**Федеральное государственное автономное образовательное учреждение высшего образования  
«Национальный исследовательский университет ИТМО»  
  
Факультет программной инженерии и компьютерной техники  
  
Направление подготовки 09.03.04 «Программная инженерия» – Системное и прикладное программное обеспечение**

**Отчёт  
По лабораторной работе №4  
По дисциплине «Программирование»  
Вар: 13132626**

**Выполнил:  
Студент 1 курса  
Васильев Артём Евгеньевич  
Группа: 3119  
Преподаватель:  
Бойко Владислав Алексеевич**

Санкт-Петербург, 2023 г.

# Оглавление

[Задание 3](#_Toc6581)

[Диаграмма классов 4](#_Toc11571)

[Исходный код программы 5](#_Toc1958)

[Вывод программы 15](#_Toc26358)

[Вывод 16](#_Toc13496)

# Задание

Доработать программу из [лабораторной работы #3](https://se.ifmo.ru/disciplines/programming" \l "lab3), обновив реализацию объектной модели в соответствии с новой версией описания предметной области.

**Программа должна удовлетворять следующим требованиям:**

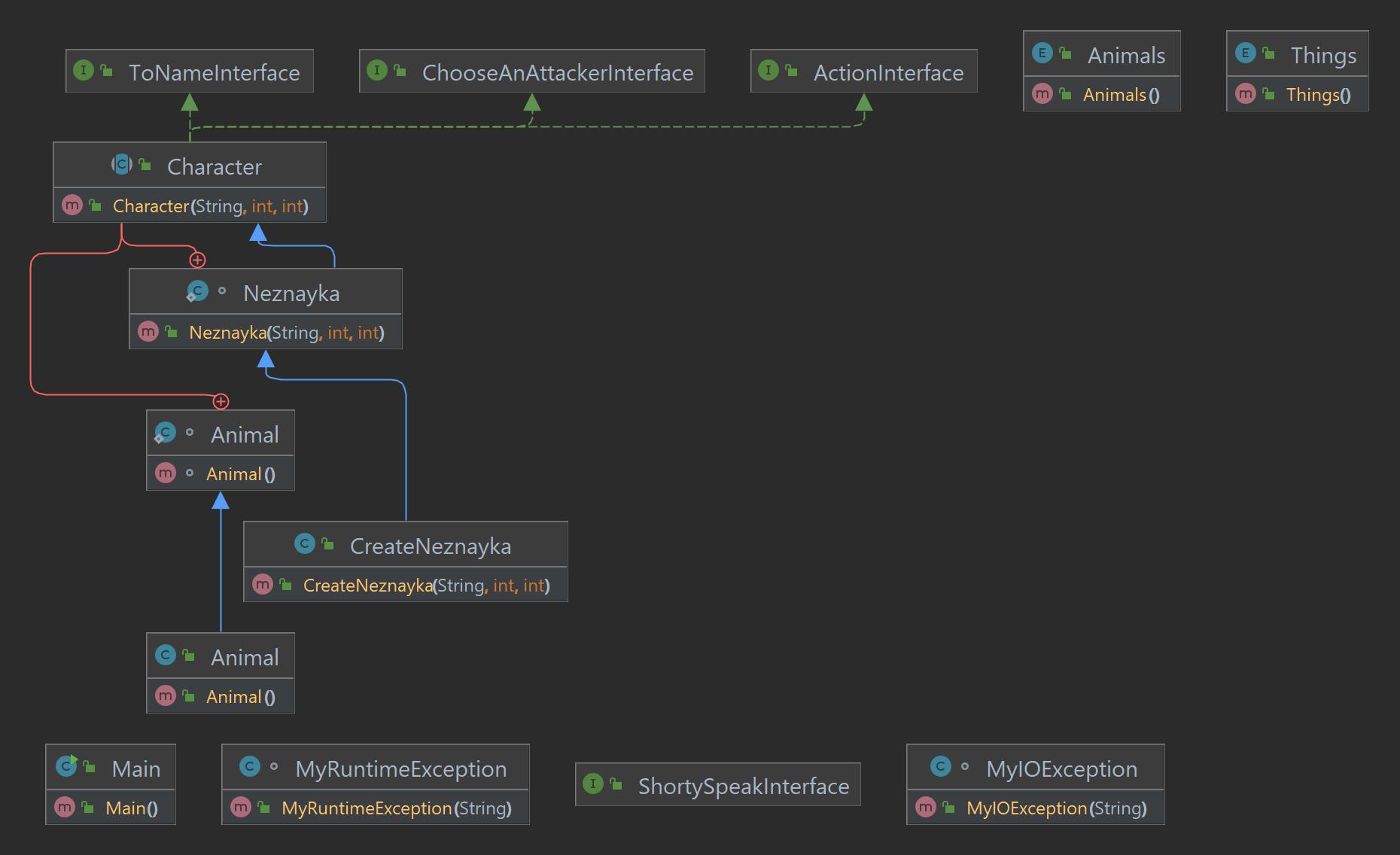
В программе должны быть реализованы 2 собственных класса исключений (checked и unchecked), а также обработка исключений этих классов.

