f_set_catalog(v_Catalog);
14
15        final Librarian c_Librarian = new Librarian(32, "Mayson Falconer", v_Library);
16        final Administrator c_Administrator = new Administrator(38, "Janette Trueman", v_Library);
17        final Reader v_Reader = new Reader(19, "Conner Thacker");
18
19        v_Library.f_add_librarian(c_Librarian);
20        v_Library.f_add_administrator(c_Administrator);
21        v_Library.f_add_reader(v_Reader);
22
23        c_Librarian.f_work();
24        c_Administrator.f_work();
25
26        if (v_Reader.f_request_book(v_Library, new Book(2005, "Haunted", "Chuck Palahniuk"))) {
27            System.out.println("Book successfully taken");
28        } else {
29            System.out.println("Can not take a book");
30        }
31
32        v_Library.f_update();
33    }
34 }
35
```

Рисунок 3.2 - Содержимое файла Main.java.

1 Worker.java X

Java > Model > src > model > 1 Worker.java > ...

```
1 package model;
2
3 public interface Worker {
4     public void f_work();
5
6     public Boolean f_is_working();
7
8     public void f_stop_working();
9 }
10
```

Рисунок 3.3 - Содержимое файла Worker.java.

Administrator.java X

Java > Model > src > model > Administrator.java > ...

```
1  package model;
2
3  public final class Administrator extends WorkingPerson {
4      private Library m_Library = new Library();
5
6      public Administrator(final Integer c_Age, final String c_Name, final Library c_Library) {
7          super(c_Age, c_Name);
8          this.m_Library = c_Library;
9      }
10
11     public final void f_set_library(final Library c_Library) {
12         this.m_Library = c_Library;
13     }
14
15     public final Library f_get_library() {
16         return this.m_Library;
17     }
18
19     public final void f_add_to_black_list(final Reader c_Reader) {
20         this.m_Library.f_add_to_black_list(c_Reader);
21     }
22
23     public final void f_update() {
24         for (final Order c_Order : m_Library.f_get_orders()) {
25             if (!m_Library.f_check_deadline(c_Order)) {
26                 this.f_add_to_black_list(c_Order.f_get_reader());
27             }
28         }
29     }
30 }
31
```

Рисунок 3.4 - Содержимое файла Administrator.java.

WorkingPerson.java X

Java > Model > src > model > WorkingPerson.java > ...

```
1  package model;
2
3  public class WorkingPerson extends Person implements Worker {
4      private Boolean m_IsWorking = Boolean.FALSE;
5
6      public WorkingPerson(final Integer c_Age, final String c_Name) {
7          super(c_Age, c_Name);
8      }
9
10     public final void f_work() {
11         this.m_IsWorking = Boolean.TRUE;
12     }
13
14     public final Boolean f_is_working() {
15         return this.m_IsWorking;
16     }
17
18     public final void f_stop_working() {
19         this.m_IsWorking = Boolean.FALSE;
20     }
21 }
22
```

Рисунок 3.5 - Содержимое файла WorkingPerson.java.

Java > Model > src > model > Book.java > ...

```
1  package model;
2
3  public final class Book {
4      private Integer m_Year = 0;
5      private String m_Title = new String(), m_Author = new String();
6
7      public Book(final Integer c_Year, final String c_Title, final String c_Author) {
8          this.m_Year = c_Year;
9          this.m_Title = c_Title;
10         this.m_Author = c_Author;
11     }
12
13     public final void f_set_year(final Integer c_Year) {
14         this.m_Year = c_Year;
15     }
16
17     public final Integer f_get_year() {
18         return this.m_Year;
19     }
20
21     public final void f_set_title(final String c_Title) {
22         this.m_Title = c_Title;
23     }
24
25     public final String f_get_title() {
26         return this.m_Title;
27     }
28
29     public final void f_set_author(final String c_Author) {
30         this.m_Author = c_Author;
31     }
32
33     public final String f_get_author() {
34         return this.m_Author;
35     }
36
37     public final boolean equals(final Object c_Other) {
38         if (this.f_get_year().equals(((Book) c_Other).f_get_year())
39             && this.f_get_title().equals(((Book) c_Other).f_get_title())
40             && this.f_get_author().equals(((Book) c_Other).f_get_author())) {
41             return Boolean.TRUE;
42         }
43
44         return Boolean.FALSE;
45     }
46 }
47
```

Рисунок 3.6 - Содержимое файла Book.java.

Java > Model > src > model > 1 Catalog.java > ...

