Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”

Факультет ПИиКТ



ОТЧЁТ

По лабораторной работе №4

Вариант 3985

Работу выполнил:

Андрейченко Леонид Вадимович

Группа P3130

Преподаватель:

Николаев Владимир Вячеславович

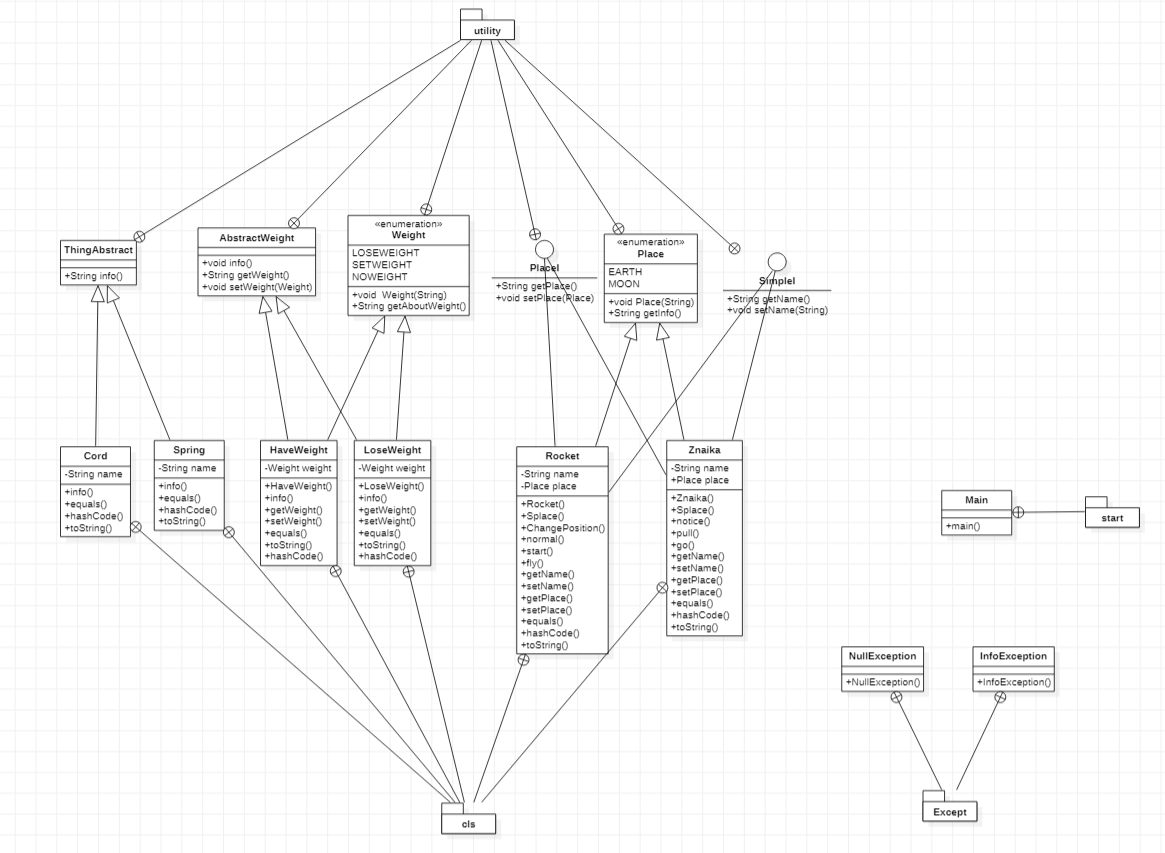
Санкт – Петербург

2021

Описание предметной области

Как известно, все предметы, теряя вес, обычно поднимаются вверх (если они, конечно, не закреплены). Ведь, находясь под действием силы тяжести, каждый предмет как бы сжимается или сплющивается хотя бы на самую ничтожную величину. Но как только предмет потеряет вес, он разжимается, выпрямляется, в результате чего отталкивается, как пружина, от поверхности, на которой до этого неподвижно стоял. Заметив, что ракета поднялась на достаточную высоту, Знайка потихонечку потянул за шнур и не спеша зашагал по лунной дорожке. Ракета приняла горизонтальное положение и послушно поплыла над поверхностью Луны. Правда, по временам она опускалась, но, едва коснувшись поверхности Луны, отталкивалась от нее и снова поднималась вверх.

