

Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Факультет программной инженерии и компьютерной техники

Лабораторная работа №4
по программированию
Вариант № 342562.2

Выполнила: Гафурова Фарангиз

Фуркатовна

Группа: Р3120

Принял: Кулинич Ярослав

Вадимович

Санкт-Петербург

2023г

Оглавление

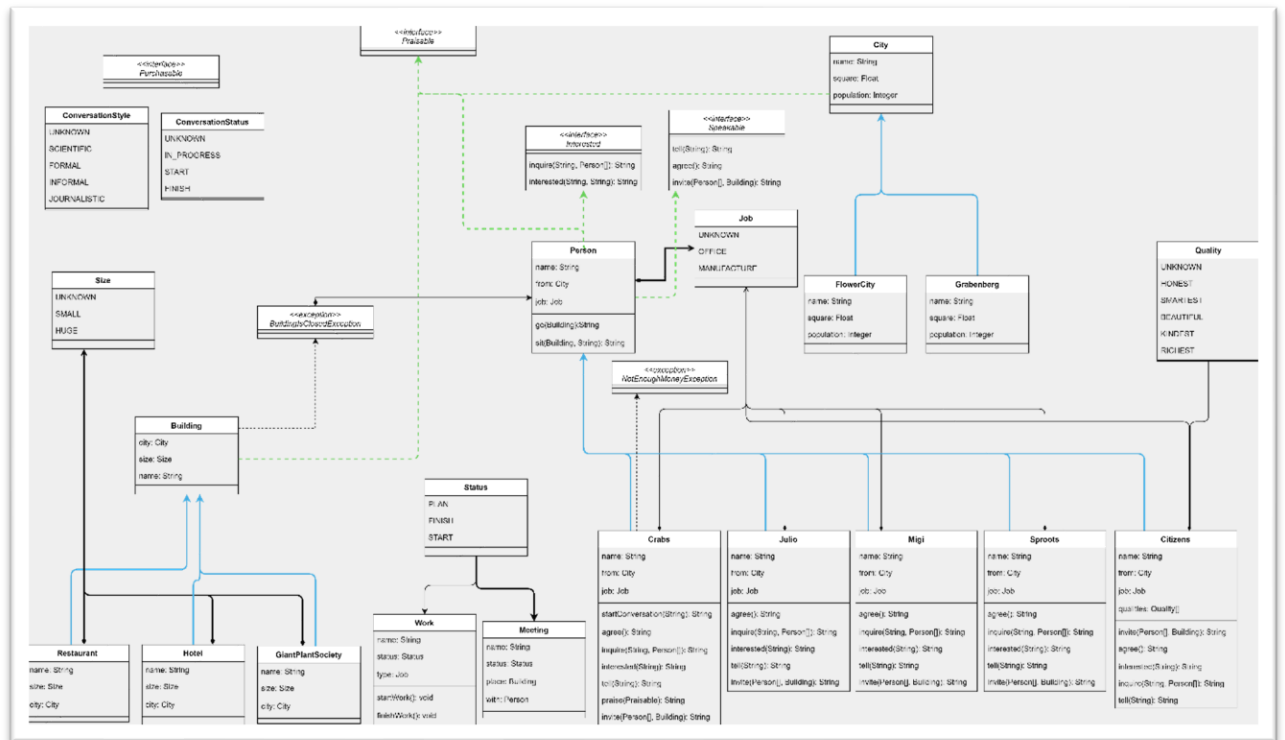
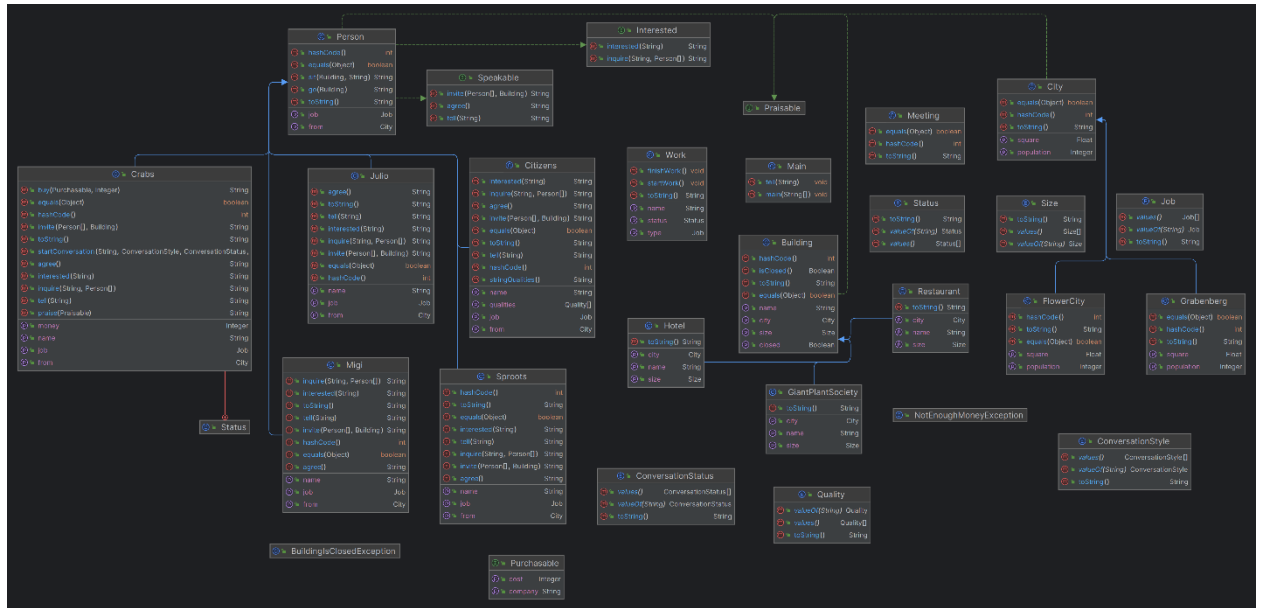
Текст задания.....	3
Диаграмма классов объектной модели	3
Исходный код программы (класс main)	4
Результат работы программы.....	6
Вывод.....	7

