Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Факультет систем управления и робототехники

**Отчет по лабораторной работе №4**

**по дисциплине «Программирование»**

Выполнил: студент гр. **R3135**

Мамиконян Б.Н.

Преподаватель: Мартин Райла

преподаватель фак. Пиикт

Санкт-Петербург 2020

# **Текст задания**

# 

# На отдельных снимках можно было разглядеть пустые несгораемые сундуки, пустую несгораемую кассу с настежь раскрытыми дверцами, а также привязанную к подоконнику веревку, по которой Незнайка и Козлик спустились вниз. Никто, конечно, не знал, что Спрутс подкупил владельцев газет, чтоб они не печатали до поры до времени сообщений о бегстве Миги и Жулио. Но теперь, когда газеты сообщили об этом, Скуперфильду оставалось только выбросить свои акции. Их и даром никто не хотел брать. Истратив почти весь свой запас денег на акции, Скуперфильд, как принято говорить, сел на мель. Ему нужно было покупать для своей макаронной фабрики муку, нужно было платить рабочим, а поскольку денег на все не хватало, он решил снизить рабочим плату: вместо фертинга в день стал платить по полфертинга.

# **Текст для 4-ой лабораторной:**

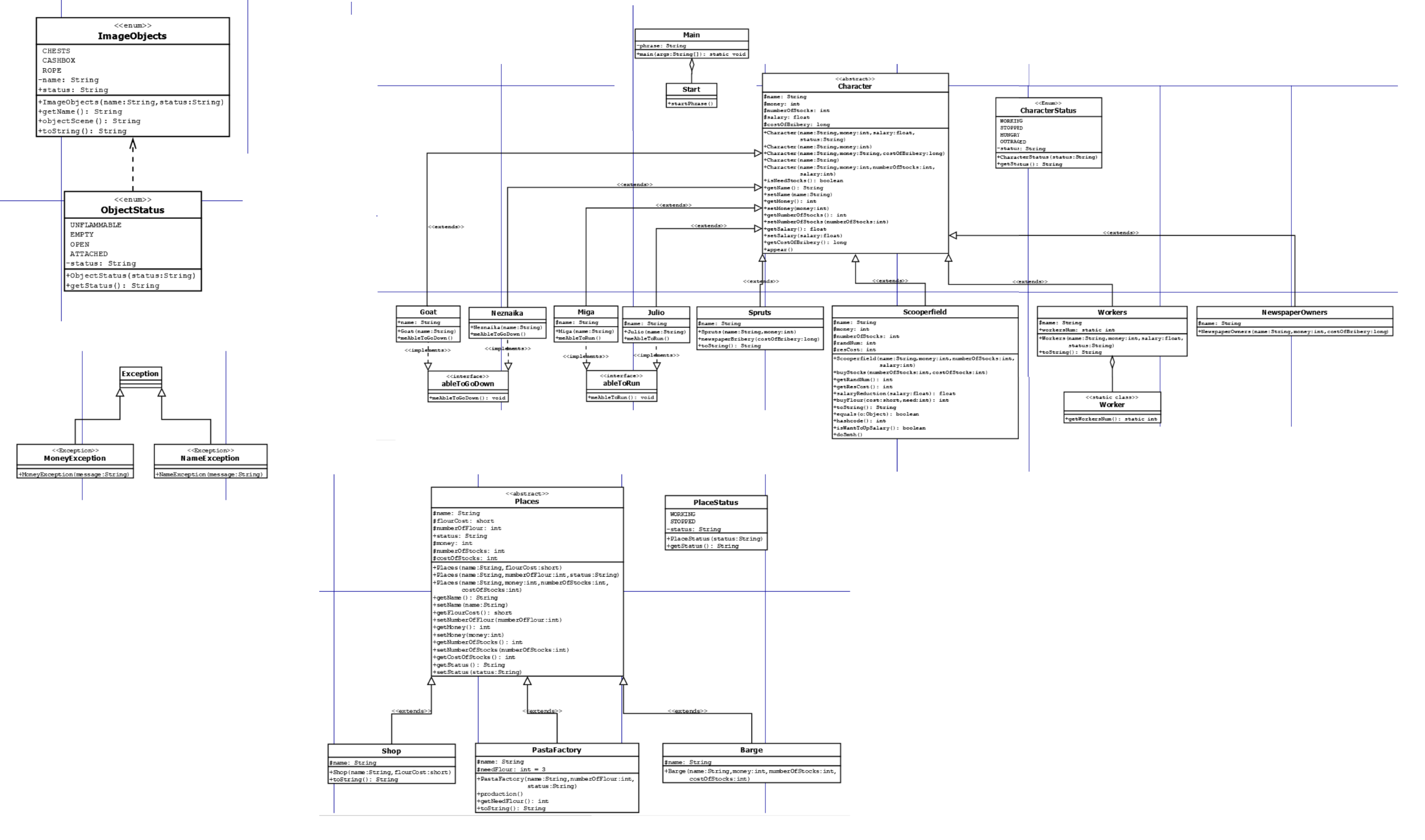
# Рабочие были возмущены, так как и на фертинг они могли существовать только впроголодь. Они сказали, что бросят работу, если Скуперфильд не прибавит плату. Скуперфильд вообразил, что рабочие решили его попугать, и не стал прибавлять плату. Тогда рабочие бросили работу. Фабрика остановилась, и теперь Скуперфильд уже не получал никаких доходов.

