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 №3-4**

**По программированию**

**Вариант: 8500-8500.141**

Выполнил:

студент 1 курса

Батманов Даниил Евгеньевич

Группа: Р3107

Принял:

Гаврилов Антон Валерьевич

Отчёт принят «\_\_»\_\_\_\_\_2022 г.

Оценка: \_\_\_\_\_\_\_\_\_\_\_

г. Санкт-Петербург, 2022

**Оглавление**

[Задание 3](#_Toc118222211)

[Диаграмма классов реализованной объектной модели 4](#_Toc118222212)

[Исходный код программы 5](#_Toc118222213)

[Результат работы программы 13](#_Toc118222214)

[Вывод 16](#_Toc118222215)

[Список литературы 17](#_Toc118222216)

# 

# 

# 

# Задание

* Доработать объектную модель приложения.
* Перерисовать диаграмму классов в соответствии с внесёнными в модель изменениями.
* Согласовать с преподавателем изменения, внесённые в модель.
* Модифицировать программу в соответствии с внесёнными в модель изменениями.

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

- Нечего там! - сказала Королева.- Сперва приговор, посовещаются потом! - Как не стыдно! - во весь голос заявила Алиса.- Стыдно даже болтать такие глупости! - Молчать! - крикнула Королева, багровея от ярости. - Как же! - сказала Алиса. - Отрубить ей голову! - завопила Королева во всю глотку. Никто не пошевелился. - Да кто вас боится! - сказала Алиса (она уже достигла своего настоящего роста).- Вы просто несчастные карты - и все! И при этих словах вся колода карт взвилась в воздух и полетела ей в лицо; Алиса вскрикнула - полуиспуганно, полусердито,- стала от них отбиваться... и вдруг оказалось, что она лежит на берегу, положив голову сестре на колени, а та осторожно смахивает с ее личика сухие листья, слетевшие с соседнего дерева. - Просыпайся, дорогая,- сказала сестра.- Что-то ты очень разоспалась! - Ой, а какой я забавный сон видела! - сказала Алиса. И она принялась рассказывать сестре все, что сумела запомнить, про свои странные приключения - то есть все то, что вы только что прочитали. А когда она закончила, сестра поцеловала ее и сказала: - Сон был и правда очень занятный, а сейчас беги пить чай, а то опоздаешь. Алиса послушно вскочила и побежала домой, но и по дороге она все думала, какой же это был чудесный сон - сон, который, наверно, никогда не забудешь...