UML диаграмма



Исходный код

Start.Main

package start;  
  
import Except.WeightException;  
import Except.ZeroException;  
import cls.\*;  
import utility.\*;  
  
public class Main {  
  
 public static void main(String[] args) throws WeightException, ZeroException {  
 int x =5;  
 ThingA thingA = new ThingA(){  
 @Override  
 public String info() {  
 return "Предмет";  
 }  
 };  
 Cord cord = new Cord("шнур");  
 Spring spring = new Spring("пружина");  
 HaveWeight haveWeight = new HaveWeight((byte) 1);  
 LoseWeight loseWeight =new LoseWeight((byte) -1);  
 Rocket rocket = new Rocket("Ракета",false);  
 Znaika znaika = new Znaika("Знайка", false);  
  
 System.out.println(loseWeight.info(cord.info()));  
 System.out.println(haveWeight.info(thingA.info()));  
 znaika.notice("ракета поднялась на достаточную высоту.");  
 znaika.pull(cord.info());  
 znaika.go("лунной дорожке");  
  
 Rocket.ChangePosition changePosition = rocket.new ChangePosition("горизонтальное");  
 changePosition.changeposition();  
  
 rocket.start();  
 rocket.fly();  
 try {  
 if (1/x!=0){  
 System.out.println("Конец");  
 }  
 }catch (ArithmeticException e){  
 throw new ZeroException("Please, enter not 0");  
 }  
 }  
}

Utility.Place

package utility;  
  
public interface Place {  
 boolean getPlace();  
 void setPlace(boolean s);  
}

Utility.Weight

package utility;  
  
public interface Weight {  
 byte getWeight();  
 void setWeight(byte s);  
 String info(String s);  
}

Utility.Simple

package utility;  
  
public interface Simple {  
 String getName();  
 void setName(String s);  
}

Utility.ThingA

package utility;  
  
public class ThingA {  
 public String info(){  
 return "";  
 }  
}

Cls.Cord

package cls;  
  
import utility.ThingA;  
  
public class Cord extends ThingA {  
 private String name;  
 public Cord(String s){  
 name=s;  
 }  
 @Override  
 public String info() {  
 return name;  
 }  
}

Cls.HaveWeight

{  
 private byte weight;  
  
 public HaveWeight( byte weight) throws WeightException {  
 if (weight>1 | weight<-1){  
 throw new WeightException("IncorrectWeightSize");  
 }  
 this.weight=weight;  
 }  
  
 @Override  
 public String info(String s) {  
 byte w = weight;  
 class Info{  
 public String AboutWeight(){  
 if (w>0){  
 return(s+", находясь под действием силы тяжести, как бы сжимается или сплющивается хотя бы на самую ничтожную величину");  
 }  
 else if (w==0){  
 return("Предмет без массы");  
 }  
 else{  
 return ("Предметы, теряя вес, обычно поднимаются вверх (если они, конечно, не закреплены). Они разжимаются, выпрямляются, в результате чего отталкиваются, как "+s+", от поверхности, на которой до этого неподвижно стоял.");  
 }  
 }  
 }  
 Info i = new Info();  
 return(i.AboutWeight());  
 }  
  
 @Override  
 public byte getWeight() {  
 return weight;  
 }  
  
 @Override  
 public void setWeight(byte s) {  
 this.weight=s;  
 }  
}

Cls.LoseWeight

package cls;  
  
import Except.WeightException;  
import utility.Weight;  
  
public class LoseWeight implements Weight {  
 private byte weight;  
  
 public LoseWeight(byte weight) throws WeightException {  
 if (weight>1 | weight<-1){  
 throw new WeightException("IncorrectWeightSize");  
 }  
 this.weight=weight;  
 }  
  
 @Override  
 public String info(String s) {  
 byte w = weight;  
 class Info{  
 public String AboutWeight(){  
 if (w>0){  
 return(s+", находясь под действием силы тяжести, как бы сжимается или сплющивается хотя бы на самую ничтожную величину");  
 }  
 else if (w==0){  
 return("Предмет без массы");  
 }  
 else{  
 return ("Предметы, теряя вес, обычно поднимаются вверх (если они, конечно, не закреплены). Они разжимаются, выпрямляются, в результате чего отталкиваются, как "+s+", от поверхности, на которой до этого неподвижно стоял.");  
 }  
 }  
 }  
 Info i = new Info();  
 return(i.AboutWeight());  
 }  
  