**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).

# 2. Исходный код программы

# Диаграмма классов



Main.java

package Story;  
  
  
public class Main {  
 private String phrase = "The beginning of the story";  
  
 public static void main(String[] args) {  
  
 Main main = new Main();  
 Start start = main.new Start();  
  
 Miga miga = new Miga("Miga");  
 Julio julio = new Julio("Julio");  
 Neznaika neznaika = new Neznaika("Neznaika");  
 Goat goat = new Goat("Goat");  
 Spruts spruts = new Spruts("Spruts", 2000);  
 NewspaperOwners newspaperOwners = new NewspaperOwners("NewspaperOwners", 500, 1500);  
 Workers workers = new Workers("Workers", 5, 1, CharacterStatus.*WORKING*.getStatus());  
 // Анонимный класс  
 Scooperfield scooperfield = new Scooperfield("Scooperfield", 1000, 0, 0) {  
 @Override  
 public void doSmth() {  
 System.*out*.println("Scooperfield ran aground. He has " + getMoney() + " fertings");  
 }  
 };  
  
 Barge barge = new Barge("Barge", 100, 100, 10);  
 PastaFactory pastaFactory = new PastaFactory("PastaFactory", 0, null);  
 Shop shop = new Shop("Shop", (short) 5);  
  
  
 boolean nameCheck = miga.getName() == null | julio.getName() == null | neznaika.getName() == null | goat.getName() == null | spruts.getName() == null | newspaperOwners.getName() == null | workers.getName() == null | scooperfield.getName() == null | barge.getName() == null | pastaFactory.getName() == null | shop.getName() == null;  
 boolean moneyCheck = spruts.getMoney() == 0 | newspaperOwners.getMoney() == 0 | workers.getMoney() == 0 | scooperfield.getMoney() == 0 | barge.getMoney() == 0;  
  
 if (nameCheck) {  
 try {  
 throw new NameException("The name value is 'null'");  
 } catch (NameException e) {  
 System.*err*.println("Incorrect name. Please change the name");  
 e.printStackTrace();  
 System.*exit*(0);  
 }  
 } else {  
 System.*out*.println("Correct names");  
 }  
  
 if (moneyCheck) {  
 try {  
 throw new MoneyException("The money value is '0'");  
 } catch (MoneyException ex) {  
 System.*err*.println("Someone has no money. Please change the value of money");  
 ex.printStackTrace();  
 System.*exit*(0);  
 }  
 } else {  
 System.*out*.println("Correct money values");  
 }  
  