# Диаграмма классов реализованной объектной модели

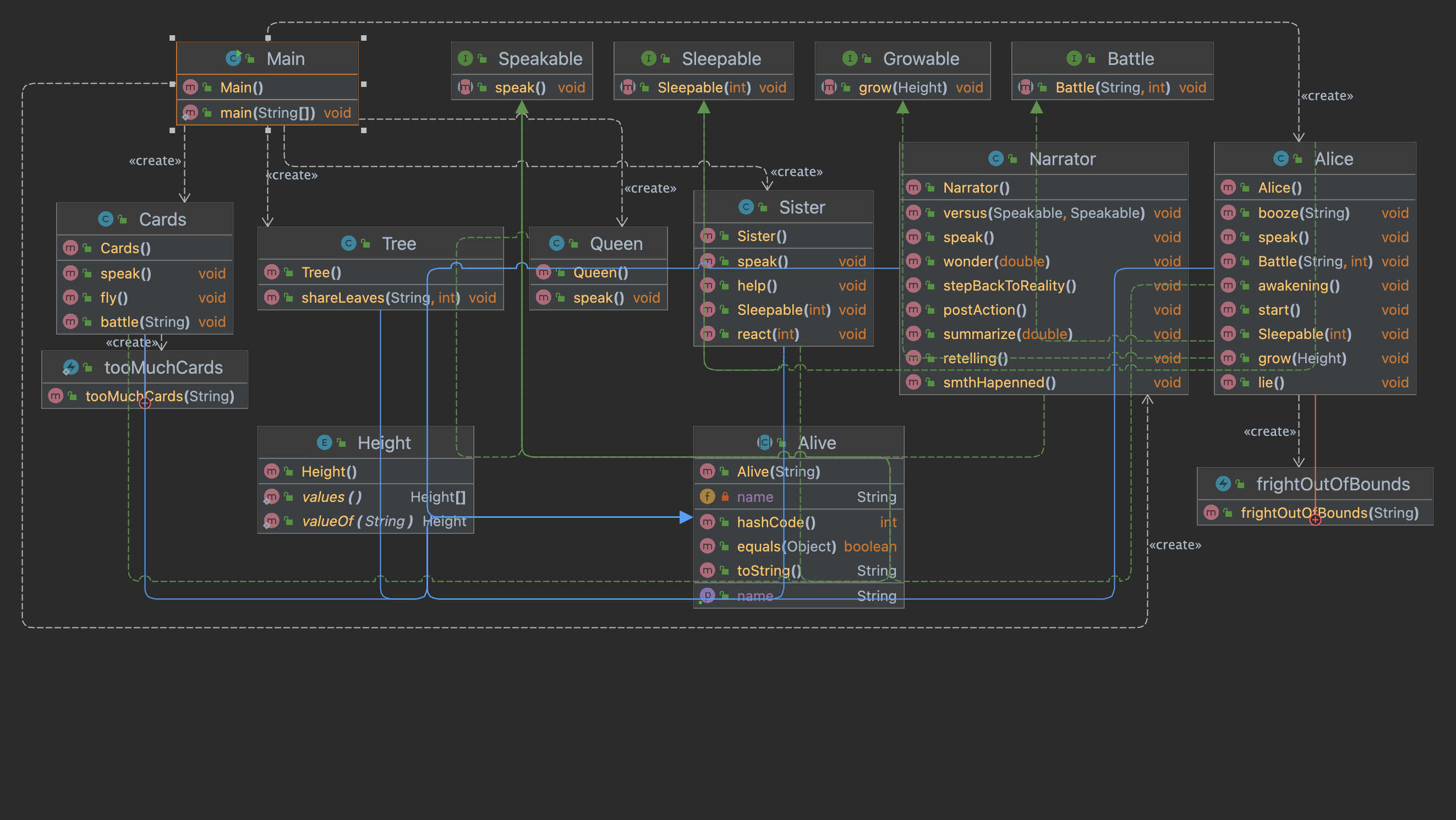


Рисунок 1

# Исходный код программы

**Alice**

public class Alice extends Alive implements Growable, Speakable, Battle, Sleepable {  
 public Alice() {  
 super("Алиса");  
 }  
  
 int fright = (int) (Math.*random*() \* 4);  
  
 @Override  
 public void speak() {  
 String[] words = new String[10];  
 words[0] = "Алиса: \"Я тебе волосы вырву!\"";  
 words[1] = "Алиса: \"Не затыкай мне рот!\"";  
 words[2] = "Алиса: \"Я в ярости\"";  
 words[3] = "Алиса: \"Не испытывай мои нервы\"";  
 words[4] = "Алиса: \"Не говори глупостей\"";  
 words[5] = "Алиса: \"Постыдись своей глупости\"";  
 words[6] = "Алиса: \"Я тебе голову отрублю!\"";  
 words[7] = "Алиса: \*Крик во всю глотку\*";  
 words[8] = "Алиса: \"Стой где стоишь!\"";  
 words[9] = "Алиса: \"Не шевелись!\"";  
 System.*out*.println(words[(int) (Math.*random*() \* 9)]);  
 }  
  