 @Override  
 public byte getWeight() {  
 return weight;  
 }  
  
 @Override  
 public void setWeight(byte s) {  
 this.weight=s;  
 }  
  
  
}

Cls.Spring

package cls;  
  
import utility.ThingA;  
  
public class Spring extends ThingA {  
 private String name;  
 public Spring(String s){  
 name = s;  
 }  
 @Override  
 public String info() {  
 return name;  
 }  
}

Cls.Rocket

package cls;  
  
import utility.Place;  
import utility.Simple;  
  
public class Rocket implements Simple, Place {  
  
 private String name;  
 static boolean place;  
  
 public Rocket(String s, boolean p){  
 name=s;  
 place=p;  
 }  
  
 static class PlaceInfo{  
 static String info(){  
 if (place==true){  
 return "Земле";  
 }  
 else{  
 return "Луны";  
 }  
 }  
 }  
  
 public PlaceInfo info(boolean place){  
 return new PlaceInfo();  
 }  
  
 public class ChangePosition{  
 private String position;  
  
 public ChangePosition(String position){  
 this.position=position;  
 }  
  
 public void changeposition(){  
 System.out.println(getName()+", поменяла положение на "+position);  
 }  
 }  
  
 public void start(){  
 System.out.println(getName()+", послушно поплыла над поверхностью "+PlaceInfo.info());  
 }  
  
 public void fly(){  
 System.out.println(getName()+ " по временам опускалась, но, едва коснувшись поверхности "+PlaceInfo.info()+", отталкивалась от нее и снова поднималась вверх.");  
 }  
  
 @Override  
 public String getName() {  
 return name;  
 }  
  
 @Override  
 public void setName(String s) {  
 this.name=s;  
 }  
  
 @Override  
 public boolean getPlace() {  
 return place;  
 }  
  
 @Override  
 public void setPlace(boolean s) {  
 place=s;  
 }  
}

Cls.Znaika

package cls;  
  
import utility.Place;  
import utility.Simple;  
  
  
  
public class Znaika implements Simple, Place {  
  
 private String name;  
 static boolean place;  
  
 public Znaika(String s, boolean p){  
 name=s;  
 place=p;  
 }  
  
 static class PlaceInfo{  
 String info(){  
 if (place==true){  
 return "Земле";  
 }  
 else{  
 return "Луны";  
 }  
 }  
 }  
  
 public PlaceInfo info(boolean place){  
 return new Znaika.PlaceInfo();  
 }  
  
 public void notice(String s){  
 System.out.println(getName()+", заметил, что "+s);  
 }  
  
 public void pull(String s){  
 System.out.println(getName()+", потянул за "+s);  
 }  
  
 public void go(String s){  
 System.out.println(getName()+", зашагал по "+s);  
 }  
  
 @Override  
 public String getName() {  
 return name;  
 }  
  
 @Override  
 public void setName(String s) {  
 this.name=s;  
 }  
  
 @Override  
 public boolean getPlace() {  
 return place;  
 }  
  
 @Override  
 public void setPlace(boolean s) {  
 place=s;  
 }  
}

Except.WeightException

package Except;  
  
public class WeightException extends Exception{  
  
 public WeightException(String message) {  
 System.err.println(message);  
 System.err.println("Please use only -1,0,1");  
 }  
}

Except.ZeroException

package Except;  
  
public class ZeroException extends Exception{  
  
 public ZeroException(String message) {  
 System.err.println(message);  
 System.err.println("Сannot be divisible by 0");  
 }  
}

Результат работы

Предметы, теряя вес, обычно поднимаются вверх (если они, конечно, не закреплены). Они разжимаются, выпрямляются, в результате чего отталкиваются, как шнур, от поверхности, на которой до этого неподвижно стоял.

Предмет, находясь под действием силы тяжести, как бы сжимается или сплющивается хотя бы на самую ничтожную величину

Знайка, заметил, что ракета поднялась на достаточную высоту.

Знайка, потянул за шнур

Знайка, зашагал по лунной дорожке

Ракета, поменяла положение на горизонтальное

Ракета, послушно поплыла над поверхностью Луны

Ракета по временам опускалась, но, едва коснувшись поверхности Луны, отталкивалась от нее и снова поднималась вверх.

Выводы

Во время выполнения данной лабораторной работы я закрепил принципы SOLID, научился создавать собственные исключения и пользоваться вложенными, внутренними, локальными и анонимными классами. Не выспался.