```

1  package model;
2
3  import java.util.Vector;
4
5  public final class Catalog {
6      private Vector<Book> m_Books = new Vector<Book>();
7
8      public Catalog() {
9      }
10
11     public Catalog(final Vector<Book> c_Books) {
12         this.m_Books = c_Books;
13     }
14
15     public final void f_set_books(final Vector<Book> c_Books) {
16         this.m_Books = c_Books;
17     }
18
19     public final Vector<Book> f_get_books() {
20         return this.m_Books;
21     }
22
23     public final void f_add_book(final Book c_Book) {
24         this.m_Books.add(c_Book);
25     }
26
27     public final void f_remove_book(final Book c_Book) {
28         this.m_Books.removeElement(c_Book);
29     }
30
31     public final Boolean f_book_search(final Book c_Book) {
32         if (this.m_Books.indexOf(c_Book) != -1) {
33             return Boolean.TRUE;
34         }
35
36         return Boolean.FALSE;
37     }
38
39     public final Book f_give_book(final Book c_Book) {
40         final Integer c_Index = this.m_Books.indexOf(c_Book);
41
42         if (c_Index == -1) {
43             return null;
44         }
45
46         final Book c_Result = this.m_Books.get(c_Index);
47         this.f_remove_book(c_Book);
48
49         return c_Result;
50     }
51 }
52

```

Рисунок 3.7 - Содержимое файла Catalog.java.

Person.java X

Java > Model > src > model > Person.java > ...

```

3  public class Person {
4      private Integer m_Age = 0;
5      private String m_Name = new String();
6
7      public Person(final Integer c_Age, final String c_Name) {
8          this.m_Age = c_Age;
9          this.m_Name = c_Name;
10     }
11
12     public final void f_set_age(final Integer c_Age) {
13         this.m_Age = c_Age;
14     }
15
16     public final Integer f_get_age() {
17         return this.m_Age;
18     }
19
20     public final void f_set_name(final String c_Name) {
21         this.m_Name = c_Name;
22     }
23
24     public final String f_get_name() {
25         return this.m_Name;
26     }
27 }
28

```

Рисунок 3.8 - Содержимое файла Person.java.

Librarian.java X

Java > Model > src > model > Librarian.java > ...

```

1  package model;
2
3  public final class Librarian extends WorkingPerson {
4      private Library m_Library = new Library();
5
6      public Librarian(final Integer c_Age, final String c_Name, final Library c_Library) {
7          super(c_Age, c_Name);
8          this.m_Library = c_Library;
9      }
10
11     public final void f_set_library(final Library c_Library) {
12         this.m_Library = c_Library;
13     }
14
15     public final Library f_get_library() {
16         return this.m_Library;
17     }
18
19     public final void f_add_order(final Order c_Order) {
20         this.m_Library.f_add_order(c_Order);
21     }
22 }
23

```

Рисунок 3.9 - Содержимое файла Librarian.java.

```

1  package model;
2
3  public final class Order {
4      private Book m_Book = null;
5      private Reader m_Reader = null;
6      private String m_Deadline = new String();
7
8      public Order(final Book c_Book, final Reader c_Reader, final String c_Deadline) {
9          this.m_Book = c_Book;
10         this.m_Reader = c_Reader;
11         this.m_Deadline = c_Deadline;
12     }
13
14     public final void f_set_book(final Book c_Book) {
15         this.m_Book = c_Book;
16     }
17
18     public final Book f_get_book() {
19         return this.m_Book;
20     }
21
22     public final void f_set_reader(final Reader c_Reader) {
23         this.m_Reader = c_Reader;
24     }
25
26     public final Reader f_get_reader() {
27         return this.m_Reader;
28     }
29
30     public final void f_set_deadline(final String c_Deadline) {
31         this.m_Deadline = c_Deadline;
32     }
33
34     public final String f_get_deadline() {
35         return this.m_Deadline;
36     }
37 }
38

```

Рисунок 3.10 - Содержимое файла Order.java.

Java > Model > src > model > 1 Reader.java > ...