 start.startPhrase();  
 float initialBalance = scooperfield.getMoney();  
 System.*out*.println(ImageObjects.*objectScene*());  
 neznaika.appear();  
 goat.appear();  
 neznaika.meAbleToGoDown(); // Незнайка и Козлик спускаются по верёвке  
 goat.meAbleToGoDown();  
 spruts.appear();  
 newspaperOwners.appear();  
 scooperfield.buyStocks(barge.getNumberOfStocks(), barge.getCostOfStocks()); // Покупка акций Скуперфильдом  
 barge.setMoney(barge.getMoney() + scooperfield.getResCost());  
 barge.setNumberOfStocks(barge.getNumberOfStocks() - scooperfield.getRandNum());  
 miga.appear();  
 julio.appear();  
 miga.meAbleToRun();  
 julio.meAbleToRun();  
 spruts.newspaperBribery(newspaperOwners.getCostOfBribery());  
 newspaperOwners.setMoney(newspaperOwners.getMoney() + (int) newspaperOwners.getCostOfBribery());  
 for (int day = 2 + (int) (Math.*random*() \* ((5 - 2) + 1)); day >= 0; day--) { // рандомное число дней  
 System.*out*.println("NewspaperOwners don't print about the escape of Migi and Julio for " + day + " days");  
 if (day == 0) {  
 System.*out*.println("They told people about the escape!");  
 }  
 }  
 if (!Character.*isNeedStocks*()) {  
 System.*out*.println("Nobody wants stocks");  
 scooperfield.setNumberOfStocks(0);  
 System.*out*.println("Scooperfield throws out stocks. He has " + scooperfield.getNumberOfStocks() + " stocks");  
 }  
 float percentage = scooperfield.getMoney() / initialBalance;  
 if (percentage < 0.2) {  
 scooperfield.doSmth();  
 System.*out*.println(); //  
 }  
 workers.setSalary(scooperfield.salaryReduction(workers.getSalary()));  
 System.*out*.println(workers.toString());  
 pastaFactory.production();  
 System.*out*.println("Number of factory employees: " + Workers.Worker.*getWorkersNum*());  
 System.*out*.println(shop.toString());  
 pastaFactory.setNumberOfFlour(scooperfield.buyFlour(shop.getFlourCost(), pastaFactory.getNeedFlour()));  
 System.*out*.println(pastaFactory.toString());  
//-----------------------------------------------------  
 System.*out*.println("Workers are " + workers.getStatus());  
 if (workers.getStatus().equals(CharacterStatus.*WORKING*.getStatus())) {  
 pastaFactory.setStatus(PlaceStatus.*WORKING*.getStatus());  
 }  
 System.*out*.println("PastaFactory is " + pastaFactory.getStatus());  
 scooperfield.setSalary(shop.getFlourCost() \* (float) 1.5);  
 System.*out*.println("Scooperfield earns " + scooperfield.getSalary() + " ferting");  
 if (workers.getSalary() == 0.5) {  
 workers.setStatus(CharacterStatus.*OUTRAGED*.getStatus() + " and " + CharacterStatus.*HUNGRY*.getStatus());  
 System.*out*.println("Workers are " + workers.getStatus());  
 }  
 if (!scooperfield.isWantToUpSalary()) {  
 System.*out*.println("Scooperfield doesn't want to rise salary");  
 workers.setStatus(CharacterStatus.*STOPPED*.getStatus());  
 System.*out*.println("Workers " + workers.getStatus());  
 }  
 if (workers.getStatus().equals(CharacterStatus.*STOPPED*.getStatus())) {  
 pastaFactory.setStatus(PlaceStatus.*STOPPED*.getStatus());  
 }  
 System.*out*.println("PastaFactory is " + pastaFactory.getStatus());  
 scooperfield.setSalary(0);  
 System.*out*.println("Scooperfield's salary is " + scooperfield.getSalary() + " ferting");  
 System.*err*.printf("%80s", "The end.\n");  
  
 }  
  
 private class Start {  
  
 void startPhrase() {  
 System.*out*.println("\n" + phrase + "\n");  
 }  
 }  
  
}

Character.java

package Story;  
  
public abstract class Character {  
 protected String name;  
 protected int money;  
 protected int numberOfStocks;  
 protected float salary;  
 protected String status;  
 protected long costOfBribery;  
  
 // Workers  
 public Character(String name, int money, float salary, String status) {  
 this.name = name;  
 this.money = money;  
 this.salary = salary;  
 this.status = status;  
 }  
  
 // Spruts  
 public Character(String name, int money) {  
 this.name = name;  
 this.money = money;  
 }  
  
 // NewspaperOwners  
 public Character(String name, int money, long costOfBribery) {  
 this.name = name;  
 this.money = money;  
 this.costOfBribery = costOfBribery;  
  
 }  
  
 // Miga Julio Neznaika Goat  
 public Character(String name) {  
 this.name = name;  
  
 }  
  
 // Scooperfield  
 public Character(String name, int money, int numberOfStocks, int salary) {  
 this.name = name;  
 this.money = money;  
 this.numberOfStocks = numberOfStocks;  
 this.salary = salary;  
  
