НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Программирование

Лабораторная работа № 4

Выполнил студент

Хвостова Ирина Леонидовна

Группа № Р3124

Преподаватель: Харитонова Анастасия Евгеньевна

г. Санкт-Петербург

2022

**Вариант 4261,9**

**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.

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

Конечно, выглядело это очень заманчиво, но Алиса была умная девочка и не спешила откликнуться на любезное приглашение. - Нет,- сказала она,- я сначала посмотрю, написано тут "Яд!" или нет. Она недаром перечитала множество поучительных рассказов про детей, с которыми случались разные неприятности - бедные крошки и погибали в пламени, и доставались на съедение диким зверям,- и все только потому, что они забывали (или не хотели помнить!) советы старших. А ведь, кажется, так просто запомнить, что, например, раскаленной докрасна кочергой можно обжечься, если будешь держать ее в руках слишком долго; что если ОЧЕНЬ глубоко порезать палец ножом, из этого пальца, как правило, пойдет кровь, и так далее и тому подобное. И уж Алиса-то отлично помнила, что если выпьешь слишком много из бутылки, на которой нарисованы череп и кости и написано "Яд!", то почти наверняка тебе не поздоровится (то есть состояние твоего здоровья может ухудшиться). Однако на этой бутылочке не было ни черепа, ни костей, ни надписи "Яд!", и Алиса рискнула попробовать ее содержимое. А так как оно оказалось необыкновенно вкусным (на вкус - точь-в-точь смесь вишневого пирога, омлета, ананаса, жареной индюшки, тянучки и горячих гренков с маслом), она сама не заметила, как пузырек опустел. Ой, что же это со мной делается! - сказала Алиса.- Я, наверное, и правда складываюсь, как подзорная труба! Спорить с этим было трудно: к этому времени в ней осталось всего лишь четверть метра. Алиса так и сияла от радости, уверенная, что она теперь свободно может выйти в чудесный сад. Но все-таки она решила на всякий случай немного подождать и убедиться, что она уже перестала уменьшаться в росте. "А то вдруг я буду делаться все меньше, меньше, как свечка, а потом совсем исчезну! - не без тревоги подумала она.- Вот бы поглядеть, на что я буду тогда похожа". И она попыталась вообразить, на что похоже пламя свечи, когда свеча погасла, но это ей не удалось,- ведь, к счастью, ей этого никогда не приходилось видеть... Подождав немного и убедившись, что все остается по-прежнему, Алиса побежала было в сад; но - такая незадача! - у самого выхода она вспомнила, что оставила золотой ключик на столе, а подбежав опять к столику, обнаружила, что теперь ей никак до ключа не дотянуться.

Исходный код:

public class Main {  
 public static void main(String[] args) {  
 System.*out*.println("\"Алиса в стране чудес\"");  
  
 Alice alice = new Alice("Алиса", 100, 100, 200);  
 Liquid flea = new Liquid(100, 100, 300);  
  
 alice.first();  
 alice.second();  
 alice.third();  
  
 alice.tease(flea);  
 alice.fourth();  
  
  
 }  
}

Блок интерфейсов:

* Insects

public interface Insects {  
 int *skillSwear* = 110;  
 */\*\*  
 \* Метод, который определяет выбор зелья  
 \** ***@Personage*** *p - персонаж  
 \** ***@return*** *true - в случае успеха, false - иначе  
 \*/* public boolean potionSelection(Personage p);  
 */\*\*  
 \* Метод, в котором зелье оказывается вкусным  
 \*/* public void potionDelicious(Personage p);  
  
 */\*\*  
 \* Метод, в котором зелье заканчивается  
 \*/* void potionOver(Personage p);  
}

Блок абстрактного класса:

* Personage

public abstract class Personage {  
 String name = "NoName";  
 String type = "Персонаж";  
 double x = 0;  
 double y = 0;  
 int skillSwear = 10;  
 int force;  
 int height;  
 Drawings mood = Drawings.*SKULL*;  
  
 @Override  
 public boolean equals(Object s) {  
 if (s == null) {  
 return false;  
 }  
 if (this == s) {  
 return true;  
 }  
 if (!(getClass() == s.getClass())){  
 return false;  
 }  
 else {  
 Personage tmp = (Personage) s;  
 return (tmp.name.equals(name) && force == tmp.force && mood == tmp.mood && tmp.type.equals(type));  
 }  
 }  
  