 @Override  
 public void grow(Height height) {  
 System.*out*.print(getName() + " достигла");  
 switch(height) {  
 case *TINY\_LITTLE*:  
 System.*out*.println(" совсем крохотного роста!");  
 break;  
 case *TINY*:  
 System.*out*.println(" крохотного роста!");  
 break;  
 case *PETTIT*:  
 System.*out*.println(" малюсенького роста!");  
 break;  
 case *SMALL*:  
 System.*out*.println(" маленького роста!");  
 break;  
 case *REAL*:  
 System.*out*.println(" реального роста!");  
 break;  
 default:  
 break;  
 }  
 }  
  
 public class frightOutOfBounds extends IndexOutOfBoundsException {  
 public frightOutOfBounds(String message) {  
 super(message);  
 }  
 }  
  
 public void booze(String target) throws frightOutOfBounds {  
 fright++;  
 String[] egg = new String[5];  
 egg[0] = "Жалкие " + target + "! Я вас совсем не боюсь!";  
 egg[1] = "Милые " + target + "! Я бы вас затискала";  
 egg[2] = "Не такие уж вы и жалкие " + target +"Я готова сразиться!";  
 egg[3] = "Я боюсь вас, " + target + "! Кто-нибудь, помогите мне!";  
 egg[4] = "Я очень сильно боюсь вас, " + target + "! Мне очень страшно!";  
 if (fright >= 0 & fright <= 4) System.*out*.println(egg[fright]);  
 else {  
 fright--;  
 throw new frightOutOfBounds("А вот программа и сломалась!");  
 }  
 }  
  
 public void start() {  
 System.*out*.println(getName() + " начинает расти");  
 }  
  
 @Override  
 public void Battle(String object, int num) {  
 if (num > 0 & num < 25) {  
 System.*out*.println(getName() + " стала отбиваться от " + object + " смеясь, ведь их совсем мало");  
 fright = 0;  
 }  
 else if (num > 25 & num < 50) {  
 System.*out*.println(getName() + " стала отбиваться от " + object + " с небольшой опаской");  
 fright = 1;  
 }  
 else if (num > 50 & num < 75) {  
 System.*out*.println(getName() + " стала отбиваться от " + object + " крича, их так много!");  
 fright = 2;  
 }  
 else if (num > 75 & num < 100) {  
 System.*out*.println(getName() + " стала отбиваться от " + object + " рыдая, их слишком много!");  
 fright = 4;  
 }  
 else {  
 System.*out*.println(getName() + " стала отбиваться от " + object + " не понимая, что происходит");  
 fright = 3;  
 }  
 }  
  
 public void lie() {  
 String[] place = new String[4];  
 place[0] = " берегу реки";  
 place[1] = " листве в парке";  
 place[2] = " опушке в лесу";  
 place[3] = " на траве";  
 System.*out*.println(getName() + " лежит на" + place[(int) (Math.*random*() \* 2) + 1]);  
 }  
  
 @Override  
 public void Sleepable(int lot) {  
 String[] way = new String[5];  
 way[0] = " очень крепко";  
 way[1] = " крепко";  
 way[2] = " не очень крепко";  
 way[3] = " некрепко";  
 way[4] = " тревожно";  
 System.*out*.println(getName() + way[fright] + " спит");  
 }  
  
 public void awakening() {  
 String[] type = new String[5];  
 type[0] = " прекрасный";  
 type[1] = " нормальный";  
 type[2] = " странный";  
 type[3] = " ужасный";  
 type[4] = " кошмарный";  
 System.*out*.println(getName() + ": \"Мне приснился" + type[fright] + " сон\"");  
 }  
}

**Alive**

public abstract class Alive {  
 private String name;  
 public Alive(String name) {  
 this.name = name;  
 }  
  
 public String getName() {  
 return this.name;  
 }  
  
 @Override  
 public int hashCode() {  
 return super.hashCode() + this.getName().hashCode();  
 }  
  
 @Override  
 public boolean equals(Object obj) {  
 return obj.hashCode() == this.hashCode();  
 }  
  
