НИУ ИТМО

Факультет программной инженерии и компьютерной техники

**ЛАБОРАТОРНАЯ РАБОТА № 2**

по дисциплине

‘ПРОГРАММИРОВАНИЕ’

Вариант № 121010

*Выполнил:*

Студент группы P3110

Абрабоу Ахмед Елсаид А.И

*Преподаватель:*Сорокин Роман Борисович



Санкт-Петербург, 2021

***Задание:***

***Диограмма:***

***GitHub//:***

***Код программы:***

***Class Main***import buildings.Factory;  
import characters.\*;  
  
public class Main {  
 public static void main(String[] args) {  
 Police police = new Police();  
 Villagers villagers = new Villagers();  
 Astronaut astronaut = new Astronaut();  
 Znayka znayka = new Znayka();  
 FactoryWorkers factoryWorkers = new FactoryWorkers();  
 Scooperfield scooperfield = new Scooperfield();  
 Factory factory = new Factory();  
 System.*out*.print("Нечего и говорить, что ");  
 police.fearOfRocket();  
 villagers.goToAstronauts(astronaut.getName());  
 villagers.getSeedsFromAstronaut(astronaut.giveSeedsToVillagers());  
 villagers.tellAboutPlantingSites();  
 znayka.giveOrder();  
 villagers.getWeightlessnessDevicesFromAstronaut(astronaut.giveWeightlessnessDevicesToVillagers());  
 villagers.getAntiluniteFromAstronaut(astronaut.giveAntiluniteToVillagers());  
 villagers.getExplainFromAstronaut(astronaut.explainHowToUse());  
 factoryWorkers.arriveAtTheAstronauts();  
 factoryWorkers.tellWhatTheyDecided();  
 factoryWorkers.thinkAboutTheyDecided();  
 }  
}

***Classes of package Characters***

package characters;  
  
import resources.Antilunite;  
import resources.Seeds;  
import resources.WeightlessnessDevices;  
  
import java.util.Objects;  
  
public class Astronaut implements AstronautHelp {  
 private final String name = "космонавтам";  
 private final String instruction = "Пользуйтесь так!";  
 private final Seeds seeds = new Seeds();  
 private final WeightlessnessDevices weightlessnessDevices = new WeightlessnessDevices();  
 private final Antilunite antilunite = new Antilunite();  
  
 public String getName() {  
 return name;  
 }  
  
 @Override  
 public WeightlessnessDevices giveWeightlessnessDevicesToVillagers() {  
 return weightlessnessDevices;  
 }  
  
 @Override  
 public Seeds giveSeedsToVillagers() {  
 return seeds;  
 }  
  
 @Override  
 public Antilunite giveAntiluniteToVillagers() {  
 return antilunite;  
 }  
  
 public String explainHowToUse() {  
 return instruction;  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Astronaut astronaut = (Astronaut) o;  
 return Objects.*equals*(name, astronaut.name) && Objects.*equals*(seeds, astronaut.seeds) && Objects.*equals*(weightlessnessDevices, astronaut.weightlessnessDevices) && Objects.*equals*(antilunite, astronaut.antilunite);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(name, seeds, weightlessnessDevices, antilunite);  
 }  
  
 @Override  
 public String toString() {  
 return "Astronaut{" +  
 "name='" + name + '\'' +  
 ", seeds=" + seeds +  
 ", weightlessnessDevices=" + weightlessnessDevices +  
 ", antilunite=" + antilunite +  
 '}';  
 }  
}

package characters;  
  
import resources.Antilunite;  
import resources.Seeds;  
import resources.WeightlessnessDevices;  
  
public interface AstronautHelp {  
 WeightlessnessDevices giveWeightlessnessDevicesToVillagers();  
  
 Seeds giveSeedsToVillagers();  
  
 Antilunite giveAntiluniteToVillagers();  
}

package characters;  
  
import java.util.Objects;  
  
public class FactoryWorkers extends Villagers {  
 private final String placeOfWork = "скуперфильдовской макаронной фабрики.";  
  
 public void arriveAtTheAstronauts() {  
 System.*out*.println("Вскоре к космонавтам прибыли несколько рабочих со " + placeOfWork);  
 }  
  
 public String sayPlan() {  
 return "прогнать с фабрики Скуперфильда, а макароны будут делать сами без всяких хозяев";  
 }  
  
 public void tellWhatTheyDecided() {  
 System.*out*.println("Они сказали, что решили " + sayPlan() + ".");  
 }  
  
 public void thinkAboutTheyDecided() {  
 System.*out*.println("Чтоб осуществить этот план, им нужно устроить на фабрике невесомость, " +  
 "так как в противном случае полицейские могут помешать им и даже вовсе прогонят их с фабрики.");  
 }  
  
 public boolean arrangeWeightlessness() {  
 return true;  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 FactoryWorkers that = (FactoryWorkers) o;  
 return Objects.*equals*(placeOfWork, that.placeOfWork);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(placeOfWork);  
 }  
  