 }  
  
 public static boolean isNeedStocks() {  
 return false;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public int getMoney() {  
 return money;  
 }  
  
 public void setMoney(int money) {  
 this.money = money;  
 }  
  
 public int getNumberOfStocks() {  
 return numberOfStocks;  
 }  
  
 public void setNumberOfStocks(int numberOfStocks) {  
 this.numberOfStocks = numberOfStocks;  
 }  
  
 public float getSalary() {  
 return salary;  
 }  
  
 public void setSalary(float salary) {  
 this.salary = salary;  
 }  
  
 public long getCostOfBribery() {  
 return costOfBribery;  
 }  
  
 public String getStatus() {  
 return status;  
 }  
  
 public void setStatus(String status) {  
 this.status = status;  
 }  
  
 public void appear() {  
 System.*out*.println(getName() + " appears on the stage");  
  
 }  
  
  
}

CharacterStatus.java

package Story;  
  
public enum CharacterStatus {  
 *WORKING*("working"),  
 *STOPPED*("stopped working"),  
 *HUNGRY*("hungry"),  
 *OUTRAGED*("outraged");  
 private final String status;  
  
 CharacterStatus(String status) {  
 this.status = status;  
 }  
  
 public String getStatus() {  
 return status;  
 }  
  
}

Places.java

package Story;  
  
public abstract class Places {  
 protected String name;  
 protected short flourCost;  
 protected int numberOfFlour;  
 protected String status;  
 protected int money;  
 protected int numberOfStocks;  
 protected int costOfStocks;  
// Shop  
 public Places(String name, short flourCost) {  
 this.name = name;  
 this.flourCost = flourCost;  
 }  
// PastaFactory  
 public Places(String name, int numberOfFlour, String status) {  
 this.name = name;  
 this.numberOfFlour = numberOfFlour;  
 this.status = status;  
 }  
// Barge  
 public Places(String name, int money, int numberOfStocks, int costOfStocks) {  
 this.name = name;  
 this.money = money;  
 this.numberOfStocks = numberOfStocks;  
 this.costOfStocks = costOfStocks;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public short getFlourCost() {  
 return flourCost;  
 }  
  
 public void setNumberOfFlour(int numberOfFlour) {  
 this.numberOfFlour = numberOfFlour;  
 }  
  
 public int getMoney() {  
 return money;  
 }  
  
 public void setMoney(int money) {  
 this.money = money;  
 }  
  
 public int getNumberOfStocks() {  
 return numberOfStocks;  
 }  
  
 public void setNumberOfStocks(int numberOfStocks) {  
 this.numberOfStocks = numberOfStocks;  
 }  
  
 public int getCostOfStocks() { //  
 return costOfStocks;  
 }  
  
 public String getStatus() {  
 return status;  
 }  
  
 public void setStatus(String status) {  
 this.status = status;  
 }  
}

PlaceStatus.java

package Story;  
  
public enum PlaceStatus {  
 *WORKING*("working"),  
 *STOPPED*("stopped");  
 private final String status;  
  
 PlaceStatus(String status) {  
 this.status = status;  
 }  
  
 public String getStatus() {  
 return status;  
 }  
  
}

Scooperfield.java

package Story;  
  
import java.util.Objects;  
  
public class Scooperfield extends Character {  
 protected String name;  
 protected int money = 0;  
 protected int numberOfStocks;  
 protected int randNum;  
 protected int resCost;  
  
  
 public Scooperfield(String name, int money, int numberOfStocks, int salary) {  
 super(name, money, numberOfStocks, salary);  
 }  
  
  
 public void buyStocks(int numberOfStocks, int costOfStocks) {  
  
 int min = 80;  
 randNum = min + (int) (Math.*random*() \* (((numberOfStocks - 5) - min) + 1));  
 resCost = randNum \* costOfStocks;  
 System.*out*.println("Scooperfield Bought " + randNum + " for the amount of " + resCost + " fertings");  
 setMoney(getMoney() - resCost);  
 setNumberOfStocks(getNumberOfStocks() + randNum);  
 System.*out*.println(toString());  
  
 }  
  
 public int getRandNum() {  
 return randNum;  
 }  
  
 public int getResCost() {  
 return resCost;  
 }  
  
 public float salaryReduction(float salary) {  
 System.*out*.println("Scooperfield reduced salary of workers to half ferting");  
 return salary / 2;  
 }  
  
 public int buyFlour(short cost, int need) {  
  
 System.*out*.println("Pasta Factory need " + need + " flour." + " Scooperfield has " + getMoney() + " fertings");  
 int resFlourCost = cost \* need;  
 System.*out*.println("Scooperfield buys flour");  
 setMoney(getMoney() - resFlourCost);  
 System.*out*.println(toString());  
  
