Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики»

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА № 3**

по дисциплине

‘ПРОГРАММИРОВАНИЕ’

Вариант №456345

Выполнил:

Студент группы P3118

Шульга Артём Игоревич

Преподаватель:

Сорокин Роман Борисович



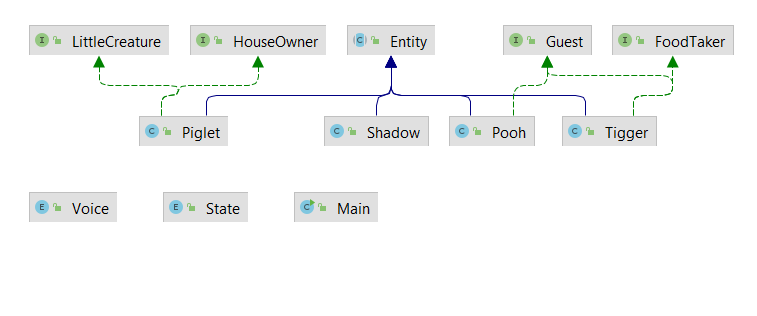
Санкт-Петербург, 2021

Задание:

Изображение выглядит как текст

Автоматически созданное описание

Диаграмма реализованных классов



// Main.java

Исходный код

**package** com.company;  
**public class** Main {  
 **public static void** main(String[] args) {  
 Pooh pooh = **new** Pooh(**"Пух"**,State.***STAY***);  
 Tigger tigger = **new** Tigger(**"Тигра"**,State.***STAY***);  
 Piglet piglet = **new** Piglet(**"Пятачок"**,State.***AT\_HOUSE***);  
 pooh.startJourneyWith(tigger,**"дом Пяточка"**);  
 Shadow shadowOfPooh = **new** Shadow(**"Тень Пуха"**,pooh.getState());  
 pooh.tellAboutLittleCreature(tigger,piglet);  
 tigger.playWithShadow(shadowOfPooh);  
 pooh.endJourneyWith(tigger,piglet);  
 piglet.giveFood(tigger);  
 piglet.hug(pooh);  
 piglet.speak(**"Так ты Тигра? Ну-ну!"**);  
 tigger.speak(**"Да да я"**);  
 }  
}

// FoodTaker.java

**package** com.company;  
  
**public interface** FoodTaker {  
 **void** takeFood();  
}

// Guest.java

**package** com.company;  
  
**public interface** Guest {  
 **void** enterInHouse();  
}

// HouseOwner.java

**package** com.company;  
  
**public interface** HouseOwner {  
 **void** giveFood(FoodTaker foodTaker);  
 **void** openDoor(Guest[] guests);  
}

// LittleCreature.java

**package** com.company;  
  
**public interface** LittleCreature {  
 String whoAmI();  
 **void** troubles();  
}

// Entity.java

**package** com.company;  
  
**public abstract class** Entity {  
 **private** Voice **voice** = Voice.***NORMAL***;  
 **protected final** String **name**;  
 **private** State **state**;  
 Entity(**final** String name, **final** State state){  
 **this**.**name** = name;  
 **this**.**state** = state;  
 }  
 **protected void** setState(State state){  
 **if**(**this**.**state**.endState() != **null**)  
 System.***out***.println(**"\*"**+**name**+**" "**+**this**.**state**.endState()+**"\*"**);  
 **this**.**state** = state;  
 **if**(state.startState() != **null**)  
 System.***out***.println(**"\*"**+**name**+**" "**+state.startState()+**"\*"**);  
 }  
 **public** State getState(){  
 **return state**;  
 }  
 **public void** speak(String phrase){  
 **if**(getVoice().**isSilence**)  
 System.***out***.println(**name**+**" "**+getVoice().getVoicePhrase());  
 **else** System.***out***.println(**name**+**" "**+getVoice().getVoicePhrase()+**": "**+phrase);  
 }  
 **protected void** setVoice(Voice voice) {  
 System.***out***.println(**"\*голос у "**+**name**+**" меняется на "**+voice.getName()+**"\*"**);  
 **this**.**voice** = voice;  
 }  
 **private** Voice getVoice(){  
 **return voice**;  
 }  
 @Override  
 **public** String toString(){  
 **return name**;  
 }  
}

