Федеральное государственное автономное образовательное учреждение высшего образования **«Национальный исследовательский университет ИТМО»**

Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа по программированию №4**

вариант: 852

Преподаватель: Харитонова Анастасия Евгеньевна

Выполнил: Дагаев Даниил Сергеевич

Группа: Р3108

Санкт-Петербург, 2021г

Оглавление

[Задание 2](#_Toc89011785)

[Диаграмма 2](#_Toc89011786)

[Исходный код программы 2](#_Toc89011787)

[Результат работы программы 3](#_Toc89011788)

[Вывод 3](#_Toc89011789)

# Задание

**Описание предметной области, по которой должна быть построена объектная модель:**

С тех пор как Пончик стал работать крутильщиком на чертовом колесе, его характер сильно переменился. Раньше он жил без всяких забот: ел да пил, а в свободное от еды время слонялся по набережной и вертелся на чертовом колесе или морском параболоиде, не задаваясь мыслью о том, какая сила приводит все эти механизмы в движение. Но теперь он на собственном опыте убедился, что никакое чертово колесо само по себе вертеться не будет, если его не начнут вертеть коротышки. Как уже говорилось, каждое чертово колесо представляло собой круг или диск, насаженный на вертикальную ось. Этот диск устанавливался на огромной круглой лохани, плавающей неподалеку от побережья и закрепленной на якорях. Лохань, прикрытая сверху диском, погружалась больше чем наполовину в воду, так что ее почти и видно-то не было. Со стороны казалось, что огромнейший деревянный диск как бы сам собой крутится над водой.

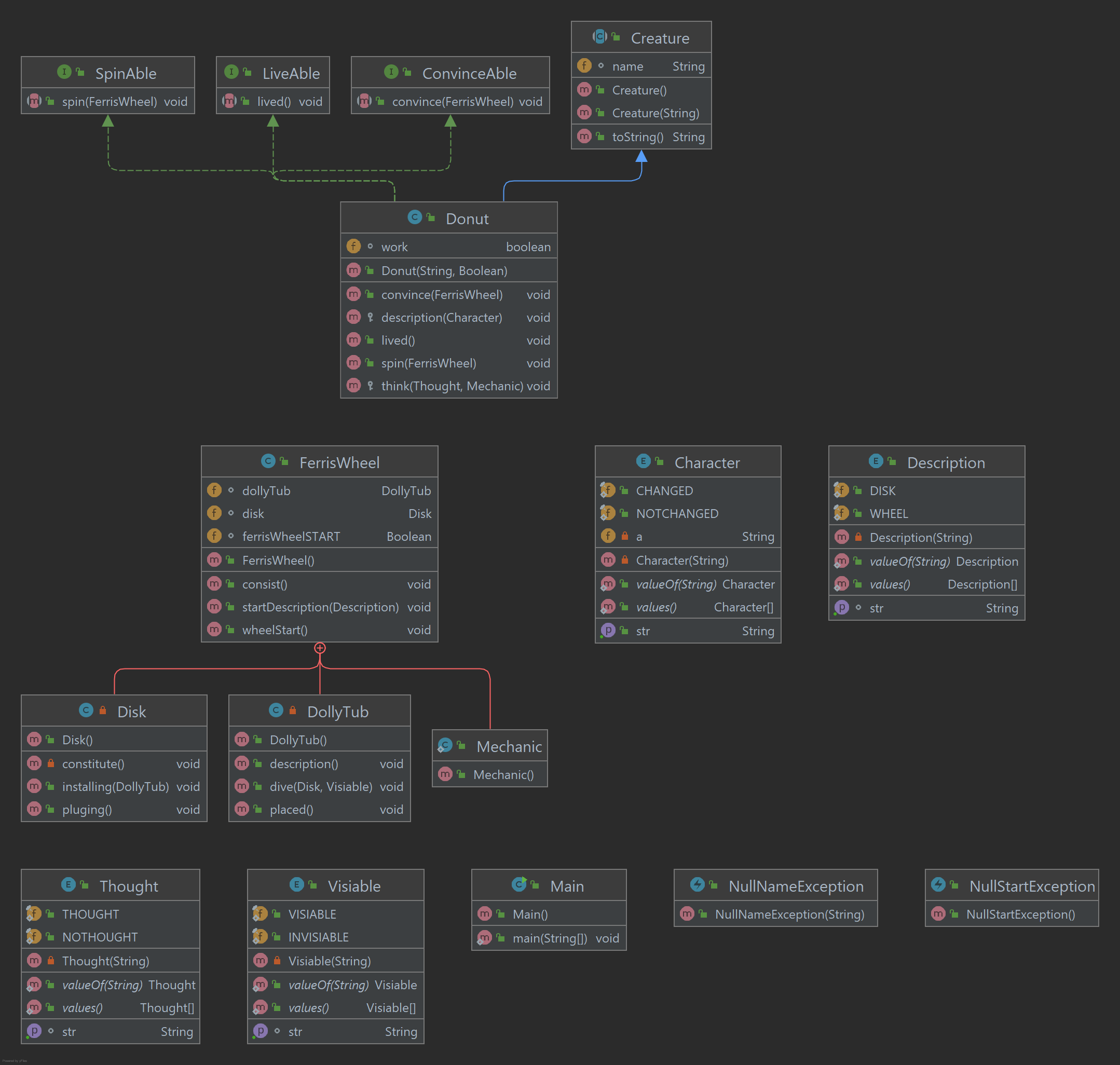
**Программа должна удовлетворять следующим требованиям:**