 return need;  
 }  
  
 @Override  
 public String toString() {  
 return "Scooperfield{" +  
 "name='" + super.name + '\'' +  
 ", money=" + super.money +  
 ", numberOfStocks=" + super.numberOfStocks +  
 '}';  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Scooperfield that = (Scooperfield) o;  
 return money == that.money && numberOfStocks == that.numberOfStocks && randNum == that.randNum && resCost == that.resCost && Objects.*equals*(name, that.name);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(name, money, numberOfStocks, randNum, resCost);  
 }  
  
 public boolean isWantToUpSalary() {  
 return false;  
 }  
  
 public void doSmth() {  
 System.*out*.println("nothing");  
 }  
}

ableToGoDown.java

package Story;  
  
public interface ableToGoDown {  
 void meAbleToGoDown();  
}

ableToRun.java

package Story;  
  
public interface ableToRun {  
 void meAbleToRun();  
}

Barge.java

package Story;  
  
public class Barge extends Places {  
 protected String name;  
  
 public Barge(String name, int money, int numberOfStocks, int costOfStocks) {  
 super(name, money, numberOfStocks, costOfStocks);  
  
 }  
  
  
}

Goat.java

package Story;  
  
public class Goat extends Character implements ableToGoDown {  
 protected String name;  
  
 public Goat(String name) {  
 super(name);  
  
 }  
  
 @Override  
 public void meAbleToGoDown() {  
 System.*out*.println(super.name + " go down the " + ImageObjects.*ROPE*.name() + " !");  
 }  
}

ImageObjects.java

package Story;  
  
  
public enum ImageObjects {  
 *CHESTS*("Chests", ObjectStatus.*EMPTY*.getStatus() + " " + ObjectStatus.*UNFLAMMABLE*.getStatus()),  
 *CASHBOX*("Cashbox", ObjectStatus.*OPEN*.getStatus() + " " + ObjectStatus.*EMPTY*.getStatus() + " " + ObjectStatus.*UNFLAMMABLE*.getStatus()),  
 *ROPE*("Rope", ObjectStatus.*ATTACHED*.getStatus());  
 private final String name;  
 private final String status;  
  
 ImageObjects(String name, String status) {  
 this.name = name;  
 this.status = status;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public static String objectScene() {  
 return ImageObjects.*CHESTS* + "\n" + ImageObjects.*CASHBOX* + "\n" + ImageObjects.*ROPE*;  
 }  
  
 @Override  
 public String toString() {  
 return "ImageObjects{" +  
 "name='" + name + '\'' +  
 ", status='" + status + '\'' +  
 '}';  
 }  
  
}

ObjectStatus.java

package Story;  
  
public enum ObjectStatus {  
 *UNFLAMMABLE*("unflammable"),  
 *EMPTY*("empty"),  
 *OPEN*("open"),  
 *ATTACHED*("attached to the window sill");  
 private final String status;  
  
 ObjectStatus(String status) {  
 this.status = status;  
 }  
  
 public String getStatus() {  
 return status;  
 }  
}

Julio.java

package Story;  
  
public class Julio extends Character implements ableToRun {  
 protected String name;  
  
 public Julio(String name) {  
 super(name);  
  
 }  
  
 @Override  
 public void meAbleToRun() {  
 System.*out*.println(super.name + " escapes!");  
 }  
}

Miga.java

package Story;  
  
public class Miga extends Character implements ableToRun {  
 protected String name;  
  
 public Miga(String name) {  
 super(name);  
  
 }  
  
 @Override  
 public void meAbleToRun() {  
 System.*out*.println(super.name + " escapes!");  
 }  
}

NewspaperOwners.java

package Story;  
  
public class NewspaperOwners extends Character {  
 protected String name;  
  
 public NewspaperOwners(String name, int money, long costOfBribery) {  
 super(name, money, costOfBribery);  
  
 }  
}

Neznaika.java

package Story;  
  
public class Neznaika extends Character implements ableToGoDown {  
 protected String name;  
  
 public Neznaika(String name) {  
 super(name);  
 }  
  
 @Override  
 public void meAbleToGoDown() {  
 System.*out*.println(super.name + " go down the " + ImageObjects.*ROPE*.name() + " !");  
 }  
}

PastaFactory.java

package Story;  
  
public class PastaFactory extends Places {  
 protected String name;  
 protected int needFlour = 3;  
  
 public PastaFactory(String name, int numberOfFlour, String status) {  
 super(name, numberOfFlour, status);  
  
