**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 №3  
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](#_Toc1387)**

[Программа должна удовлетворять следующим требованиям: 3](#_Toc17846)

**[Описание предметной области, по которой должна быть построена объектная модель: 3](#_Toc12481)**

**[Диаграмма классов 4](#_Toc17164)**

**[Исходный код программы 5](#_Toc19542)**

**[Вывод программы 12](#_Toc163)**

**[Вывод 13](#_Toc302)**

# Задание:

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.

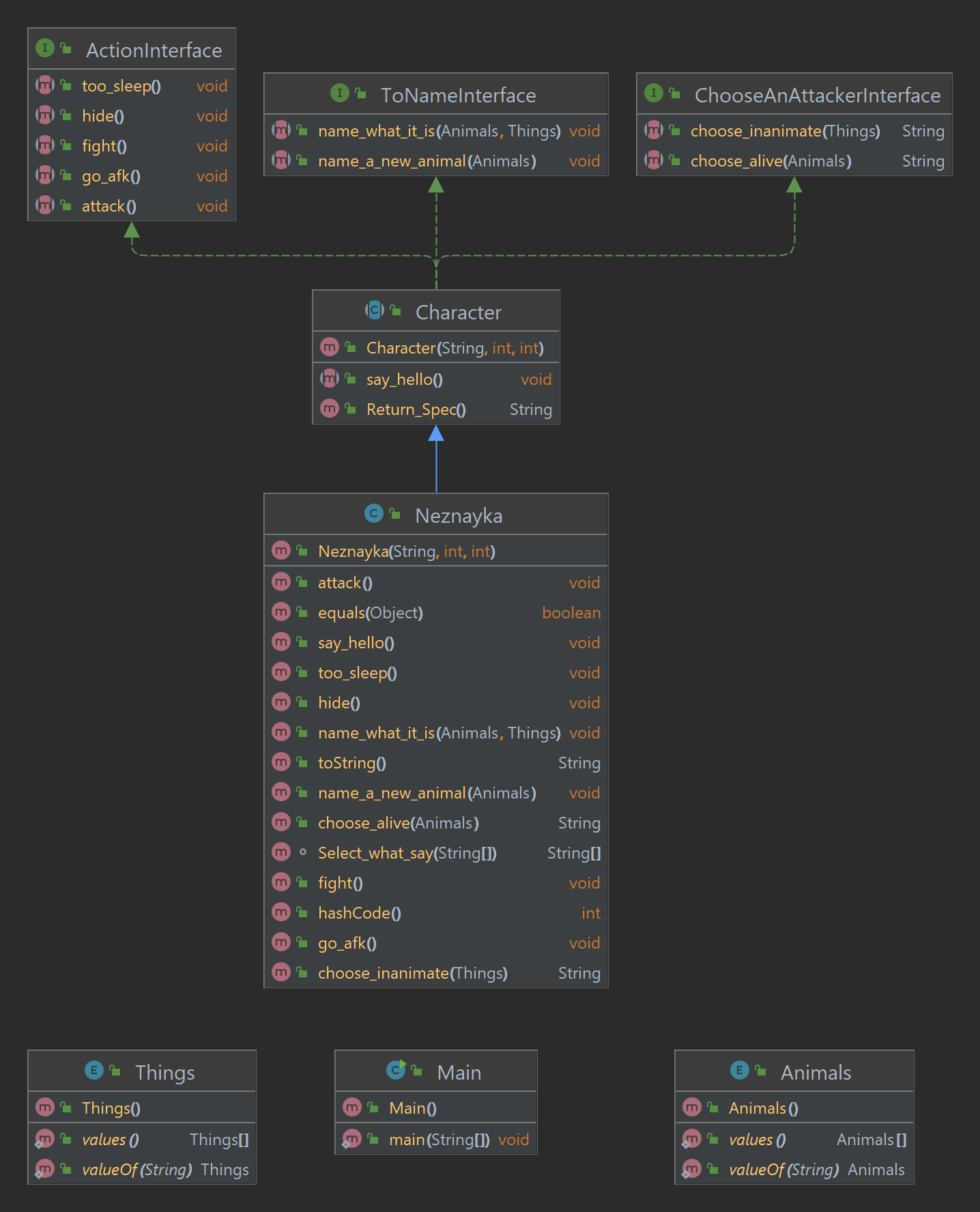
## Программа должна удовлетворять следующим требованиям:

1. Доработанная модель должна соответствовать [принципам SOLID](https://en.wikipedia.org/wiki/SOLID_(object-oriented_design)" \t "https://se.ifmo.ru/courses/_blank).
2. Программа должна содержать как минимум два интерфейса и один абстрактный класс (номенклатура должна быть согласована с преподавателем).
3. В разработанных классах должны быть переопределены методы equals(), toString() и hashCode().
4. Программа должна содержать как минимум один перечисляемый тип (enum).

