**Министерство образования и науки**

**Российской Федерации**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Факультет программной инженерии и компьютерной техники

Дисциплина: Программирование

**Отчёт по лабораторной работе №3**

Вариант 6543234

Выполнил студент группы Р3133 Анисимов Максим Дмитриевич

Проверил Петренко Никита Алексеевич

Санкт-Петербург

2022 г

# Задание:

Доработать программу из [лабораторной работы #3](https://se.ifmo.ru/disciplines/programming#lab3), обновив реализацию объектной модели в соответствии с новой версией описания предметной области.

**Программа должна удовлетворять следующим требованиям:**

1. В программе должны быть реализованы 2 собственных класса исключений (checked и unchecked), а также обработка исключений этих классов.
2. В программу необходимо добавить использование локальных, анонимных и вложенных классов (static и non-static).

**Порядок выполнения работы:**

1. Доработать объектную модель приложения.
2. Перерисовать диаграмму классов в соответствии с внесёнными в модель изменениями.
3. Согласовать с преподавателем изменения, внесённые в модель.
4. Модифицировать программу в соответствии с внесёнными в модель изменениями.

**Отчёт по работе должен содержать:**

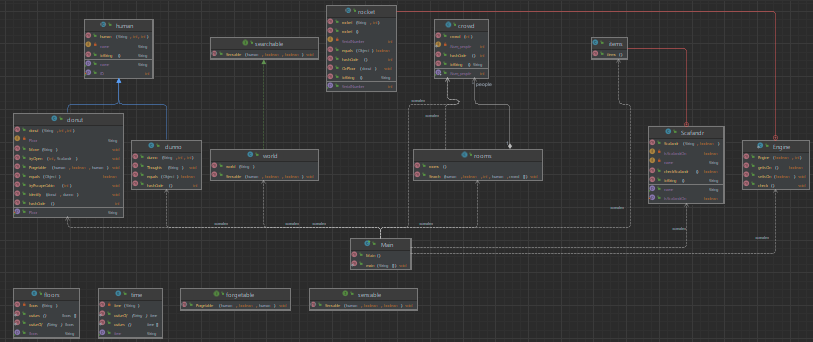
1. Текст задания.
2. Диаграмма классов объектной модели.
3. Исходный код программы.
4. Результат работы программы.
5. Выводы по работе.

# Описание предметной области:

На первых порах Пончик даже не понял, какую страшную он совершил вещь. Почувствовав, что попал в состояние невесомости, он стал делать попытки выкарабкаться из кнопочной кабины, воображая, что в другом каком-нибудь месте состояния невесомости нет. После ряда усилий это ему удалось, и он вернулся обратно к лифту. На этот раз он как следует разобрался в кнопках, которые имелись в кабине лифта, и нажал именно ту, которая обеспечивала спуск кабины на самый нижний этаж, то есть в хвостовую часть ракеты. Выйдя из лифта, он очутился перед дверью в шлюзокамеру, через которую, как уже сказано, можно было выйти наружу. Рядом с дверью Пончик обнаружил на стене кнопку. Однако сколько ни нажимал он на эту кнопку, сколько ни колотил в дверь ногами, дверь и не думала открываться. Пончик не знал, что дверь шлюзокамеры могла открыться лишь в том случае, если бы он надел на себя космический скафандр. И, надо сказать, хорошо, что Пончик этого не знал. Если бы он нажал кнопку, предварительно надев на себя скафандр, дверь отворилась бы и Пончик, покинув ракету, вывалился бы прямо в космическое пространство. Конечно, в этом случае он уже никогда бы не смог вернуться домой, так как остался бы на веки вечные летать в космосе на манер планеты.

