**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: 3](#_Toc126967378)

[Описание предметной области: 3](#_Toc126967379)

[Диаграмма классов: 4](#_Toc126967380)

[Исходный код: 4](#_Toc126967381)

[Результат программы: 9](#_Toc126967382)

[Вывод: 10](#_Toc126967383)

# 

# Задание:

**Программа должна удовлетворять следующим требованиям:**

1. Доработанная модель должна соответствовать [принципам SOLID](https://en.wikipedia.org/wiki/SOLID_(object-oriented_design)).
2. Программа должна содержать как минимум два интерфейса и один абстрактный класс (номенклатура должна быть согласована с преподавателем).
3. В разработанных классах должны быть переопределены методы equals(), toString() и hashCode().
4. Программа должна содержать как минимум один перечисляемый тип (enum).

**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:

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

# Диаграмма классов:

Изображение выглядит как текст, внутренний

Автоматически созданное описание

# Исходный код:

Main.java – главный класс

import characters.\*;  
import environment.\*;  
  
  
public class Main {  
 public static void main(String[] args) {  
  
Dunno dunno=new Dunno("Незнайка",17,1);  
  
Donut donut=new Donut("Пончик",17,2);  
donut.Identify(donut,dunno);  
  
 World world=new World(Time.*EARLY\_TIME*.getTime());  
  
 Rocket rocket = new Rocket();  
  
 Crowd cr1=new Crowd(0);  
 Crowd cr2=new Crowd(7);  
 Crowd cr3=new Crowd(3);  
 Crowd cr4=new Crowd(4);  
  
 Rooms rooms=new Rooms(Floors.*FIRST\_FLOOR*.getFloors());  
  
 donut.Thoughts(donut.getName()+" решил найти "+dunno.getName());  
  
 dunno.Forgetable(donut,true,dunno);  
  
 rooms.Searchable(donut,true,80,dunno,new Crowd[]{cr1,cr2,cr3,cr4});  
  
 dunno.Thoughts(dunno.getName()+" вспомнил, что нарочно забрался в ракету");  
  
 rocket.Sensable(dunno,true,false);  
  
 dunno.Thoughts("\"Значит, пока я спал, Знайка и остальные коротышки погрузились в корабль и отправились на Луну. Всё получилось точно, как я рсчитал!\" -- подумал "+dunno.getName());  
 }  
}

Human.java – абстрактный класс, отвечающий за главных героев

package characters;  
  
public abstract class Human implements Thoughts, Forgetable{  
 private String name;  
 private int age;  
public Human(String name, int age){  
 this.name=name;  
 this.age=age;  
}  
public String getName(){  
 return name;  
}  
public int getAge(){return age;}  
  
@Override  
 public String toString(){  
 return "Character{" +  
 "name= "+name+  
 ", age= "+age+'}';}  
}

Donut.java – класс, отвечающий за главного героя Пончика

package characters;  
  
public class Donut extends Human{  
 public Donut(String name,int age,int id){  
 super(name, age,id);  
 }  
 public void Thoughts(String thoughts){  
 System.*out*.println(thoughts);  
 }  
 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){  
 if(donut.hashCode()==dunno.hashCode()){  
 System.*out*.println(donut.getName()+" не может являться "+dunno.getName());  
 System.*exit*(0);  
 }  
 }  
 public boolean equals(Object o) {  
 if (this == o) {  
 return true;  
 }  
 if (o == null || getClass() == o.getClass()) {  
 return false;  
 }  
 Human human = (Human) o;  
 return human.getID() == this.getID();  
 }  
 @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 thoughts){  
 System.*out*.println(thoughts);  
 }  
 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 boolean equals(Object o) {  
 if (this == o) {  
 return true;  
 }  
 if (o == null || getClass() == o.getClass()) {  
 return false;  
 }  
 Human human = (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*++;  
 }  
  
@Override  
public int hashCode(){  
 return id;  
}  
public static int getNum\_people(){  
 return *Num\_people*;  
}  
}

Rocket.java – класс, отвечающий за части ракеты

package environment;  
  
import characters.Human;  
  
public class Rocket implements Sensable{  
 private boolean EngineSound;  
 public boolean Gravity;  
 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"));  
 System.*exit*(0);  
 }  
 }  
}

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

package environment;  
  
import characters.\*;  
  
import static java.lang.Math.*random*;  
  
public class Rooms implements Searchable {  
 private String Which\_floor;  
  
public Rooms(String Which\_floor) {  
 this.Which\_floor=Which\_floor;  
}  
 private Crowd[]people=new Crowd[Crowd.*getNum\_people*()];  
 public void Searchable(Human Donut, boolean searching, int Donut\_Energy, Human Dunno, Crowd[]people) {  
  
 int Final\_energy1 = Donut\_Energy;  
  
 if(Final\_energy1>100||Final\_energy1<10){  
  
 System.*out*.println("Пожалуйста, введите значение энергии Пончика от 10 до 100");  
  
 System.*exit*(0);}  
  
 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 - 10;  
  
 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 время суток на корабле

package environment;  
  
public class World {  
 private String time;  
 public World( String time) {  
 this.time = time;  
 if (time.equals(Time.*LATE\_TIME*.getTime())) {  
 System.*out*.println("Время позднее, Пончик быстро заснул, история закончилась :(");  
 System.*exit*(0);  
 }  
 }  
  
}

Forgetable.java – интерфейс, отвечающий за забывчивость Пончика

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

Identify.java – интерфейс, отвечающий за проверку уникального id персонажей

package characters;  
  
public interface Identify {  
 public void Identify(Donut donut,Dunno 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);  
}

Thoughts.java – интерфейс, отвечающий за мысли персонажей

package characters;  
  
public interface Thoughts {  
 public void Thoughts(String thought);  
}

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 найти Незнайка

Пончик забыл, где находится Незнайка

Пончик пошёл на Первый этаж

Пончик пытается найти Незнайка

Незнайка не найден

Пончик теряет энергию

Пончик пошёл на Второй 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

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 вспомнил, что нарочно забрался в ракету

Незнайка догадался, что ракета находится в космосе

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

# Вывод:

В процессе выполнения лабораторной работы были получены навыки использования объектно-ориентированного подхода программирования при использовании языка Java.