Федеральное государственное автономное образовательное учреждение высшего образования **«Научно-образовательная корпорация ИТМО»**

Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа по программированию №3**

вариант: 310914

Преподаватель: Наумова Надежда Александровна

Выполнил: Степанов Арсений

Группа: Р3109

Оглавление

[Цели 3](#_Toc117803320)

[Задание 3](#_Toc117803321)

[Код классов 3](#_Toc117803322)

[UML Диаграммы 11](#_Toc117803323)

[Вывод программы 12](#_Toc117803324)

[Итоги 15](#_Toc117803325)

# Цели

Изучить принципы SOLID и научиться применять их на практике. Научиться работать с классом Object и переопределять его методы.

# 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.

# Предметная область

Иа вошел в дом и снова вышел. Все попрощались со счастливым хозяином дома, и Кристофер Робин пошел обедать со своими друзьями -- Пухом и Пятачком. По дороге друзья рассказали ему об Ужасной Ошибке, которую они совершили, и, когда он кончил смеяться, все трое дружно запели Дорожную Шумелку для Снежной Погоды и пели ее всю дорогу, причем Пятачок, который все еще был немного не в голосе, только тирлимбомбомкал.

# Код классов

Entity:

package ru.armemius.lab3;  
  
import java.util.Objects;  
  
public abstract class Entity implements ITalker, IMoving {  
 public Entity(String name, Status status) {  
 this.name = name;  
 this.status = status;  
 }  
  
 private final String name;  
 protected Status status;  
  
 public String getName() {  
 return this.name;  
 }  
  
 @Override  
 public void moveTo(String name, MoveAction action, IMoving... companions) {  
 if (companions.length == 0) {  
 System.*out*.println(this + " идёт в: " + name);  
 action.action();  
 System.*out*.println(this + " пришёл");  
 } else {  
 String names = this.toString();  
 for (int it = 0; it < companions.length; ++it) {  
 if (it + 1 == companions.length) {  
 names += " и " + companions[it].toString();  
 } else {  
 names += ", " + companions[it].toString();  
 }  
 }  
 System.*out*.println(names + " идут в: " + name);  
 action.action();  
 System.*out*.println(names + " пришли");  
 }  
  
 }  
  
 @Override  
 public void tell(ITalker name, String text, ITalker... companions) {  
 if (companions.length == 0) {  
 System.*out*.println(this + " говорит " + name + ": " + text);  
 } else {  
 String names = this.toString();  
 for (int it = 0; it < companions.length; ++it) {  
 if (it + 1 == companions.length) {  
 names += " и " + companions[it].toString();  
 } else {  
 names += ", " + companions[it].toString();  
 }  
 }  
 System.*out*.println(names + " говорят " + name + ": " + text);  
 }  
 }  
  
 @Override  
 public String toString() {  
 return name;  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Entity entity = (Entity) o;  
 return Objects.*equals*(name, entity.name) && status == entity.status;  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(name, status);  
 }  
}

FriendEntity:

package ru.armemius.lab3;  
  
public class FriendEntity extends Entity implements ISinger {  
 public FriendEntity(String name, Status status) {  
 super(name, status);  
 }  
  
 @Override  
 public void sing(String name) {  
 if (super.status == Status.*BAD\_VOICE*) {  
 System.*out*.println(this + " тирлимбомбомкает");  
 } else {  
 System.*out*.println(this + " поёт " + name);  
 }  
 }  
  
 @Override  
 public String toString() {  
 return "Друг " + super.toString();  
 }  
  
 @Override  
 public String getName() {  
 return "Друг " + super.getName();  
 }  
}

House:

package ru.armemius.lab3;  
  
public class House {  
 static void enter(IMoving mover) {  
 System.*out*.println(mover.toString() + " зашёл в дом");  
 }  
  
 static void leave(IMoving mover) {  
 System.*out*.println(mover.toString() + " вышел из дома");  
 }  
}

IMoving:

package ru.armemius.lab3;  
  
public interface IMoving {  
 void moveTo(String name, MoveAction action, IMoving... companions);  
}  
  
@FunctionalInterface  
interface MoveAction {  
 void action();  
}

ISinger:

package ru.armemius.lab3;  
  
public interface ISinger {  
 void sing(String name);  
}

ITalker:

package ru.armemius.lab3;  
  
public interface ITalker {  
 void tell(ITalker name, String text, ITalker... companions);  
}