   Отбив о дверь кулаки и пятки, Пончик решил вернуться к Незнайке и категорически потребовать, чтобы он выпустил его из ракеты. Это решение он, однако, не мог исполнить, так как забыл, на каком этаже оставил Незнайку. Пришлось ему ездить по всем этажам, лазить по всем кабинетам, каютам, отсекам. Время было позднее. Пончик очень устал и к тому же зверски захотел спать. Можно было бы сказать, что Пончик валился от усталости с ног, если бы он вообще мог стоять на ногах. Из-за состояния невесомости Пончик вообще не имел возможности стоять на ногах, а плавал на манер карася в банке, то и дело стукаясь головой о стены и кувыркаясь в воздухе. В конце концов он вообще перестал что-либо соображать. В голове у него помутилось, глаза стали закрываться сами собой, и, выбившись из последних сил, он заснул как раз в тот момент, когда поднимался в кабине лифта.  
   Незнайка тем временем безмятежно спал в пищевом отсеке и даже не чувствовал, что космический полёт начался. Среди ночи он, однако, проснулся и никак не мог понять, почему находится здесь, а не дома в постели. Постепенно он вспомнил, что нарочно забрался в ракету. Почувствовав невесомость и обратив внимание на мерный шум реактивного двигателя, Незнайка понял, что космический корабль находится в полёте. «Значит, пока я спал, Знайка и остальные коротышки погрузились на корабль и отправились на Луну. Всё получилось точно, как я рассчитал!» – подумал Незнайка

# Диаграмма классов:



# Исходный код:

Main.java – главный класс

import characters.\*;  
import environment.\*;  
  
  
public class Main {  
 public static void main(String[] args) {  
 donut donut = new donut("Пончик",17,12);  
 dunno dunno = new dunno("Незнайка", 17, 11);  
 crowd cr1=new crowd(1);  
 crowd cr2=new crowd(2);  
 crowd cr3=new crowd(3);  
 crowd cr4=new crowd(4);  
 rooms room=new rooms();  
 items.Scafandr sc=new items().new Scafandr("Scafandr-1",false);  
 world world = new world(time.*EARLY\_TIME*.getTime());  
 try{  
 donut.Identify(donut,dunno);  
 }  
 catch(Exception e){  
 e.printStackTrace();  
 }  
 donut.Move(donut.getName()+" решил выбраться из кнопочной кабины");  
 donut.tryEscapeCabin(2);  
 new rocket(){  
 @Override  
 public void OnFloor(characters.donut donut){  
 donut.setFloor(floors.*FIRST\_FLOOR*.getFloors());  
 }  
 }.OnFloor(donut);  
 donut.Move(donut.getName()+" спустился на "+donut.getFloor());  
 donut.tryOpen(2,sc);  
 donut.Move(donut.getName()+" решил вернуться к "+dunno.getName());  
 donut.Forgetable(donut,true,dunno);  
 room.Searchable(donut,true,80,dunno,new crowd[]{cr1,cr2,cr3,cr4});  
 world.Sensable(dunno,true,false);  
dunno.Thoughts("\"Значит, пока я спал, Знайка и остальные коротышки погрузились на корабль и отправились на Луну. Всё получилось точно, как я рассчитал!\" -- подумал "+dunno.getName());  
donut donut2=new donut("Donut2",19,453);  
rocket.Engine Eng1=new rocket.Engine(true,60);  
try{  
 Eng1.check();  
}  
catch(Exception e){  
 e.printStackTrace();  
}  
 }  
 }

Human.java – абстрактный класс, отвечающий за главных героев

import environment.searchable;  
import environment.sensable;  
  
public abstract class human implements forgetable{  
 private String name;  
 private int age;  
 private int id;  
public human(String name, int age, int id){  
 this.name=name;  
 this.age=age;  
 this.id=id;  
}  
  
public String getName(){  
 return name;  
}  
public void setID(int id){  
 this.id=id;  
}  
public int getID(){  
 return id;  
}  
@Override  
 public String toString(){  
 return "Character{" +  
 "name= "+name+  
 ", age= "+age+'}';}  
}