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.

# Диаграмма



# Исходный код программы

import java.io.FileInputStream;  
import java.io.FileNotFoundException;  
  
public class Main {  
 public static void main(String[] args) throws NullNameException {  
 FerrisWheel ferrisWheel = new FerrisWheel();  
 Donut Ponchik = new Donut("Пончик",true);  
 Ponchik.description(Character.*CHANGED*);  
 Ponchik.lived();  
 Ponchik.spin(ferrisWheel);  
 FerrisWheel.Mechanic mechanic= new FerrisWheel.Mechanic();  
 Ponchik.think(Thought.*NOTHOUGHT*,mechanic);  
 Ponchik.convince(ferrisWheel);  
 SpinAble Korotishki = new SpinAble() {  
 @Override  
 public void spin(FerrisWheel ferrisWheel) {  
 System.*out*.print(", если его не начнут вертеть коротышки.");  
 }  
 };  
 Korotishki.spin(ferrisWheel);  
 ferrisWheel.consist();  
 ferrisWheel.wheelStart();  
 try {  
 ferrisWheel.startDescription(Description.*DISK*);  
 }catch (NullPointerException e){  
 System.*err*.print("\n"+e+"\nFerrisWheel didn't start");  
 }  
 try {  
 FileInputStream fileInputStream = new FileInputStream("");  
 } catch (FileNotFoundException e) {  
 System.*err*.println("File Not found");  
 }  
 }  
}

public class Donut extends Creature implements LiveAble,ConvinceAble,SpinAble{  
 public Donut(String name,Boolean work) throws NullNameException {  
 super.name = name;  
 this.work = work;  
 if(name == null || name.equals("")){  
 throw new NullNameException("Строка пустая или null.");  
 }  
  
 }  
 boolean work;  
 protected void description(Character character){  
 if(work) System.*out*.println("С тех пор как "+ name +" стал работать крутильщиком на чертовом колесе, "+ character.getStr());  
 else System.*out*.println("Пончик не работал");  
 }  
 public void lived(){  
 System.*out*.print("Раньше он жил без всяких забот:");  
 class FreeTime{  
 String eat =" ел да пил,";  
 public void eating(){  
 System.*out*.println(eat + "а в свободное от еды время слонялся по набережной и");  
 }  
 }  
 FreeTime freeTime = new FreeTime();  
 freeTime.eating();  
 }  
 public void spin(FerrisWheel ferrisWheel){  
 System.*out*.println("вертелся на чертовом колесе или морском параболоиде,");  
 }  
 protected void think(Thought thought,FerrisWheel.Mechanic mechanic){  
 System.*out*.println(thought.getStr() + ", какая сила приводит все эти механизмы в движение.");  
 }  
 public void convince(FerrisWheel ferrisWheel){  
 if(work)System.*out*.print("Но теперь он на собственном опыте убедился, что никакое чертово колесо само по себе вертеться не будет\n");  
 }  
}