Main:

package ru.armemius.lab3;  
  
public class Main {  
 public static void main(String[] args) {  
 FriendEntity ia = new FriendEntity("Ослик Иа", Status.*NEUTRAL*);  
 FriendEntity robin = new FriendEntity("Кристофер Робин", Status.*NEUTRAL*);  
 FriendEntity pooh = new FriendEntity("Винни-Пух", Status.*NEUTRAL*);  
 FriendEntity piglet = new FriendEntity("Пятачок", Status.*BAD\_VOICE*);  
 Entity owner = new StatusEntity("Владелец дома", Status.*HAPPY*);  
 House.*enter*(ia);  
 House.*leave*(ia);  
 ia.tell(owner, "До связи", robin, pooh, piglet);  
 robin.moveTo("Место для обеда", () -> {  
 ia.tell(robin, "Мы совершили ужасную ошибку", pooh, piglet);  
 ia.sing("Дорожная Шумелка для Снежной Погоды");  
 robin.sing("Дорожная Шумелка для Снежной Погоды");  
 pooh.sing("Дорожная Шумелка для Снежной Погоды");  
 piglet.sing("Дорожная Шумелка для Снежной Погоды");  
 }, ia, pooh, piglet);  
 }  
}  
  
/\*  
Иа вошел в дом и снова вышел. Все попрощались со счастливым хозяином дома,  
и Кристофер Робин пошел обедать со своими друзьями -- Пухом и Пятачком.  
По дороге друзья рассказали ему об Ужасной Ошибке, которую они совершили,  
и, когда он кончил смеяться, все трое дружно запели Дорожную Шумелку для Снежной Погоды и пели ее всю дорогу,  
причем Пятачок, который все еще был немного не в голосе, только тирлимбомбомкал.  
\*/

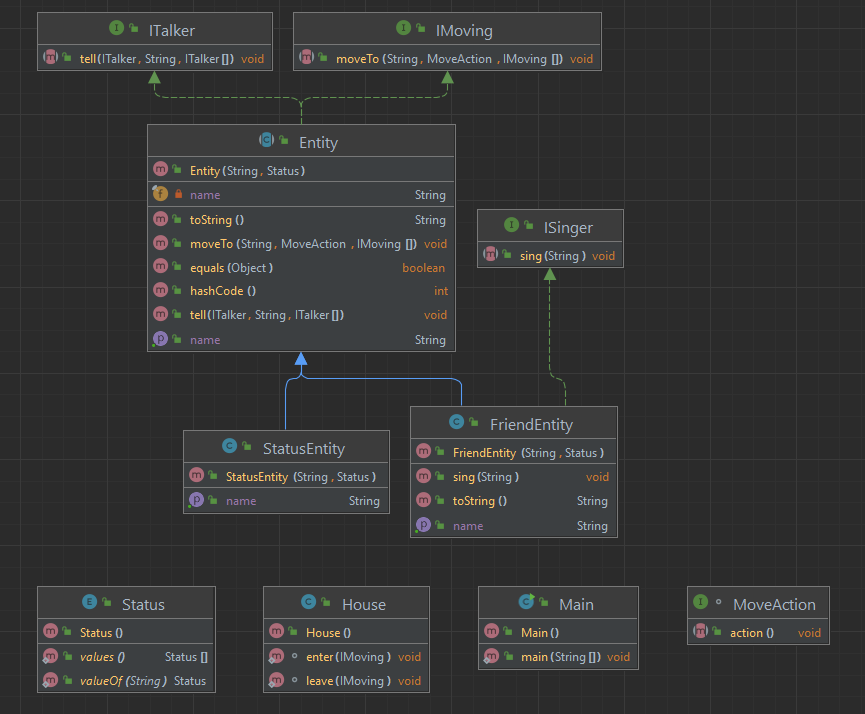
Status:

package ru.armemius.lab3;  
  
public enum Status {  
 *HAPPY*,  
 *BAD\_VOICE*,  
 *NEUTRAL*}

StatusEntity:

package ru.armemius.lab3;  
  
public class StatusEntity extends Entity{  
 public StatusEntity(String name, Status status) {  
 super(name, status);  
 }  
  
 @Override  
 public String getName() {  
 return (status == Status.*HAPPY* ? "Счастливый " : "") + super.getName();  
 }  
}

# UML Диаграммы



# Вывод программы

Друг Ослик Иа зашёл в дом

Друг Ослик Иа вышел из дома

Друг Ослик Иа, Друг Кристофер Робин, Друг Винни-Пух и Друг Пятачок говорят Владелец дома: До связи

Друг Кристофер Робин, Друг Ослик Иа, Друг Винни-Пух и Друг Пятачок идут в: Место для обеда

Друг Ослик Иа, Друг Винни-Пух и Друг Пятачок говорят Друг Кристофер Робин: Мы совершили ужасную ошибку

Друг Ослик Иа поёт Дорожная Шумелка для Снежной Погоды

Друг Кристофер Робин поёт Дорожная Шумелка для Снежной Погоды

Друг Винни-Пух поёт Дорожная Шумелка для Снежной Погоды

Друг Пятачок тирлимбомбомкает

Друг Кристофер Робин, Друг Ослик Иа, Друг Винни-Пух и Друг Пятачок пришли

Process finished with exit code 0

# Итоги

Я научился работать с классами и базовыми принципами объектно-ориентированного программирования, а также с пакетами. Полученные в процессе работы знания являются фундаментальными и будут полезны при будущей работе и учёбе.