```

1  package model;
2
3  import java.util.HashSet;
4  import java.util.Set;
5
6  public final class Reader extends Person {
7      private Set<Order> m_Orders = new HashSet<Order>();
8
9      public Reader(final Integer c_Age, final String c_Name) {
10         super(c_Age, c_Name);
11     }
12
13     public Reader(final Integer c_Age, final String c_Name, final Set<Order> c_Orders) {
14         super(c_Age, c_Name);
15         this.m_Orders = c_Orders;
16     }
17
18     public final void f_set_orders(final Set<Order> c_Orders) {
19         this.m_Orders = c_Orders;
20     }
21
22     public final Set<Order> f_get_orders() {
23         return this.m_Orders;
24     }
25
26     public final void f_add_order(final Order c_Order) {
27         this.m_Orders.add(c_Order);
28     }
29
30     public final void f_remove_order(final Order c_Order) {
31         this.m_Orders.remove(c_Order);
32     }
33
34     public final Boolean f_request_book(final Library c_Library, final Book c_Book) {
35         if (c_Library.f_book_search(c_Book)) {
36             final Order c_Order = c_Library.f_process_order(this, c_Book);
37
38             if (c_Order == null) {
39                 return Boolean.FALSE;
40             }
41
42             this.f_add_order(c_Order);
43             return Boolean.TRUE;
44         }
45
46         return Boolean.FALSE;
47     }
48 }
49

```

Рисунок 3.11 - Содержимое файла Reader.java.

Java > Model > src > model > Library.java > ...

```

1  package model;
2
3  import java.util.HashSet;
4  import java.util.Set;
5
6  public final class Library {
7      private Catalog m_Catalog = new Catalog();
8      private Set<Administrator> m_Administrators = new HashSet<Administrator>();
9      private Set<Librarian> m_Librarians = new HashSet<Librarian>();
10     private Set<Reader> m_Readers = new HashSet<Reader>(), m_BlackList = new HashSet<Reader>();
11     private Set<Order> m_Orders = new HashSet<Order>();
12
13     private String c_Deadline = new String("11/10/2021");
14
15     public Library() {
16     }
17
18     public Library(final Catalog c_Catalog) {
19         this.m_Catalog = c_Catalog;
20     }
21
22     public Library(final Catalog c_Catalog, final Set<Administrator> c_Administrators) {
23         this.m_Catalog = c_Catalog;
24         this.m_Administrators = c_Administrators;
25     }
26
27     public Library(final Catalog c_Catalog, final Set<Administrator> c_Administrators,
28         final Set<Librarian> c_Librarians) {
29         this.m_Catalog = c_Catalog;
30         this.m_Administrators = c_Administrators;
31         this.m_Librarians = c_Librarians;
32     }
33
34     public Library(final Catalog c_Catalog, final Set<Administrator> c_Administrators,
35         final Set<Librarian> c_Librarians, final Set<Reader> c_Readers) {
36         this.m_Catalog = c_Catalog;
37         this.m_Administrators = c_Administrators;
38         this.m_Librarians = c_Librarians;
39         this.m_Readers = c_Readers;
40     }
41
42     public Library(final Catalog c_Catalog, final Set<Administrator> c_Administrators,
43         final Set<Librarian> c_Librarians, final Set<Reader> c_Readers, final Set<Reader> c_BlackList) {
44         this.m_Catalog = c_Catalog;
45         this.m_Administrators = c_Administrators;
46         this.m_Librarians = c_Librarians;
47         this.m_Readers = c_Readers;
48         this.m_BlackList = c_BlackList;
49     }
50
51     public Library(final Catalog c_Catalog, final Set<Administrator> c_Administrators,
52         final Set<Librarian> c_Librarians, final Set<Reader> c_Readers, final Set<Reader> c_BlackList,
53         final Set<Order> c_Orders) {
54         this.m_Catalog = c_Catalog;
55         this.m_Administrators = c_Administrators;
56         this.m_Librarians = c_Librarians;
57         this.m_Readers = c_Readers;
58         this.m_BlackList = c_BlackList;
59         this.m_Orders = c_Orders;
60     }
61 }

```

Рисунок 3.12 - Содержимое файла Library.java.

Продолжение рисунка 3.12.