 @Override  
 public String toString() {  
 return "Имя" + getName();  
 }  
}

**Battle**

public interface Battle {  
 public void Battle(String object, int sum);  
}

**Cards**

import jdk.jshell.spi.ExecutionControlProvider;  
  
public class Cards extends Alive implements Speakable {  
 public Cards() {  
 super("Карты");  
 }  
  
 int howMuch = (int) (Math.*random*() \* 99) + 1;  
  
 @Override  
 public void speak() {  
 System.*out*.println("\*Шелест карт\*");  
 }  
  
 public void fly() {  
 System.*out*.println(getName() + " в количестве " + howMuch + " единиц" + " взвивают в воздух на " + (howMuch % 4 + 1) + " метров");  
 }  
  
 public static class tooMuchCards extends Exception {  
 public tooMuchCards(String message) {  
 super(message);  
 }  
 }  
  
 public void battle(String wh) throws tooMuchCards {  
 howMuch += 2;  
 if (howMuch > 0 & howMuch < 25) System.*out*.println(getName() + " атакуют в лицо " + wh + " опасаясь поражения");  
 else if (howMuch > 25 & howMuch < 50) System.*out*.println(getName() + " атакуют в лицо " + wh + " понимая шансы проиграть");  
 else if (howMuch > 50 & howMuch < 75) System.*out*.println(getName() + " атакуют в лицо " + wh + " с предвкушением победы");  
 else if (howMuch > 75 & howMuch < 100) System.*out*.println(getName() + " атакуют в лицо " + wh + " уверенно, веря на все 100 в свою победу");  
 else {  
 throw new tooMuchCards(getName() + " атакуют в лицо " + wh + " не теряя надежду");  
 }  
 howMuch -= 2;  
 }  
  
}

**Growable**

public interface Growable {  
 public void grow(Height height);  
}

**Height**

public enum Height {  
 *TINY\_LITTLE*,  
 *TINY*,  
 *PETTIT*,  
 *SMALL*,  
 *REAL*}

**Main**

//lab3,4, var: 8500.141  
public class Main {  
 public static void main(String[] args) {  
 Queen queen = new Queen();  
 Narrator narrator = new Narrator();  
 Alice alice = new Alice();  
 Cards cards = new Cards();  
 Sister sister = new Sister();  
 Tree tree = new Tree();  
  
 narrator.speak();  
 narrator.smthHapenned();  
 narrator.versus(queen, alice);  
 narrator.wonder(Math.*random*());  
  
 alice.start();  
 alice.grow(Height.*TINY\_LITTLE*);  
 alice.grow(Height.*TINY*);  
 alice.grow(Height.*PETTIT*);  
 alice.grow(Height.*SMALL*);  
 alice.grow(Height.*REAL*);  
  
 cards.speak();  
 try { alice.booze("Карты"); }  
 catch (Alice.frightOutOfBounds exp) {  
 System.*out*.println(exp.getMessage());  
 }  
 cards.speak();  
 cards.fly();  
 try { cards.battle("Алиса"); }  
 catch (Cards.tooMuchCards ogg) {  
 System.*out*.println(ogg.getMessage());  
 }  
 alice.Battle("Карты", cards.howMuch);  
  
 narrator.stepBackToReality();  
  
 alice.lie();  
 tree.shareLeaves("Алиса", cards.howMuch);  
 alice.Sleepable(alice.fright);  
 sister.speak();  
 sister.help();  
 sister.Sleepable(alice.fright);  
 alice.awakening();  
 narrator.retelling();  
 sister.react(alice.fright);  
 narrator.postAction();  
  
 narrator.summarize(Math.*random*());  
 }  
}

**Narrator**

public class Narrator extends Alive implements Speakable {  
 public Narrator() {  
 super("Рассказчик");  
 }  
  
 public void versus(Speakable q, Speakable a) {  
 int x = 0;  
 int y = 0;  
 for (int i = 0; i <= 2 \* (int) (Math.*random*() \* 5) + 5; i++) {  
 int r = (int) (Math.*random*() + 0.5);  
 if (r == 1) {  
 a.speak();  
 x++;  
 }  
 else {  
 q.speak();  
 y++;  
 }  
 }  
 if (x > y) System.*out*.println("В этой словесной перепалке победила Алиса");  
 else System.*out*.println("В этой словесной перепалке победила Королева");  
 }  
  