В программу необходимо добавить использование локальных, анонимных и вложенных классов (static и non-static).

**Описание предметной области:**

Коротышки притихли. Отвратительный запах уже не так тревожил обоняние Незнайки, но было нестерпимо душно. Крыса, не особенно торопясь, отбежала в сторонку и, остановившись неподалеку, стала поглядывать на Незнайку своими блестящими, словно бусинки, глазками. С тревогой осмотревшись по сторонам. Животные отбежали в сторонку и стали поглядывать на незнайку своими глазками. С тревогой осмотревшись по сторонам.

# Диаграмма классов



# Исходный код программы ([GitHub](https://github.com/frizyyu/lab4_prog/tree/main))

Main.java

import Characters.\*;  
import enums.Animals;  
import enums.Things;  
import interfaces.ShortySpeakInterface;  
  
import java.io.IOException;  
  
public class Main  
{  
 public static void main(String[] args) throws MyIOException  
 {  
 CreateNeznayka m = new CreateNeznayka("Незнайка", 15, 150);  
 System.out.println(m.toString());  
 m.SayHello();  
  
 ShortySpeakInterface ShortySpeak = new ShortySpeakInterface() {  
 @Override  
 public void Say(String name, int num) throws MyRuntimeException{  
 System.out.println(String.format("Я коротышка %s. Меня зовут %s", num, name));  
 switch (num){  
 case 1:  
 System.out.println("Мне стало скучно и я притихла.");  
 break;  
 case 2:  
 System.out.println("Мне тоже стало скучно и я замолчала.");  
 break;  
 }  
 }  
 };  
 ShortySpeak.Say("Алина", 1);  
 ShortySpeak.Say("Марина", 2);  
  
 m.TooSleep();  
 m.Attack();  
 m.Hide();  
 m.Fight();  
 m.NameWhatItIs(Animals.BUGS, Things.GARBAGE);  
 m.GoAfk();  
 m.NameANewAnimal(Animals.DINOSAURS);  
 Animal r = new Animal();  
 r.TellYourRole();  
 }  
}  
  
class MyIOException extends IOException {  
 public MyIOException(String e){  
 super(e);  
 }  
}  
  
class MyRuntimeException extends RuntimeException{  
 public MyRuntimeException(String e){  
 super(e);  
 }  
}

Character.java

package Characters;  
  
import enums.Animals;  
import enums.Things;  
import interfaces.ToNameInterface;  
import interfaces.\*;  
  
public abstract class Character implements ToNameInterface, ActionInterface, ChooseAnAttackerInterface {  
 String name;  
 int age;  
 int force;  
 public Character(String name, int age, int force){  
 this.name = name;  
 this.age = age;  
 this.force = force;  
 }  
 static class Neznayka extends Character{  
 public Neznayka(String name, int age, int force) {  
 super(name, age, force);  
 }  
  
 @Override  
 public void SayHello() {  
 class SayHello{  
 public SayHello() {  
 System.out.println("Всем привет!!");  
 }  
 }  
  
 }  
  
 @Override  
 public void TooSleep() {  
  
 }  
  
 @Override  
 public void Hide() {  
  
 }  
  
 @Override  
 public void GoAfk() {  
  
 }  
  
 @Override  
 public void Attack() {  
  
 }  
  
 @Override  
 public void Fight() {  
  
 }  
  
 @Override  
 public String ChooseInanimate(Things t) {  
 return null;  
 }  
  
 @Override  
 public String ChooseAlive(Animals a) {  
 return null;  
 }  
  
 @Override  
 public void NameWhatItIs(Animals a, Things t) {  
  
 }  
  
 @Override  
 public void NameANewAnimal(Animals a) {  
  
 }  
 }  
 public String Return\_Spec(){  
 return String.format("Имя: %s\nВозраст: %s\nСила: %s\n", name, age, force);  
 }  
  
 public abstract void SayHello();  
  
 static class Animal{  
 Animal(){  
 }  
 public void TellYourRole(){  
 }  
 }  
}