Donut.java – класс, отвечающий за главного героя Пончика

package characters;  
  
import environment.items;  
  
public class donut extends human {  
 private String Floor;  
 public donut(String name, int age, int id){  
 super(name, age,id);  
 }  
 public void Move(String moves){  
 System.*out*.println(moves);  
 }  
 @Override  
 public void Forgetable(human Donut, boolean forget, human Dunno){  
 if(forget){  
 System.*out*.println(Donut.getName()+" забыл, где находится "+Dunno.getName());  
 }  
 else {  
 System.*out*.println(Donut.getName()+" вспомнил, где находится "+Dunno.getName()+" и быстро нашёл его");  
 System.*exit*(0);  
 }  
 }  
 public void Identify(donut donut, dunno dunno) throws Exception{  
 class IsIdDifferent extends Exception {  
 public IsIdDifferent(String message) {  
 super(message);  
 }  
 }  
 if(donut.hashCode()==dunno.hashCode()){  
 throw new IsIdDifferent(donut.getName()+" не может являться "+dunno.getName());  
 }  
 }  
 public void setFloor(String Floor){  
 this.Floor=Floor;  
 }  
 public String getFloor(){  
 return Floor;  
 }  
 public void tryOpen(int Num\_of\_tries, items.Scafandr sc){  
 for(int i=0;i<Num\_of\_tries;i++){  
 if(sc.checkScafandr()){  
 System.*out*.println("Пончик попытался открыть дверь в шлюзе"+'\n'+"Компьютер: \*сканирует наличие скафандра\*");  
 System.*out*.println("Дверь открылась, Пончик в открытом космосе!");  
 System.*exit*(0);  
 }  
 else{  
 System.*out*.println("Пончик попытался открыть дверь в шлюзе"+'\n'+"Компьютер: \*сканирует наличие скафандра\*");  
 System.*out*.println("Отказано в доступе!"+  
 "Причина:"+ sc.getName()+" не надет");  
 }  
 }  
 }  
 public void tryEscapeCabin(int Num\_tries){  
 for(int i=0;i<Num\_tries;i++){  
 if(i==Num\_tries-1){  
 System.*out*.println(this.getName()+" наконец-то смог выбраться из кнопочной кабины ");  
 }  
 else{  
 System.*out*.println(this.getName()+" пыатется выбраться из кнопочной кабины");  
 System.*out*.println(this.getName()+" ударился об потолок из-за гравитации");  
 }  
 }  
 }  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) {  
 return true;  
 }  
 if (o == null || !(getClass() == o.getClass())) {  
 return false;  
 }  
 human human = (characters.human) o;  
 return human.hashCode() == this.hashCode();  
 }  
 @Override  
 public int hashCode(){  
 return getID();  
 }  
}

Dunno.java – класс, отвечающий за главного героя Незнайку

package characters;  
  
public class dunno extends human {  
 public dunno(String name, int age, int id){  
 super(name,age,id);  
 }  
 public void Thoughts(String thought){  
 System.*out*.println(thought);  
 }  
 @Override  
 public void Forgetable(human Donut,boolean forget, human Dunno){  
 if(forget){  
 System.*out*.println(Donut.getName()+" забыл, где находится "+Dunno.getName());  
 }  
 else {  
 System.*out*.println(Donut.getName()+" вспомнил, где находится "+Dunno.getName()+" и быстро нашёл его");  
 System.*exit*(0);  
 }  
  
 }  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) {  
 return true;  
 }  
 if (o == null || getClass() == o.getClass()) {  
 return false;  
 }  
 human human = (characters.human) o;  
 return human.getID() == this.getID();  
 }  
 @Override  
 public int hashCode(){  
 return getID();  
 }  
}

Crowd.java – класс, отвечающий за толпу людей на космическом корабле

package characters;  
  
public class crowd {  
 private int id;  
 private static int *Num\_people*;  
 public crowd(int id){  
 this.id=id;  
this.*Num\_people*=*Num\_people*++;  
 }  
public static int getNum\_people(){  
 return *Num\_people*;  
}  
 @Override  
 public int hashCode(){  
 return id;  
 }  
 @Override  
 public String toString(){  
 return "Crowd{" +  
 "Number of people= "+*Num\_people*+ '}';}  
}

Rocket.java – класс, отвечающий за части ракеты

package environment;  
  
import characters.donut;  
  
public abstract class rocket {  
 private String name;  
 private int SerialNumber;  
 public rocket(String name, int SerialNumber){  
 this.name=name;  
 this.SerialNumber=SerialNumber;  
 }  
 public rocket(){}  
public static class Engine{  
 private boolean IsOn;  
 private int FuelLevel;  
 public Engine(boolean IsOn, int FuelLevel){  
 this.IsOn=IsOn;  
 this.FuelLevel=FuelLevel;  
  