 @Override  
 public void speak() {  
 String[] mas = new String[5];  
 mas[0] = " очень давно";  
 mas[1] = " давным-давно";  
 mas[2] = " совсем недавно";  
 mas[3] = " пару дней назад";  
 mas[4] = " больше 1000 лет назад";  
 System.*out*.println("Случилось это" + mas[(int) (Math.*random*() \* 3) + 1]);  
 }  
  
 public void wonder(double shock) {  
 String[] arr = new String[6];  
 arr[0] = "Вау! Посмотрите!!!";  
 arr[1] = "Ничего необычного, так происходит при каждом запуске программы" + "\n" + "Сами посмотрите:";  
 arr[2] = "Это просто невероятно! Я никогда ещё не видел ничего подобного! Смотрите:";  
 arr[3] = "Бесподобно! Вы только посмотрите!";  
 arr[4] = "Все сюда! Что-то неладное творится! \*Раздаётся звон колокола\*";  
 arr[5] = "Ой, что это?";  
 System.*out*.println(arr[(int) (shock \* 4) + 1]);  
 }  
  
 public void summarize(double mood) {  
 String[] end = new String[6];  
 end[0] = "Вот и сказочке конец, а кто слушал – молодец";  
 end[1] = "Сказка ложь, да в ней намёк: добрым молодцам урок";  
 end[2] = "А вот что произошло дальше – это уже совсем другая история";  
 end[3] = "Выводы делайте сами, потому что я ничего не понял";  
 end[4] = "Сказка кончилась, это печально";  
 end[5] = "Конец!";  
 System.*out*.println(end[(int) (mood \* 4) + 1]);  
 }  
  
 public void stepBackToReality() {  
 System.*out*.println("Яркая вспышка, громкий звук: все резко вернулись в реальность");  
 }  
  
 public void smthHapenned() {  
 System.*out*.println("<...>");  
 }  
  
 public void retelling() {  
 System.*out*.println("Алиса принялась рассказывать про сон, который ей приснился, про все её приключения");  
 }  
  
 public void postAction() {  
 System.*out*.println("Алиса собралась, попрощалась с сестрой и убежала по своим делам" + '\n' + "Такой сон она никогда не забудет!");  
 }  
}

**Queen**

public class Queen extends Alive implements Speakable {  
 public Queen() {  
 super("Королева");  
 }  
  
 @Override  
 public void speak() {  
 String[] phrases = new String[10];  
 phrases[0] = "Королева: \"Я тебе волосы вырву!\"";  
 phrases[1] = "Королева: \"Не затыкай мне рот!\"";  
 phrases[2] = "Королева: \"Я в ярости\"";  
 phrases[3] = "Королева: \"Не испытывай мои нервы\"";  
 phrases[4] = "Королева: \"Не говори глупостей\"";  
 phrases[5] = "Королева: \"Постыдись своей глупости\"";  
 phrases[6] = "Королева: \"Я тебе голову отрублю!\"";  
 phrases[7] = "Королева: \*Крик во всю глотку\*";  
 phrases[8] = "Королева: \"Стой где стоишь!\"";  
 phrases[9] = "Королева: \"Не шевелись!\"";  
 System.*out*.println(phrases[(int) (Math.*random*() \* 9)]);  
 }  
}

**Sister**

public class Sister extends Alive implements Speakable, Sleepable {  
 public Sister() {  
 super("Сестра Алисы");  
 }  
  
 @Override  
 public void speak() {  
 System.*out*.println(getName() + " молчит");  
 }  
  