```
62 public final void f_set_catalog(final Catalog c_Catalog) {
63     this.m_Catalog = c_Catalog;
64 }
65
66 public final Catalog f_get_catalog() {
67     return this.m_Catalog;
68 }
69
70 public final void f_set_administrators(final Set<Administrator> c_Administrators) {
71     this.m_Administrators = c_Administrators;
72 }
73
74 public final Set<Administrator> f_get_administrators() {
75     return this.m_Administrators;
76 }
77
78 public final void f_set_librarians(final Set<Librarian> c_Librarians) {
79     this.m_Librarians = c_Librarians;
80 }
81
82 public final Set<Librarian> f_get_librarians() {
83     return this.m_Librarians;
84 }
85
86 public final void f_set_readers(final Set<Reader> c_Readers) {
87     this.m_Readers = c_Readers;
88 }
89
90 public final Set<Reader> f_get_readers() {
91     return this.m_Readers;
92 }
93
94 public final void f_set_black_list(final Set<Reader> c_BlackList) {
95     this.m_BlackList = c_BlackList;
96 }
97
98 public final Set<Reader> f_get_black_list() {
99     return this.m_BlackList;
100 }
101
102 public final void f_set_orders(final Set<Order> c_Orders) {
103     this.m_Orders = c_Orders;
104 }
105
106 public final Set<Order> f_get_orders() {
107     return this.m_Orders;
108 }
109
110 public final void f_add_book(final Book c_Book) {
111     this.m_Catalog.f_add_book(c_Book);
112 }
113
114 public final void f_remove_book(final Book c_Book) {
115     this.m_Catalog.f_remove_book(c_Book);
116 }
117
118 public final void f_add_administrator(final Administrator c_Administrator) {
119     this.m_Administrators.add(c_Administrator);
120 }
121
```

Продолжение рисунка 3.12.

```
122 public final void f_remove_administrator(final Administrator c_Administrator) {
123 |     this.m_Administrators.remove(c_Administrator);
124 | }
125
126 public final void f_add_librarian(final Librarian c_Librarian) {
127 |     this.m_Librarians.add(c_Librarian);
128 | }
129
130 public final void f_remove_librarian(final Librarian c_Librarian) {
131 |     this.m_Librarians.remove(c_Librarian);
132 | }
133
134 public final void f_add_reader(final Reader c_Reader) {
135 |     this.m_Readers.add(c_Reader);
136 | }
137
138 public final void f_remove_reader(final Reader c_Reader) {
139 |     this.m_Readers.remove(c_Reader);
140 | }
141
142 public final void f_add_to_black_list(final Reader c_Reader) {
143 |     this.m_BlackList.add(c_Reader);
144 | }
145
146 public final void f_remove_from_black_list(final Reader c_Reader) {
147 |     this.m_BlackList.remove(c_Reader);
148 | }
149
150 public final void f_add_order(final Order c_Order) {
151 |     this.m_Orders.add(c_Order);
152 | }
153
154 public final void f_remove_order(final Order c_Order) {
155 |     this.m_Orders.remove(c_Order);
156 | }
157
158 public final Boolean f_book_search(final Book c_Book) {
159 |     return this.m_Catalog.f_book_search(c_Book);
160 | }
161
162 public final String f_generate_deadline() {
163 |     return c_Deadline;
164 | }
165
166 public final Boolean f_check_deadline(final Order c_Order) {
167 |     return c_Deadline.equals(c_Order.f_get_deadline());
168 | }
169
```

Продолжение рисунка 3.12.

```
170     public final Order f_process_order(final Reader c_Reader, final Book c_Book) {
171         final Book c_OrderedBook = this.m_Catalog.f_give_book(c_Book);
172
173         if (c_OrderedBook == null) {
174             return null;
175         }
176
177         final String c_Deadline = this.f_generate_deadline();
178         final Order c_Order = new Order(c_OrderedBook, c_Reader, c_Deadline);
179
180         Boolean v_Processed = Boolean.FALSE;
181
182         while (!v_Processed) {
183             for (final Librarian c_Librarian : m_Librarians) {
184                 if (c_Librarian.f_is_working()) {
185                     c_Librarian.f_add_order(c_Order);
186                     v_Processed = Boolean.TRUE;
187                     break;
188                 }
189             }
190         }
191
192         this.m_Orders.add(c_Order);
193         return c_Order;
194     }
195
196     public final void f_update() {
197         Boolean v_Updated = Boolean.FALSE;
198
199         while (!v_Updated) {
200             for (final Administrator c_Administrator : m_Administrators) {
201                 if (c_Administrator.f_is_working()) {
202                     c_Administrator.f_update();
203                     v_Updated = Boolean.TRUE;
204                     break;
205                 }
206             }
207         }
208     }
209 }
210
```

```
PS C:\Users\User\Documents\Visual Studio Code> & 'c:\Users\User\.vscode\extensions\vscjava.vscod
e-java-debug-0.36.0\scripts\launcher.bat' 'C:\Program Files\Eclipse Foundation\jdk-11.0.12.7-hotsp
ot\bin\java.exe' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspac
eStorage\35645f84fe309ce1d5ab2a9af314d21a\redhat.java\jdt_ws\Visual Studio Code_9125e193\bin' 'mo
del.Main'
Book successfully taken
PS C:\Users\User\Documents\Visual Studio Code>
```

Рисунок 3.13 - Результат работы программы.

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