 }  
 public void check() throws Exception{  
 class negFuelNumber extends Exception{  
 public negFuelNumber(String message){  
 super(message);  
 }  
 }  
 if(FuelLevel<0){  
 throw new negFuelNumber("Уровень топлива не может быть ниже 0!");  
 }  
 else if(FuelLevel==0){  
 setisOn(false);  
 System.*out*.println("Уровень топлива: "+FuelLevel);  
 System.*out*.println("Режим работы двигателя (true - работает, false - не работает): "+IsOn);  
 }  
 else {  
 System.*out*.println("Уровень топлива: "+FuelLevel);  
 System.*out*.println("Режим работы двигателя (true - работает, false - не работает): "+IsOn);  
 }  
 }  
 public void setisOn(boolean isOn){  
 this.IsOn=IsOn;  
 }  
 public boolean getisOn(){  
 return IsOn;  
 }  
  
  
}  
 public void OnFloor(donut donut){  
 }  
  
 public int getSerialNumber(){  
 return SerialNumber;  
 }  
 @Override  
 public int hashCode(){  
 return getSerialNumber();  
 }  
 @Override  
 public boolean equals(Object o){  
 if (this==o){  
 return true;  
 }  
 else if(this==null){  
 return false;  
 }  
 else if(getClass()==o.getClass()){  
 return true;  
 }  
 rocket Rocket=(rocket) o;  
 return this.hashCode()==Rocket.hashCode();  
 }  
 @Override  
 public String toString(){  
 return "Rocket{" +  
 "name= "+name+  
 ", serial number= "+getSerialNumber()+'}';}  
 }

Rooms.java – класс, отвечающий за комнаты в локации «ракета»

package environment;  
  
import characters.crowd;  
import characters.human;  
  
import static java.lang.Math.*random*;  
  
public class rooms implements searchable {  
 private crowd[]people=new crowd[crowd.*getNum\_people*()];  
 @Override  
 public void Searchable(human Donut, boolean searching, int Donut\_Energy, human Dunno, crowd[]people)throws RuntimeException{  
 class InCorrectNumber extends RuntimeException {  
 public InCorrectNumber(String message) {  
 super(message);  
 }  
 }  
 int Final\_energy1 = Donut\_Energy;  
 if(Final\_energy1>100||Final\_energy1<10){  
  
 throw new InCorrectNumber("Введено некорректное значение энергии Пончика");}  
  
 for(int i = 0; i< crowd.*getNum\_people*(); i++){  
  
 this.people[i]=people[i];  
 }  
 int rand;  
 if (searching) {  
 for(int i=0;i<people.length;i++){  
  
 if(Final\_energy1>=10){  
  
 Final\_energy1 = Final\_energy1 - 20;  
  
 rand = (int) (3 \* *random*() + 1);  
  
 if (rand==1) {  
  
 System.*out*.println(Donut.getName() + " пошёл на " + floors.*FIRST\_FLOOR*.getFloors());  
  
 } else if (rand==2) {  
  
 System.*out*.println(Donut.getName() + " пошёл на " + floors.*SECOND\_FLOOR*.getFloors());  
  
 } else if (rand==3) {  
  
 System.*out*.println(Donut.getName() + " пошёл на " + floors.*THIRD\_FLOOR*.getFloors());  
 }  
  
 System.*out*.println(Donut.getName() + " пытается найти " + Dunno.getName());  
  
 if(Dunno.hashCode()==people[i].hashCode()){  
 System.*out*.println(Dunno.getName()+" найден");  
 System.*exit*(0);  
 }  
 else{  
 System.*out*.println(Dunno.getName()+" не найден");}  
 System.*out*.println(Donut.getName() + " теряет энергию");  
 }}  
 if (Final\_energy1 <10) {  
 System.*out*.println(time.*LATE\_TIME*.getTime());  
 System.*out*.println(Donut.getName() + " валится с ног от усталости");  
 System.*out*.println(Donut.getName() + " заснул");  
 System.*out*.println(Dunno.getName() + " проснулся в пищевом отсеке");  
  