 @Override  
 public int hashCode(){  
 final int power = 239;  
 int hash = 0;  
 for (int i = 0; i < name.length(); ++i) {  
 hash = hash \* power + (int)name.charAt(i);  
 }  
  
 for (int i = 0; i < type.length(); ++i) {  
 hash = hash \* power + (int)type.charAt(i);  
 }  
  
  
 if (mood == Drawings.*SKULL*) {  
 hash \*= 2;  
 } else if (mood == Drawings.*BONES*) {  
 hash \*= 3;  
 } else if (mood == Drawings.*POISON*) {  
 hash \*= 4;  
 }  
  
 hash \*= force \* x \* y \* height;  
 return hash;  
 }  
  
  
 */\*\*  
 \* Кто-то что-то попробовал  
 \*/* public void feel(){  
 System.*out*.print(name + " попробовала ");  
 }  
}

Блок enum:

* Drawings

enum Drawings {  
 *SKULL*,  
 *BONES*,  
 *POISON*}

Блок других классов:

* Liquid

public class Liquid implements Insects{  
 String name = "она";  
 String type = " ";  
 int force = 1;  
 double x = 0;  
 double y = 0;  
 double z = 0;  
 int skillSwear = 0;  
  
 Liquid(double x, double y, double z) {  
 String name = "Зелье";  
 String type = "Жидкость";  
 int force = 9;  
 this.x = x;  
 this.y = y;  
 this.z = z;  
 }  
  
 @Override  
 public boolean potionSelection(Personage p) {  
 if (this.x == p.x && this.y == p.y && this.z >= p.height) {  
 System.*out*.println(this.name + " без надписей костей, черепов, яда.");  
 this.z = 6 \* p.height / 7;  
 return true;  
 }  
 else {  
 System.*out*.println(this.name + " с надписями костей, черепов, яда.");  
 this.z = 0;  
 return false;  
 }  
 }  
  
 @Override  
 public void potionDelicious(Personage p) {  
 System.*out*.println(this.name + " оказалось необыкновенно вкусным.");  
 }  
 @Override  
 public void potionOver(Personage p) {  
 p.force--;  
 System.*out*.println(this.name + " закончилось.");  
 }  
  
 @Override  
 public boolean equals(Object s) {  
 if (s == null) {  
 return false;  
 }  
 if (this == s) {  
 return true;  
 }  
 if (!(getClass() == s.getClass())){  
 return false;  
 }  
 else {  
 Alice tmp = (Alice) s;  
 return (tmp.name.equals(this.name) && this.force == tmp.force && this.type.equals(tmp.type));  
 }  
 }  
  
 @Override  
 public int hashCode(){  
 final int power = 239;  
 int hash = 0;  
 for (int i = 0; i < name.length(); ++i){  
 hash = hash \* power + (int)name.charAt(i);  
 }  
  
 for (int i = 0; i < type.length(); ++i){  
 hash = hash \* power + (int)type.charAt(i);  
 }  
  
 hash \*= force \* x \* y \* z;  
 return hash;  
 }  
  
 @Override  
 public String toString(){  
 return this.type + " " + this.name;  
 }  
}