 @Override  
 public String toString() {  
 return "FactoryWorkers{" +  
 "placeOfWork='" + placeOfWork + '\'' +  
 '}';  
 }  
}

package characters;  
  
import resources.Antilunite;  
import resources.Seeds;  
import resources.WeightlessnessDevices;  
  
public interface GetHelpFromAstronaut {  
 void getWeightlessnessDevicesFromAstronaut(WeightlessnessDevices weightlessnessDevices);  
  
 void getAntiluniteFromAstronaut(Antilunite antilunite);  
  
 void getSeedsFromAstronaut(Seeds seeds);  
}

package characters;  
  
import java.util.Objects;  
  
public class Police {  
 private final String policeName = "полицейские";  
  
 public void fearOfRocket() {  
 System.*out*.println(policeName + " боялись теперь и близко " + goToRocket() + ", а не то что " + shootNearRocket() + ".");  
 }  
  
 public String goToRocket() {  
 return "подходить к ракете";  
 }  
  
 public String shootNearRocket() {  
 return "стрелять возле неё";  
 }  
  
 public void preventEvent() {  
 }  
  
 public void banishFromFactory() {  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Police police = (Police) o;  
 return Objects.*equals*(policeName, police.policeName);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(policeName);  
 }  
  
 @Override  
 public String toString() {  
 return "Police{" +  
 "policeName='" + "полиция" + '\'' +  
 '}';  
 }  
}

package characters;  
  
import java.util.Objects;  
  
public class Scooperfield {  
 private String placeOfWork = "Директор макаронной фабрики";  
 private String name = "Скуперфильд";  
  
 public void banishByWorkers() {  
 placeOfWork = "безработный";  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Scooperfield that = (Scooperfield) o;  
 return Objects.*equals*(placeOfWork, that.placeOfWork) && Objects.*equals*(name, that.name);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(placeOfWork, name);  
 }  
  
 @Override  
 public String toString() {  
 return "Scooperfield{" +  
 "placeOfWork='" + placeOfWork + '\'' +  
 ", name='" + name + '\'' +  
 '}';  
 }  
}

package characters;  
  
import buildings.Village;  
import resources.Antilunite;  
import resources.Seeds;  
import resources.WeightlessnessDevices;  
  
import java.util.Objects;  
  
public class Villagers implements GetHelpFromAstronaut {  
 private final String nameOfVillagers = "Деревенские жители";  
 private final String name = "лунтакам";  
 private Seeds seeds;  
 String instruction;  
 WeightlessnessDevices weightlessnessDevices;  
 Antilunite antilunite;  
  
 public void goToAstronauts(String name) {  
 System.*out*.print(nameOfVillagers + " могли беспрепятственно приходить к " + name + " ");  
 }  
  
 public void setWeightlessnessDevices(WeightlessnessDevices weightlessnessDevices) {  
 this.weightlessnessDevices = weightlessnessDevices;  
 }  
  
 public void setAntilunite(Antilunite antilunite) {  
 this.antilunite = antilunite;  
 }  
  
 public void setSeeds(Seeds seeds) {  
 this.seeds = seeds;  
 }  
  
 @Override  
 public void getWeightlessnessDevicesFromAstronaut(WeightlessnessDevices weightlessnessDevices) {  
 setWeightlessnessDevices(weightlessnessDevices);  
 }  
  
 @Override  
 public void getAntiluniteFromAstronaut(Antilunite antilunite) {  
 setAntilunite(antilunite);  
 }  
  
 @Override  
 public void getSeedsFromAstronaut(Seeds seeds) {  
 setSeeds(seeds);  
 System.*out*.println("и получать у них " + seeds.getName() + " гигантских растений.");  
 }  
  
 public void setInstruction(String instruction) {  
 this.instruction = instruction;  
 }  
  
 public void getExplainFromAstronaut(String instruction) {  
 setInstruction(instruction);  
 }  
  
 public void tellAboutPlantingSites() {  
 System.*out*.println("Теперь гигантские семена сажали не только в деревне Нееловке, но и в селе "  
 + Village.*GOLOPYATKINA*.name() + ", "  
 + Village.*BESKHLEBNOV*.name() + ", "  
 + Village.*GOLODAYEVKA*.name() + ", "  
 + Village.*IMPASSABLE*.name() + "и во многих других.");  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Villagers villagers = (Villagers) o;  
 return Objects.*equals*(nameOfVillagers, villagers.nameOfVillagers) && Objects.*equals*(name, villagers.name) && Objects.*equals*(seeds, villagers.seeds) && Objects.*equals*(instruction, villagers.instruction) && Objects.*equals*(weightlessnessDevices, villagers.weightlessnessDevices) && Objects.*equals*(antilunite, villagers.antilunite);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(nameOfVillagers, name, seeds, instruction, weightlessnessDevices, antilunite);  
 }  
  