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

Незнайка решил последовать совету Строптивого и растянулся на полке, не снимая одежды. Скоро он почувствовал, что на него напали какие-то мелкие зверушки и принялись немилосердно кусать. Незнайка чесался, раздирая чуть ли не до крови тело, но это не помогало. Незнайке, однако же, не хотелось дожидаться, когда клопы насытятся его кровью. Правда, некоторое время он терпел, а потом соскочил на пол и принялся сбрасывать с себя этих отвратительных насекомых. Воздух внизу был не такой душный, поэтому Незнайка решил сидеть всю ночь на полу, надеясь, что клопам не придет в голову искать его здесь. Сев на пол и прислонившись спиной к деревянной стойке, он задремал, но в тот же момент почувствовал, как что-то коснулось его ноги. Открыв глаза, он увидел перед собой жирную серую крысу, которая вытянула вперед свою острую усатую мордочку и, шевеля кончиком носа, обнюхивала его ботинок.

# Диаграмма классов



# Исходный код программы

*Main.java*

import Characters.\*;  
import enums.Animals;  
import enums.Things;  
  
public class Main  
{  
 public static void main(String[] args)  
 {  
 Neznayka m = new Neznayka("Незнайка", 15, 150);  
 System.*out*.println(m.toString());  
 m.say\_hello();  
 m.too\_sleep();  
 m.attack();  
 m.hide();  
 m.fight();  
 m.name\_what\_it\_is(Animals.*BUGS*, Things.*GARBAGE*);  
 m.go\_afk();  
 m.name\_a\_new\_animal(Animals.*DINOSAURS*);  
 }  
}

*Character.java*

package Characters;  
  
import Moves.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;  
 }  
  
 public String Return\_Spec(){  
 return String.*format*("Имя: %s\nВозраст: %s\nСила: %s\n", name, age, force);  
 }  
  
 public abstract void say\_hello();  
  
}

Neznayka.java

package Characters;  
  
import enums.Animals;  
import enums.Things;  
  
import java.util.Objects;  
import static java.lang.Math.\*;  
  
public class Neznayka extends Character {  
 String strenght;  
 boolean is\_dropped\_down = false;  
 public Neznayka(String name, int age, int force) {  
 super(name, age, force);  
 }  
  
 String[] Select\_what\_say(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 say\_hello(){  
 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);  
 }  
  
  
 @Override  
 public void too\_sleep() {  
 System.*out*.println(Select\_what\_say(  
 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 = Select\_what\_say(  
 "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(Select\_what\_say(  
 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(Select\_what\_say(  
 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(Select\_what\_say(  
 "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(Select\_what\_say(  
 String.*format*("%s долго дрался с живностью и выиграл.", name),  
 String.*format*("%s победил зверушек.", name))[0]);  
 }  
  
 @Override  
 public void name\_what\_it\_is(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, choose\_alive(a));  
 else  
 System.*out*.printf("%s подошёл к разбросанным вещам. Как оказалось, это %s.%n", name, choose\_inanimate(t));  
 }  
  
 @Override  
 public String choose\_alive(Animals a) {  
 return switch (a) {  
 case *BUGS* -> "жуки";  
 case *COCKROACHES* -> "тараканы";  
 case *DINOSAURS* -> "динозавры";  
 };  
 }  
  
 @Override  
 public String choose\_inanimate(Things t) {  
 return switch (t) {  
 case *SUITCASES* -> " были пустые чемоданы";  
 case *KETTLEBELLS* -> "были разные гантели";  
 case *GARBAGE* -> "был всякий мусор";  
 };  
 }  
  
 @Override  
 public void go\_afk(){  
 System.*out*.println(Select\_what\_say(  
 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 name\_a\_new\_animal(Animals a) {  
 System.*out*.printf("Оказывается, что %s почувствовал, как его ногу облизывают %s.%n", name, choose\_alive(a));  
 }  
}

Animals.java

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

Things.java

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

ChooseAnAttackerInterface.java

package interfaces;  
  
import enums.Animals;  
import enums.Things;  
  
public interface ChooseAnAttackerInterface {  
 String choose\_inanimate(Things t);  
 String choose\_alive(Animals a);  
}

ActionInterface.java

package interfaces;  
  
public interface ActionInterface {  
 void too\_sleep();  
 void hide();  
 void go\_afk();  
 void attack();  
 void fight();  
}

ToNameInterface.java

package interfaces;  
  
import enums.Animals;  
import enums.Things;  
  
public interface ToNameInterface {  
 void name\_what\_it\_is(Animals a, Things t);  
 void name\_a\_new\_animal(Animals a);  
  
}

Ссылка на гитхаб с исходником: <https://github.com/frizyyu/lab3_prog>

# 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!

Незнайка лёг на полку, чтобы хорошенько отдохнуть.

Герой лёг поудобнее, закрыл глаза и погрузился в сон

Вдруг произошло что-то неожиданное.

Скоро он почувствовал, что на него напали будто какие-то насекомые и начали кусать.

Незнайка был растерян, поэтому просто сел на пол.

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.

# Вывод

В ходе выполнения лабораторной работы я заметил, что программу, написанную по принципам SOLID легче модифицировать и изменить.

Принципы SOLID включают в себя:

* Принцип единой ответственности
* Принцип открытости/закрытости
* Принцип заменяемости
* Принцип разделения интерфейсов
* Принцип инверсии зависимостей

Список литературы

<https://javarush.com/groups/posts/osnovnye-principy-dizajna-klassov-solid-v-java>

<https://javarush.com/groups/posts/principy-oop>