public class FerrisWheel {  
 DollyTub dollyTub;  
 Disk disk;  
 Boolean ferrisWheelSTART;  
 public FerrisWheel(){  
 dollyTub = new DollyTub();  
 disk = new Disk();  
 }  
 private class Disk{  
 private void constitute(){  
 System.*out*.print("Как уже говорилось, каждое чертово колесо представляло собой круг или диск");  
 }  
 public void pluging(){  
 System.*out*.print(", насаженный на вертикальную ось.");  
 }  
 public void installing(DollyTub dollyTub){  
 System.*out*.print("\nЭтот диск устанавливался на огромной круглой лохани");  
 }  
 }  
 public static class Mechanic{  
 }  
 public void consist(){  
 disk.constitute();  
 disk.pluging();  
 }  
 private class DollyTub{  
 public void description(){  
 System.*out*.println("Лохань, прикрытая сверху диском");  
 }  
 public void placed(){  
 System.*out*.print(", плавающей неподалеку от побережья и закрепленной на якорях.\n");  
 }  
 public void dive(Disk disk,Visiable visiable){  
 System.*out*.println("Лохань, прикрытая сверху диском, погружалась больше чем наполовину в воду, так что"+ visiable.getStr()+".");  
 }  
 }  
 public void wheelStart(){  
 disk.installing(dollyTub);  
 dollyTub.placed();  
 dollyTub.dive(disk,Visiable.*INVISIABLE*);  
 ferrisWheelSTART = true;  
 }  
 public void startDescription(Description description){  
 if(ferrisWheelSTART){  
 System.*out*.println("Со стороны казалось, что "+description.getStr()+".");  
 }  
 }  
  
}

public abstract class Creature {  
 String name;  
 @Override  
 public String toString() {  
 return this.name;  
 }  
 public Creature(String name){  
 this.name =name;  
 }  
 public Creature(){  
  
 }  
}

public interface ConvinceAble {  
 void convince(FerrisWheel ferrisWheel);  
}

public interface LiveAble {  
 void lived();  
}

public interface SpinAble {  
 void spin(FerrisWheel ferrisWheel);  
}

public enum Character {  
 *CHANGED*("его характер сильно переменился. "),*NOTCHANGED*("его характер вообше не менялся");  
 private String a;  
 Character(String str){  
 a=str;  
 }  
 public String getStr(){  
 return a;  
 }  
}

public enum Description {  
 *DISK*("огромнейший деревянный диск как бы сам собой крутится над водой"),*WHEEL*("колесо вращалось каким-то волшебным образом");  
 String str;  
 Description(String str){  
 this.str=str;  
 }  
 String getStr(){  
 return str;  
 }  
}

public enum Thought {  
 *THOUGHT*("думая о том"),*NOTHOUGHT*("не задаваясь мыслью о том");  
 String str;  
 Thought(String str){  
 this.str=str;  
 }  
 String getStr(){  
 return str;  
 }  
}

public enum Visiable {  
 *VISIABLE*("она была видна всем"),*INVISIABLE*(" ее почти и видно-то не было");  
 String str;  
 Visiable(String str){  
 this.str = str;  
 }  
 String getStr(){  
 return str;  
 }  
}

# Результат работы программы

С тех пор как Пончик стал работать крутильщиком на чертовом колесе, его характер сильно переменился.

Раньше он жил без всяких забот: ел да пил,а в свободное от еды время слонялся по набережной и

вертелся на чертовом колесе или морском параболоиде,

не задаваясь мыслью о том, какая сила приводит все эти механизмы в движение.

Но теперь он на собственном опыте убедился, что никакое чертово колесо само по себе вертеться не будет

, если его не начнут вертеть коротышки. Как уже говорилось, каждое чертово колесо представляло собой круг или диск, насаженный на вертикальную ось.

Этот диск устанавливался на огромной круглой лохани, плавающей неподалеку от побережья и закрепленной на якорях.

Лохань, прикрытая сверху диском, погружалась больше чем наполовину в воду, так что ее почти и видно-то не было.

Со стороны казалось, что огромнейший деревянный диск как бы сам собой крутится над водой.

File Not found

Process finished with exit code 0

# Вывод

В процессе выполнения лабораторной работы, я изучил обрабатывать исключения, изучил разные виды классов(абстрактные, вложенные и локальные) и хочу спать .