 }  
  
 void production() {  
 // Локальный класс  
 class Noodles {  
 final protected String productName = "noodles";  
 }  
  
 class Pasta {  
 final protected String productName = "pasta";  
 }  
 Noodles noodles = new Noodles();  
 Pasta pasta = new Pasta();  
 System.*out*.println("PastaFactory is producing " + noodles.productName + " and " + pasta.productName);  
  
 }  
  
 public int getNeedFlour() {  
 return needFlour;  
 }  
  
 @Override  
 public String toString() {  
 return "PastaFactory{" + "name='" + super.name + '\'' + ", numberOfFlour=" + super.numberOfFlour + "}";  
 }  
}

Shop.java

package Story;  
  
public class Shop extends Places {  
 protected String name;  
  
 public Shop(String name, short flourCost) {  
 super(name, flourCost);  
  
 }  
  
 @Override  
 public String toString() {  
 return "Shop{" +  
 "name='" + super.name + '\'' +  
 ", flourCost=" + super.flourCost +  
 '}';  
 }  
}

Spruts.java

package Story;  
  
  
public class Spruts extends Character {  
 protected String name;  
  
 public Spruts(String name, int money) {  
 super(name, money);  
  
 }  
  
 public void newspaperBribery(long costOfBribery) {  
  
 System.*out*.println("Cost of bribery: " + costOfBribery);  
 setMoney(getMoney() - (int) costOfBribery);  
 System.*out*.println(toString());  
  
 }  
  
 @Override  
 public String toString() {  
 return "Spruts has " + getMoney() + " money";  
 }  
  
  
}

Workers.java

package Story;  
  
  
public class Workers extends Character {  
 protected String name;  
 static int *workersNum* = 20;  
  
 public Workers(String name, int money, float salary, String status) {  
 super(name, money, salary, status);  
  
 }  
  
 @Override  
 public String toString() {  
 return "Workers{" + "name='" + super.name + '\'' + ", salary=" + super.salary + " fertings}";  
 }  
  
 // Статический класс  
 public static class Worker {  
 public static int getWorkersNum() {  
 return *workersNum*;  
 }  
  
 }  
  
}

NameException.java

package Story;  
// Unchecked exception  
public class NameException extends RuntimeException {  
 public NameException(String message){  
 System.*err*.println(message);  
 }  
}

MoneyException.java

package Story;  
// Checked exception  
public class MoneyException extends Exception {  
 public MoneyException(String message){  
 System.*err*.println(message);  
 }  
}

3.Результат работы программы

# Correct names

# Correct money values

# The beginning of the story

# ImageObjects{name='Chests', status='empty unflammable'}

# ImageObjects{name='Cashbox', status='open empty unflammable'}

# ImageObjects{name='Rope', status='attached to the window sill'}

# Neznaika appears on the stage

# Goat appears on the stage

# Neznaika go down the ROPE !

# Goat go down the ROPE !

# Spruts appears on the stage

# NewspaperOwners appears on the stage

# Scooperfield Bought 88 for the amount of 880 fertings

# Scooperfield{name='Scooperfield', money=120, numberOfStocks=88}

# Miga appears on the stage

# Julio appears on the stage

# Miga escapes!

# Julio escapes!

# Cost of bribery: 1500

# Spruts has 500 money

# NewspaperOwners don't print about the escape of Migi and Julio for 2 days

# NewspaperOwners don't print about the escape of Migi and Julio for 1 days

# NewspaperOwners don't print about the escape of Migi and Julio for 0 days

# They told people about the escape!

# Nobody wants stocks

# Scooperfield throws out stocks. He has 0 stocks

# Scooperfield ran aground. He has 120 fertings

# Scooperfield reduced salary of workers to half ferting

# Workers{name='Workers', salary=0.5 fertings}

# PastaFactory is producing noodles and pasta

# Number of factory employees: 20

# Shop{name='Shop', flourCost=5}

# Pasta Factory need 3 flour. Scooperfield has 120 fertings

# Scooperfield buys flour

# Scooperfield{name='Scooperfield', money=105, numberOfStocks=0}

# PastaFactory{name='PastaFactory', numberOfFlour=3}

# Workers are working

# PastaFactory is working

# Scooperfield earns 7.5 ferting

# Workers are outraged and hungry

# Scooperfield doesn't want to rise salary

# Workers stopped working

# PastaFactory is stopped

# Scooperfield's salary is 0.0 ferting

# The end.

# Process finished with exit code 0

# 4.Выводы по работе

# В этой лабораторной работе я научился создавать свои собственные классы исключений и обрабатывать их

# Научился использовать вложенные классы