Текст задания

Доработать программу из [лабораторной работы #3](#), обновив реализацию объектной модели в соответствии с новой версией описания предметной области.

Результатом этого разговора было то, что на следующее утро господин Крабс появился в конторе Общества гигантских растений. Купив для отвода глаз несколько акций, он отозвал в сторонку Мигу и Жулио и сказал: Миге и Жулио чрезвычайно интересно было узнать, что понадобилось от них знаменитому фабриканту, и они тотчас согласились на встречу. Как только работа в конторе была закончена, они отправились в номер гостиницы, где у них было назначено свидание с Крабсом. Господин Крабс предложил им поужинать вместе, и через минуту все трое сидели в ресторане за столиком. По обыкновению, свойственному всем деловым коротышкам, господин Крабс начал разговор с отдаленных предметов. Осведомившись у Миги и Жулио, случилось ли им бывать в городе Грабенберге, и узнав, что им уже довелось побывать там, он начал всячески расхваливать этот город и его жителей, говоря, что все они умнейшие, добрейшие и честнейшие коротышки на свете и что господин Спрутс, который является коренным грабенбержцем, также умнейший, достойнейший и честнейший коротышка и вдобавок ко всему обладает таким колоссальным богатством, какое многим даже во сне не снилось.

Диаграмма классов объектной модели



Исходный код программы (класс main)

```
1  > import ...
8
9  public class Main {
10     public static void main(String[] args) {
11         Grabenberg grabenberg = new Grabenberg( name: "Grabenberg", square: 1439F, population: 7_121_049);
12         FlowerCity flowerCity = new FlowerCity( name: "Flower City", square: 331.2F, population: 642_152);
13
14         Migi migi = new Migi( name: "Migi", flowerCity, Job.OFFICE);
15         Julio julio = new Julio( name: "Julio", flowerCity, Job.OFFICE);
16         Crabs crabs = new Crabs( name: "Crabs", grabenberg, Job.MANUFACTURE, money: 15000);
17         Sproots sproots = new Sproots( name: "Sproots", grabenberg, Job.UNKNOWN);
18         Citizens citizens = new Citizens( name: "citizens", grabenberg, Job.UNKNOWN, new Quality[] {Quality.SMARTEST, Quality.KINDEST, Qual
19
20         Hotel hotel = new Hotel( name: "hotel", Size.HUGE, grabenberg, isClosed: false);
21         Restaurant restaurant = new Restaurant( name: "restaurant", Size.SMALL, grabenberg, isClosed: false);
22         GiantPlantSociety giantPlantSociety = new GiantPlantSociety( name: "Giant Plant Society", Size.HUGE, grabenberg, isClosed: false);
23
24         Work work = new Work( name: "work at office ", Status.START, Job.OFFICE);
25         Meeting meeting = new Meeting( name: "meeting", Status.PLAN, crabs, hotel);
26         Conversation unknownConversation = new Conversation();
27         Conversation informalConversation = new Conversation();
28
29         //Анонимный класс
30         Purchasable shareOfGiantPlantSociety = new Purchasable() {
31             2 usages
32             Building company = giantPlantSociety;
33             2 usages
34             Integer cost = 150;
35             no usages
36             @Override
37             public void setCompany(Building company) { this.company = company; }
38
39             @Override
40             public String getCompany() { return this.company.getName(); }
41
42             no usages
43             @Override
44             public void setCost(Integer cost) { this.cost = cost; }
45
46             1 usage
47             @Override
48             public Integer getCost() { return cost; }
49
50             @Override
51             public String toString() { return "share"; }
52
53             };
54
55         //4th Tab--
56         try {
57             crabs.startConversation( topic: "", ConversationStyle.UNKNOWN, ConversationStatus.FINISH, extra: "");
58             tell( msg: "The result of this conversation was that the next morning " + Crabs.Status.name + crabs.go(giantPlantSociety) + "