 @Override  
 public String toString() {  
 return "Villagers{" +  
 "nameOfVillagers='" + nameOfVillagers + '\'' +  
 ", name='" + name + '\'' +  
 ", seeds=" + seeds +  
 ", instruction='" + instruction + '\'' +  
 ", weightlessnessDevices=" + weightlessnessDevices +  
 ", antilunite=" + antilunite +  
 '}';  
 }  
}

package characters;  
  
import java.util.Objects;  
  
public class Znayka {  
 private String name = "Знайка";  
 private String order = "чтоб лунатикам давали не только нужные им семена, но снабжали их приборами невесомости," +  
 " а также антилунитом и объясняли им, как всем этим пользоваться, чтоб защититься от полицейских.";  
  
 public void giveOrder() {  
 System.*out*.println(name + " распорядился, " + order);  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Znayka znayka = (Znayka) o;  
 return Objects.*equals*(name, znayka.name) && Objects.*equals*(order, znayka.order);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(name, order);  
 }  
  
 @Override  
 public String toString() {  
 return "Znayka{" +  
 "name='" + name + '\'' +  
 ", order='" + order + '\'' +  
 '}';  
 }  
}

***classes of Package buildings***

package buildings;  
  
import java.util.Objects;  
  
public class Factory {  
 private String nameOfFactory="макаронная фабрика";  
 private boolean weightlessness=false;  
  
 public boolean isWeightlessness() {  
 return weightlessness;  
 }  
  
 public void setWeightlessness(boolean weightlessness) {  
 this.weightlessness = weightlessness;  
 }  
  
 public String getNameOfFactory() {  
 return nameOfFactory;  
 }  
  
 public void setNameOfFactory(String nameOfFactory) {  
 this.nameOfFactory = nameOfFactory;  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Factory factory = (Factory) o;  
 return weightlessness == factory.weightlessness && Objects.*equals*(nameOfFactory, factory.nameOfFactory);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(nameOfFactory, weightlessness);  
 }  
  
 @Override  
 public String toString() {  
 return "Factory{" +  
 "nameOfFactory='" + nameOfFactory + '\'' +  
 ", weightlessness=" + weightlessness +  
 '}';  
 }  
}

package buildings;  
  
public class Rocket {  
 public Rocket() {  
 }  
}

package buildings;  
  
public enum Village {  
 *GOLOPYATKINA*,  
 *BESKHLEBNOV*,  
 *GOLODAYEVKA*,  
 *IMPASSABLE*}

***classes of package resources***

package resources;  
  
import java.util.Objects;  
  
public class Antilunite extends Resource {  
 private String name="антилунит";  
 public Antilunite() {  
 super("антилунитом");  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Antilunite that = (Antilunite) o;  
 return Objects.*equals*(name, that.name);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(name);  
 }  
  
 @Override  
 public String toString() {  
 return "Antilunite{" +  
 "name='" + name + '\'' +  
 '}';  
 }  
}

package resources;  
  
public abstract class Resource {  
 private final String name;  
  
 public Resource(String name) {  
 this.name = name;  
 }  
  
 public String getName() {  
 return name;  
 }  
}

package resources;  
  
import java.util.Objects;  
  
public class Seeds extends Resource {  
 private String name="семена";  
 public Seeds() {  
 super("семена");  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Seeds seeds = (Seeds) o;  
 return Objects.*equals*(name, seeds.name);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(name);  
 }  
  
 @Override  
 public String toString() {  
 return "Seeds{" +  
 "name='" + name + '\'' +  
 '}';  
 }  
}

package resources;  
  
import java.util.Objects;  
  
public class WeightlessnessDevices extends Resource {  
 private String name = "прибор невесомости";  
  
 public WeightlessnessDevices() {  
 super("приборами невесомости");  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 WeightlessnessDevices that = (WeightlessnessDevices) o;  
 return Objects.*equals*(name, that.name);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(name);  
 }  
  
 @Override  
 public String toString() {  
 return "WeightlessnessDevices{" +  
 "name='" + name + '\'' +  
 '}';  
 }  
}

**Вывод:**