 }  
 } else {  
 System.*out*.println(Donut.getName() + " не знает, что делать");  
 }  
  
 }}

World.java – класс отвечающий за окружающий мир

package environment;  
  
import characters.human;  
  
public class world implements sensable {  
 private String time;  
 private boolean EngineSound;  
 private boolean Gravity;  
 public world(String time){  
 this.time = time;  
 if (time.equals(environment.time.*LATE\_TIME*.getTime())) {  
 System.*out*.println("Время позднее, Пончик быстро заснул, история закончилась :(");  
 System.*exit*(0);  
 }  
 }  
 @Override  
 public void Sensable(human human, boolean EngineSound, boolean Gravity) {  
 this.EngineSound = EngineSound;  
 this.Gravity = Gravity;  
 if (EngineSound && !Gravity) {  
 System.*out*.println(human.getName() + " догадался, что ракета находится в космосе");  
 } else if (!EngineSound || Gravity) {  
 System.*out*.println((human.getName() + " подумал, что ракета на земле"));  
 }  
 }  
}

Forgetable.java – интерфейс, отвечающий за забывчивость Пончика

package characters;  
  
public interface forgetable {  
 public void Forgetable(Human Donut,boolean forget,Human Dunno);  
}

Searchable.java – интерфейс, отвечающий за поиск Незнайки Пончиком

package characters;  
  
public interface searchable {  
 public void Searchable(Human Donut, boolean searching, int Donut\_Energy, Human Dunno, Crowd[]people);  
}

Sensable.java – интерфейс, отвечающий за чувствтиельность Незнайки к окружающим факторам

package environment;  
import characters.Human;  
public interface sensable {  
 public void Sensable(Human human,boolean EngineSound,boolean Gravitation);  
}

Floors.java – перечисление этажей в ракете

package environment;  
  
public enum Floors {  
 *FIRST\_FLOOR*("Первый этаж"),  
 *SECOND\_FLOOR*("Второй этаж"),  
 *THIRD\_FLOOR*("Третий этаж");  
 private String name;  
 Floors(String name){  
 this.name=name;  
 }  
 public String getFloors(){  
 return name;  
 }  
}

Time.java – перечисление времён

package environment;  
  
public enum Time {  
 *EARLY\_TIME*("Время раннее"),  
 *LATE\_TIME*("Время стало позднее");  
 private String name;  
 Time(String name){  
 this.name=name;  
 }  
 public String getTime(){  
 return name;  
 }  
}

# Результат программы:

Пончик решил выбраться из кнопочной кабины

Пончик пыатется выбраться из кнопочной кабины

Пончик ударился об потолок из-за гравитации

Пончик наконец-то смог выбраться из кнопочной кабины

Пончик спустился на Первый этаж

Пончик попытался открыть дверь в шлюзе

Компьютер: \*сканирует наличие скафандра\*

Отказано в доступе!Причина:Scafandr-1 не надет

Пончик попытался открыть дверь в шлюзе

Компьютер: \*сканирует наличие скафандра\*

Отказано в доступе!Причина:Scafandr-1 не надет

Пончик решил вернуться к Незнайка

Пончик забыл, где находится Незнайка

Пончик пошёл на Третий этаж

Пончик пытается найти Незнайка

Незнайка не найден

Пончик теряет энергию

Пончик пошёл на Первый этаж

Пончик пытается найти Незнайка

Незнайка не найден

Пончик теряет энергию

Пончик пошёл на Первый этаж

Пончик пытается найти Незнайка

Незнайка не найден

Пончик теряет энергию

Пончик пошёл на Третий этаж

Пончик пытается найти Незнайка

Незнайка не найден

Пончик теряет энергию

Время стало позднее

Пончик валится с ног от усталости

Пончик заснул

Незнайка проснулся в пищевом отсеке

Незнайка догадался, что ракета находится в космосе

"Значит, пока я спал, Знайка и остальные коротышки погрузились на корабль и отправились на Луну. Всё получилось точно, как я рассчитал!" -- подумал Незнайка

Уровень топлива: 60

Режим работы двигателя (true - работает, false - не работает): true