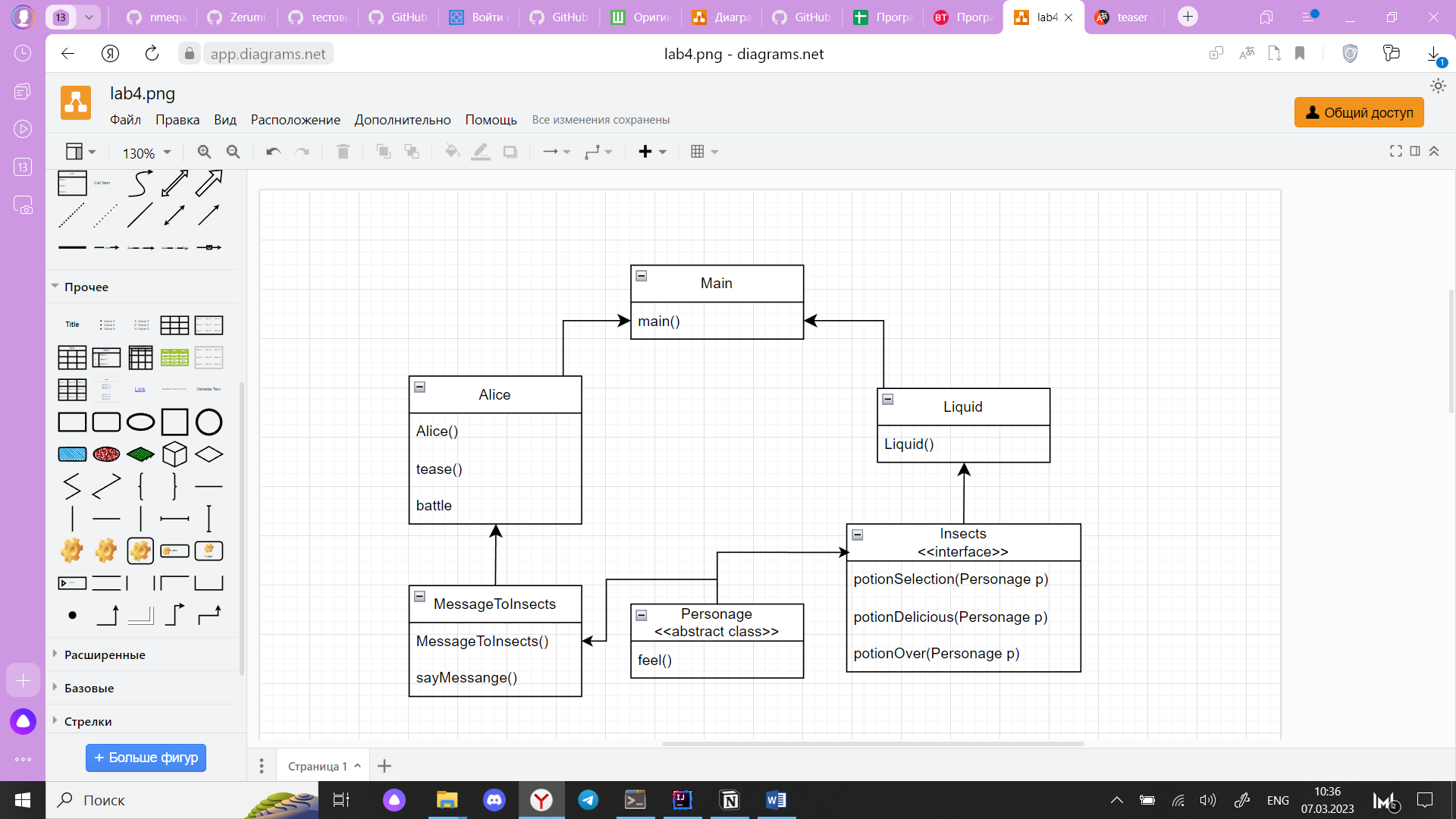
* Alice

public class Alice extends Personage {  
 String type = "Алиса";  
  
 */\*\*  
 \* Проверка исключений в компиляторе  
 \*/* class unchException extends RuntimeException{  
 unchException(String msg){  
 super(msg);  
 }  
 }  
  
 Alice(String name, double x, double y, int height) {  
 this.name = name;  
 this.type = "Алиса";  
 this.x = x;  
 this.y = y;  
 this.force = 7;  
 this.height = height;  
 }  
  
 public void first() {  
 System.*out*.println(this.name + " была умная девочка и не спешила откликнуться на любезное приглашение.");  
 }  
  