Animal.java

package Characters;  
  
public class Animal extends Character.Animal{  
 @Override  
 public void TellYourRole(){  
 System.out.println("Животные отбежали в сторонку и стали поглядывать на незнайку своими глазками. С тревогой осмотревшись по сторонам.");  
 }  
}

CreateNeznayka.java

package Characters;  
  
import enums.Animals;  
import enums.Things;  
  
import java.util.Objects;  
import static java.lang.Math.\*;  
  
public class CreateNeznayka extends Character.Neznayka {  
 String strenght;  
 boolean is\_dropped\_down = false;  
  
 public CreateNeznayka(String name, int age, int force) {  
 super(name, age, force);  
 }  
  
 String[] SelectWhatSay(String... args){  
 int random = (int) (random() \* args.length - 1);  
 String[] what\_say = new String[args.length];  
 for (int i = 0; i < args.length; i += 1){  
 what\_say[i] = args[i];  
 }  
 String res = Integer.toString(random);  
 String[] ress = new String[2];  
 ress[0] = what\_say[random];  
 ress[1] = res;  
 return ress;  
 }  
  
 @Override  
 public int hashCode() {  
 return super.hashCode() + name.hashCode();  
 }  
  
 @Override  
 public boolean equals(Object obj) {  
 return obj.hashCode() == this.hashCode();  
 }  
  
 @Override  
 public String toString() {  
 return String.format("Статистика персонажа:\n%s", Return\_Spec());  
 }  
  
 @Override  
 public void SayHello(){  
 class SayHello{  
 SayHello(){  
 if (force > 100)  
 strenght = "очень сильный!";  
 else  
 strenght = "слабоват..";  
 String s = String.format("Привет, меня зовут %s! Я - главный герой этой истории.\nМне %s лет и я %s\n", name, age, strenght);  
 System.out.println(s);  
 }  
 }  
 SayHello sh = new SayHello();  
 }  
  
  
 @Override  
 public void TooSleep() {  
 System.out.println(SelectWhatSay(  
 String.format("%s лёг на полку, чтобы хорошенько отдохнуть.", name),  
 String.format("Захотев поспать, %s решил сесть на стул и опереться на стену.", name),  
 String.format("%s решил отдохнуть, поэтому он начал укладываться на диван.", name),  
 String.format("%s уже начинал засыпать, поэтому он прилёг на диван.", name))[0]);  
 System.out.println("Герой лёг поудобнее, закрыл глаза и погрузился в сон");  
 System.out.println("Вдруг произошло что-то неожиданное.");  
 }  
 @Override  
 public void Attack(){  
 String[] r = SelectWhatSay(  
 "напали какие-то маленькие зверушки.",  
 "напали будто какие-то насекомые и начали кусать.",  
 "упали какие-то тяжёлые вещи.",  
 "что-то навалилось и придавило сверху.");  
 is\_dropped\_down = !Objects.equals(r[1], "0") && !Objects.equals(r[1], "1");  
 String s = String.format("Скоро он почувствовал, что на него %s", r[0]);  
 System.out.println(s);  
 }  
  
 @Override  
 public void Hide() {  
 if (!is\_dropped\_down){  
 System.out.println(SelectWhatSay(  
 String.format("%s отпрыгнул в сторону и попятился назад.", name),  
 String.format("%s счёл это неожиданным.", name),  
 String.format("%s был растерян, поэтому просто сел на пол.", name),  
 String.format("%s решил, что лучшем решением будет отойти в сторону.", name))[0]);  
 }  
 else  
 System.out.println(SelectWhatSay(  
 String.format("%s с трудом поднял вещи с себя и принялся их рассматривать.", name),  
 String.format("%s потёр затылок и неспешно поднялся.", name),  
 String.format("Вещи были очень тяжёлыми, %s с трудом вылез из под них.", name),  
 String.format("%s отполз в сторону, скинув с себя груз.", name))[0]);  
 }  
  