 @Override  
 public void Sleepable(int lot) {  
 String[] method = new String[5];  
 method[0] = " нехотя, ведь она очень крепко спит";  
 method[1] = " нехотя, ведь она крепко спит";  
 method[2] = " охотно, ведь она не очень крепко спит";  
 method[3] = " охотно, ведь она некрепко спит";  
 method[4] = " желанно, ведь она тревожно спит";  
 System.*out*.println(getName() + " будит Алису" + method[lot]);  
 }  
  
 public void help() {  
 System.*out*.println(getName() + " смахивает листья, летящие на сестрёнку");  
 }  
  
 public void react(int stat) {  
 String[] how = new String[5];  
 how[0] = " хорошим";  
 how[1] = " интересным";  
 how[2] = " занятным";  
 how[3] = " страшным";  
 how[4] = "-очень страшным";  
 System.*out*.println(getName() + ": \"Сон и правда оказался очень" + how[stat] + "! Но уже пора торипиться, собирайся скорее");  
 }  
}

**Sleepable**

public interface Sleepable {  
 public void Sleepable(int lot);  
}

**Speakable**

public interface Speakable {  
 public void speak();  
}

**Tree**

public class Tree extends Alive {  
 public Tree() {  
 super("Дерево");  
 }  
  
  
 public void shareLeaves(String forWhom, int sum) {  
 System.*out*.println(getName() + " осыпает лицо " + forWhom + " " + sum + " единицами листьев");  
 }  
}

# Результат работы программы

# Случилось это давным-давно

# <...>

# Королева: "Я в ярости"

# Королева: "Не говори глупостей"

# Королева: "Не затыкай мне рот!"

# Королева: "Постыдись своей глупости"

# Алиса: "Постыдись своей глупости"

# Королева: "Я тебе волосы вырву!"

# Королева: "Не говори глупостей"

# Алиса: \*Крик во всю глотку\*

# Алиса: "Не говори глупостей"

# Королева: "Я тебе голову отрублю!"

# Королева: "Я в ярости"

# В этой словесной перепалке победила Королева

# Ничего необычного, так происходит при каждом запуске программы

# Сами посмотрите:

# Алиса начинает расти

# Алиса достигла совсем крохотного роста!

# Алиса достигла крохотного роста!

# Алиса достигла малюсенького роста!

# Алиса достигла маленького роста!

# Алиса достигла реального роста!

# \*Шелест карт\*

# Я боюсь вас, Карты! Кто-нибудь, помогите мне!

# \*Шелест карт\*

# Карты в количестве 96 единиц взвивают в воздух на 1 метров

# Карты атакуют в лицо Алиса уверенно, веря на все 100 в свою победу

# Алиса стала отбиваться от Карты рыдая, их слишком много!

# Яркая вспышка, громкий звук: все резко вернулись в реальность

# Алиса лежит на опушке в лесу

# Дерево осыпает лицо Алиса 96 единицами листьев

# Алиса тревожно спит

# Сестра Алисы молчит

# Сестра Алисы смахивает листья, летящие на сестрёнку

# Сестра Алисы будит Алису желанно, ведь она тревожно спит

# Алиса: "Мне приснился кошмарный сон"

# Алиса принялась рассказывать про сон, который ей приснился, про все её приключения

# Сестра Алисы: "Сон и правда оказался очень-очень страшным! Но уже пора торипиться, собирайся скорее

# Алиса собралась, попрощалась с сестрой и убежала по своим делам

# Такой сон она никогда не забудет!

# Сказка кончилась, это печально

# Вывод

В ходе выполнения данной лабораторной работы я научился переопределять методы, работать с интерфейсами, абстрактными и вложенными классами, ознакомился с принципами SOLID, научился работать с enum и исключениями.

# Список литературы

Oracle Java Documentation. - Т. https://docs.oracle.com/en/java/.

Михалевский Петр Легкий старт в Java <...>. - [б.м.] : Stepik. - Т. https://stepik.org/course/90684/syllabus.