 public void second() {  
 System.*out*.println(this.name + " недаром перечитала множество поучительных рассказов про детей, с которыми случались разные неприятности. А ведь, кажется, так просто запомнить, что, например, раскаленной докрасна кочергой можно обжечься, если будешь держать ее в руках слишком долго; что если ОЧЕНЬ глубоко порезать палец ножом, из этого пальца, как правило, пойдет кровь, и так далее и тому подобное. ");  
 }  
 public void third() {  
 System.*out*.println(this.name + "-то отлично помнила, что если выпьешь слишком много из бутылки, на которой нарисованы череп и кости и написано \"Яд!\", то почти наверняка тебе не поздоровится (то есть состояние твоего здоровья может ухудшиться). ");  
 }  
 public void fourth() {  
 System.*out*.println(this.name + "Подождав немного и убедившись, что все остается по-прежнему, Алиса побежала было в сад; но - такая незадача! - у самого выхода она вспомнила, что оставила золотой ключик на столе, а подбежав опять к столику, обнаружила, что теперь ей никак до ключа не дотянуться.");  
 }  
 public boolean tease (Liquid p){  
 if (p == null){  
 throw new unchException("error");  
 }  
 try {  
 if (p.skillSwear < this.skillSwear) {  
 battle(p);  
 }  
 MessangeToInsects msg = new MessangeToInsects("Ой, что же это со мной делается!", "сказала", p, this);  
 msg.sayMessange();  
 if (p.force >= this.force){  
 this.feel();  
 p.potionSelection(this);  
 p.potionDelicious(this);  
  
 MessangeToInsects msg3 = new MessangeToInsects("Я начинаю взлеать!", "выкрикнула", p, this);  
  
 System.*out*.println("Она взлетала все выше и выше. И уже начала подумывать о том, что вот-вот может улететь в космос.");  
 msg3.sayMessange();  
 }  
 else {  
 MessangeToInsects msg3 = new MessangeToInsects("Я, наверное, и правда складываюсь, как подзорная труба!", "воскликнула", p, this);  
 System.*out*.println("Спорить с этим было трудно: к этому времени в ней осталось всего лишь четверть метра. Алиса так и сияла от радости, уверенная, что она теперь свободно может выйти в чудесный сад. Но все-таки она решила на всякий случай немного подождать и убедиться, что она уже перестала уменьшаться в росте.");  
 msg3.sayMessange();  
 }  
  
  
  
 } catch (unchException e){  
 System.*out*.println(e.getMessage());  
 } catch (Exception e) {  
 System.*out*.println(this.name + " немой.");  
 }  
 return false;  
 }  
  
 */\*\*  
 \* Алиса и Зелья  
 \** ***@param*** *p - Зелье  
 \** ***@return*** *true - Алиса выпила, false - иначе  
 \*/* public boolean battle(Liquid p){  
 if (p.force >= this.force){  
 this.feel();  
 p.potionSelection(this);  
 p.potionDelicious(this);  
  
 System.*out*.println(this.name + " Однако на этой бутылочке был череп, кости и надпись \"Яд!\", и Алиса рискнула попробовать ее содержимое.");  
 if (this.mood != Drawings.*BONES*){  
 changeMood(Drawings.*BONES*);  
 }  
 return true;  
 }  
 else {  
 System.*out*.println(" Однако на этой бутылочке не было ни черепа, ни костей, ни надписи \"Яд!\", и Алиса рискнула попробовать ее содержимое.");  
 if (this.mood != Drawings.*POISON*){  
 changeMood(Drawings.*POISON*);  
 }  
 return false;  
 }  
 }  
 public void changeMood(Drawings newMood){  
 switch (newMood){  
 case *SKULL*: {  
 System.*out*.println(this.name + "громко икнула.");  
 break;  
 }  
 case *BONES*: {  
 System.*out*.println(this.name + "начала себя странно чувствовать.");  
 break;  
 }  
 case *POISON*: {  
 System.*out*.println("А так как оно оказалось необыкновенно вкусным (на вкус - точь-в-точь смесь вишневого пирога, омлета, ананаса, жареной индюшки, тянучки и горячих гренков с маслом.");  
 break;  
 }  
 default: {  
 System.*out*.println(this.name + " выпила зелье без эффекта");  
 }  
 }  
 }  
  
  
}

* MessangeToInsects

import java.io.FileNotFoundException;  
import java.io.FileReader;  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Scanner;  
  
public class MessangeToInsects {  
 public String tone;  
 public String msg;  
 public Insects toWhom;  
 public Personage fromWhom;  
 class PotionException extends Exception{  
 PotionException(String msg){  
 super(msg);  
 }  
 }  
  