59         } catch (BuildingIsClosedException e) {
60             System.err.println(e.getMessage());
61             System.exit( status: 0);
62         }
63
64         tell( msg: Crabs.Status.name + crabs.buy(shareOfGiantPlantSociety, count: 6) + " of " + shareOfGiantPlantSociety.getCompany());
65         tell(crabs.tell( speech: ""));
66         //--4th Tab
67
68         System.out.println();
69
70         tell( msg: Crabs.Status.name + crabs.invite(new Person[] {migi, julio}, hotel));
71         tell(migi.agree());
72         tell(julio.agree());
73
74
75
```

```

76     tell(migi.interested( thing: "what the famous manufacturer wanted from her"));
77     tell(julio.interested( thing: "what the famous manufacturer wanted from him"));
78     tell(meeting.toString());
79
80
81     work.finishWork();
82
83     tell( msg: "when " + work + "had " + work.getStatus() + "..");
84     try {
85         tell(migi.go(hotel));
86         tell(julio.go(hotel));
87     } catch (BuildingIsClosedException e) {
88         System.err.println(e.getMessage());
89         System.exit( status: 0);
90     }
91     tell( msg: Crabs.Status.name + crabs.tell( speech: "Let's have dinner together"));
92
93     System.out.println("\nIn a minute...\n");
94
95     tell(migi.sit(restaurant, extra: " at a table "));
96     tell(julio.sit(restaurant, extra: " at a table "));
97     tell( msg: Crabs.Status.name + crabs.sit(restaurant, extra: " at a table "));
98
99     tell( msg: Crabs.Status.name + crabs.startConversation( topic: "Grabenberg", ConversationStyle.INFORMAL, ConversationStatus.START, extra: " "));
100    tell( msg: Crabs.Status.name + crabs.inquire( thing: " were in " + grabenberg + " earlier", new Person[]{migi, julio}));
101    tell(migi.tell( speech: "Yes, we were"));
102    tell( msg: Crabs.Status.name + crabs.praise(grabenberg));
103    tell( msg: Crabs.Status.name + crabs.praise(citizens));
104    tell( msg: Crabs.Status.name + crabs.tell( speech: citizens + " are " + citizens.stringQualities()));
105    tell( msg: Crabs.Status.name + crabs.tell( speech: "senior " + sproots + ", who is a native " + grabenberg + "er " + "is also " + citizens));
106 }
107 // class Msg {

```

```

108 @ > public static void tell(String msg) { System.out.println(msg.trim().replaceAll( regex: "{2,}", replacement: " ") + "."); }
111 // }
112 }

```

Результат работы программы

```
The result of this conversation was that the next morning Mr. Crabs went to huge Giant Plant Society office.
Mr. Crabs bough some shares of Giant Plant Society.
Crabs said: "".

Mr. Crabs invited Migi from Flower City and Julio from Flower City to huge hotel.
Migi agreed.
Julio agreed.
Migi was interested what the famous manufacturer wanted from her.
Julio was interested what the famous manufacturer wanted from him.
meeting with Crabs from Grabenberg had planned in the huge hotel.
when work at office had finished..
Migi went to huge hotel.
Julio went to huge hotel.
Mr. Crabs said: "Let's have dinner together".

In a minute...

Migi was sitting at a table in a small restaurant.
Julio was sitting at a table in a small restaurant.
Mr. Crabs was sitting at a table in a small restaurant.
Mr. Crabs started informal conversation about Grabenberg with distant objects.
Mr. Crabs had inquired if Migi from Flower City and Julio from Flower City were in Grabenberg earlier.
Migi said: "Yes, we were".
Mr. Crabs was praising the Grabenberg.
Mr. Crabs was praising the citizens.
Mr. Crabs said: "citizens are the smartest, the kindest, most honest".|
Mr. Crabs said: "senior Sproots, who is a native Grabenberger is also the smartest, the kindest, most honest and the richest".
```

Вывод

Во время выполнения лабораторной работы я научилась работать с обработкой ошибок и исключений. Изучила локальные, анонимные и внутренние классы. Лучше освоила работу с полями, методами, классами и построением UML диаграмм.