 @Override  
 public void Fight() {  
 int opp\_force = (int) (random() \* 100);  
 if (force < opp\_force && !is\_dropped\_down) {  
 System.out.println(SelectWhatSay(  
 "Герой долго дрался с живностью, но проиграл.",  
 "Герой был сильно закусан.")[0]);  
 System.exit(0);  
 }  
 else if (force >= opp\_force && !is\_dropped\_down)  
 System.out.println(SelectWhatSay(  
 String.format("%s долго дрался с живностью и выиграл.", name),  
 String.format("%s победил зверушек.", name))[0]);  
 }  
  
 @Override  
 public void NameWhatItIs(Animals a, Things t){  
 if (!is\_dropped\_down)  
 //System.out.println(Select\_an(a));  
 System.out.printf("%s подошёл к животным. Как оказалось, это были %s.%n", name, ChooseAlive(a));  
 else  
 System.out.printf("%s подошёл к разбросанным вещам. Как оказалось, это %s.%n", name, ChooseInanimate(t));  
 }  
  
 @Override  
 public String ChooseAlive(Animals a) {  
 return switch (a) {  
 case BUGS -> "жуки";  
 case COCKROACHES -> "тараканы";  
 case DINOSAURS -> "динозавры";  
 };  
 }  
  
 @Override  
 public String ChooseInanimate(Things t) {  
 return switch (t) {  
 case SUITCASES -> " были пустые чемоданы";  
 case KETTLEBELLS -> "были разные гантели";  
 case GARBAGE -> "был всякий мусор";  
 };  
 }  
  
 @Override  
 public void GoAfk(){  
 System.out.println(SelectWhatSay(  
 String.format("%s поглядел на это всё и отправился дальше спать.", name),  
 "Наш герой решил продолжить отдыхать.")[0]);  
 System.out.println("Вдруг он что-то почувствовал.");  
 }  
 @Override  
 public void NameANewAnimal(Animals a) {  
 System.out.printf("Оказывается, что %s почувствовал, как его ногу облизывают %s.%n", name, ChooseAlive(a));  
 }  
}

ShortySpeakInterface.java

package interfaces;  
  
public interface ShortySpeakInterface {  
 void Say(String name, int num);  
}

Things.java

package enums;  
  
public enum Things {  
 SUITCASES,  
 KETTLEBELLS,  
 GARBAGE  
}

Animals.java

package enums;  
  
public enum Animals{  
 DINOSAURS,  
 COCKROACHES,  
 BUGS  
  
}

ActionInterface.java

package interfaces;  
  
public interface ActionInterface {  
 void TooSleep();  
 void Hide();  
 void GoAfk();  
 void Attack();  
 void Fight();  
}

ChooseAnAttackerInterface.java

package interfaces;  
  
import enums.Animals;  
import enums.Things;  
  
public interface ChooseAnAttackerInterface {  
 String ChooseInanimate(Things t);  
 String ChooseAlive(Animals a);  
}

ToNameInterface.java

package interfaces;  
  
import enums.Animals;  
import enums.Things;  
  
public interface ToNameInterface {  
 void NameWhatItIs(Animals a, Things t);  
 void NameANewAnimal(Animals a);  
  
}

# Вывод программы

Статистика персонажа:

Имя: Незнайка

Возраст: 15

Сила: 150

Привет, меня зовут Незнайка! Я - главный герой этой истории.

Мне 15 лет и я очень сильный!

Я коротышка 1. Меня зовут Алина

Мне стало скучно и я притихла.

Я коротышка 2. Меня зовут Арина

Я замолчала, потому что моей подруге скучно.

Захотев поспать, Незнайка решил сесть на стул и опереться на стену.

Герой лёг поудобнее, закрыл глаза и погрузился в сон

Вдруг произошло что-то неожиданное.

Скоро он почувствовал, что на него напали какие-то маленькие зверушки.

Незнайка счёл это неожиданным.

Незнайка долго дрался с живностью и выиграл.

Незнайка подошёл к животным. Как оказалось, это были жуки.

Незнайка поглядел на это всё и отправился дальше спать.

Вдруг он что-то почувствовал.

Оказывается, что Незнайка почувствовал, как его ногу облизывают динозавры.

Животные отбежали в сторонку и стали поглядывать на незнайку своими глазками. С тревогой осмотревшись по сторонам.

# Вывод

В ходе выполнения данной лабораторной работы я заметил, что в джава можно использовать предупреждение о том, что метод может выбросить исключение. Также исключения делятся на checked и unchecked. Checked исключения - исключения, проверяемые при компиляции программы. Unchecked исключения - исключения, не проверяемые компилятором.