 */\*\*  
 \* Метод, который создает сообщение  
 \*/* MessangeToInsects(String msg, String tone, Insects toWhom, Personage fromWhom){  
 this.msg = msg;  
 this.toWhom = toWhom;  
 this.fromWhom = fromWhom;  
 this.tone = tone;  
 }  
 */\*\*  
 \* Метод в котором Алиса говорит об эффекте зелья  
 \* Причем, если skillSwear у зелья больше, чем у персонажа, то монолог идет дальше.  
 \*/* void sayMessange() throws IOException {  
 System.*out*.println("\"" + msg + "\"" + " - " + tone + " " + fromWhom.name + ".");  
  
 if (toWhom.*skillSwear* >= fromWhom.skillSwear){  
  
 class Answer {  
 private ArrayList<String> tone = new ArrayList<>();  
 private ArrayList<String> ans = new ArrayList<>();  
  
 Answer() throws IOException {  
 FileReader answer = new FileReader("src\\materials\\answer.txt");  
 FileReader toneFile = new FileReader("src\\materials\\tone.txt");  
 Scanner scanAns = new Scanner(answer);  
 Scanner scanTone = new Scanner(toneFile);  
 while(scanAns.hasNextLine()) {  
 ans.add(scanAns.nextLine());  
 }  
 while (scanTone.hasNextLine()) {  
 tone.add(scanTone.nextLine());  
 }  
  
 answer.close();  
 toneFile.close();  
 }  
  
 private void sayAnswer() throws PotionException {  
 if (ans.size() == 0 || tone.size() == 0){  
 throw new PotionException(toWhom + " не понимала, что делать дальше.");  
 }  
 System.*out*.println("\"" + ans.get((int)(Math.*random*()\*100 % ans.size())) + "\"" + " - " + tone.get((int)(Math.*random*()\*100 % tone.size())) + " " + toWhom + ".");  
 }  
 }  
  
 Answer answer = new Answer();  
  
 try {  
 answer.sayAnswer();  
 } catch (ArrayIndexOutOfBoundsException e) {  
 System.*out*.println(toWhom + " заплакала и промолчала, потому что обратилась к недопустимому индексу массива.");  
 } catch (PotionException e) {  
 System.*out*.println(e.getMessage());  
 } catch (NullPointerException e) {  
 System.*out*.println("Никто ничего не ответил, потому что, на самом деле, никакого зелья не существует. Как и Алисы.");  
 } catch (Exception e){  
 System.*out*.println(toWhom + " просто испарилась. Никто не знает почему.");  
 }  
 }  
 }  
}

Диаграмма классов реализованной объектной модели. Иерархия объектов.



**Результат работы программы:**

"Алиса в стране чудес"

Алиса была умная девочка и не спешила откликнуться на любезное приглашение.

Алиса недаром перечитала множество поучительных рассказов про детей, с которыми случались разные неприятности. А ведь, кажется, так просто запомнить, что, например, раскаленной докрасна кочергой можно обжечься, если будешь держать ее в руках слишком долго; что если ОЧЕНЬ глубоко порезать палец ножом, из этого пальца, как правило, пойдет кровь, и так далее и тому подобное.

Алиса-то отлично помнила, что если выпьешь слишком много из бутылки, на которой нарисованы череп и кости и написано "Яд!", то почти наверняка тебе не поздоровится (то есть состояние твоего здоровья может ухудшиться).

Однако на этой бутылочке не было ни черепа, ни костей, ни надписи "Яд!", и Алиса рискнула попробовать ее содержимое.

А так как оно оказалось необыкновенно вкусным (на вкус - точь-в-точь смесь вишневого пирога, омлета, ананаса, жареной индюшки, тянучки и горячих гренков с маслом.

"Ой, что же это со мной делается!" - сказала Алиса.

"Почему я себя так странно ощущаю?" - вскрикнула она.

Спорить с этим было трудно: к этому времени в ней осталось всего лишь четверть метра. Алиса так и сияла от радости, уверенная, что она теперь свободно может выйти в чудесный сад. Но все-таки она решила на всякий случай немного подождать и убедиться, что она уже перестала уменьшаться в росте.

"Я, наверное, и правда складываюсь, как подзорная труба!" - воскликнула Алиса.

"А то вдруг я буду делаться все меньше, меньше, как свечка, а потом совсем исчезну!" - не без тревоги подумала она.

Алиса подождав немного и убедившись, что все остается по-прежнему, Алиса побежала было в сад; но - такая незадача! - у самого выхода она вспомнила, что оставила золотой ключик на столе, а подбежав опять к столику, обнаружила, что теперь ей никак до ключа не дотянуться.

**Вывод:**

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