// Voice.java

**package** com.company;  
  
**public enum** Voice {  
 ***NORMAL***(**"нормальный"**, **false**) {  
 @Override  
 **public** String getVoicePhrase() {  
 **return "говорит"**;  
 }  
 },  
 ***ALMOST\_FUNNY***(**"почти весёлый"**, **false**) {  
 @Override  
 **public** String getVoicePhrase() {  
 **return "почти весёлым голосом произносит"**;  
 }  
 },  
 ***CLOGGED\_WITH\_ACORNS***(**"забитый желудями"**, **true**) {  
 @Override  
 **public** String getVoicePhrase() {  
 **return "не может говорить, так как его рот забит желудями"**;  
 }  
 };  
 **public final** String **name**;  
 **public final boolean isSilence**;  
 Voice(**final** String name,**final boolean** isSilence){  
 **this**.**name** = name;  
 **this**.**isSilence** = isSilence;  
 }  
 **public abstract** String getVoicePhrase();  
 **public** String getName(){  
 **return name**;  
 }  
}

// State.java

**package** com.company;  
  
**public enum** State {  
 ***STAY***(**null**,**null**),  
 ***ON\_WAY***(**"отправляется в путь"**,**"прибыл на место"**),  
 ***AT\_HOUSE***(**"теперь в доме"**,**"вышел из дома"**);  
 **private final** String **startPhrase**;  
 **private final** String **endPhrase**;  
 State(**final** String startPhrase,**final** String endPhrase){  
 **this**.**startPhrase** = startPhrase;  
 **this**.**endPhrase** = endPhrase;  
 }  
 **public** String startState(){  
 **return startPhrase**;  
 }  
 **public** String endState(){  
 **return endPhrase**;  
 }  
}

// Piglet.java

**package** com.company;  
  
**public class** Piglet **extends** Entity **implements** LittleCreature, HouseOwner{  
 Piglet(**final** String name,**final** State state){  
 **super**(name, state);  
 }  
 @Override  
 **public** String whoAmI() {  
 **return name**+**" - очень маленькое существо"**;  
 }  
 @Override  
 **public void** troubles(){  
 System.***out***.println(**name**+**" не любит, когда на него напрыгивают"**);  
 }  
 @Override  
 **public void** giveFood(FoodTaker foodTaker){  
 System.***out***.println(**name**+**" подвинул корзинку с желудями к гостю"**);  
 speak(**"Угощайтесь, пожалуйста"**);  
 foodTaker.takeFood();  
 }  
 @Override  
 **public void** openDoor(Guest[] guests){  
 System.***out***.println(**name**+**" открывает дверь"**);  
 **for**(Guest i: guests){  
 i.enterInHouse();  
 }  
 }  
 **public void** hug(Pooh pooh){  
 System.***out***.println(**name**+**" прижимается к "**+pooh);  
 setVoice(Voice.***ALMOST\_FUNNY***);  
 }  
}

// Pooh.java

**package** com.company;  
  
**public class** Pooh **extends** Entity **implements** Guest{  
 Pooh(**final** String name,**final** State state){  
 **super**(name,state);  
 }  
 @Override  
 **public void** enterInHouse(){  
 System.***out***.println(**name**+**" входит в дом"**);  
 }  
 **public void** tellAboutLittleCreature(Entity entity, LittleCreature littleCreature){  
 System.***out***.println(**name**+**" начинает рассказывать "**+ entity +**" о том, что "**+littleCreature.whoAmI());  
 littleCreature.troubles();  
 System.***out***.println(**name**+**" просит "**+entity+**" не наскакивать во время первого знакомства"**);  
 }  
 **public void** startJourneyWith(Entity entity, String place){  
 setState(State.***ON\_WAY***);  
 entity.setState(State.***ON\_WAY***);  
 System.***out***.println(**name**+**" и "**+ entity +**" после завтрака начали путешествие в "**+place);  
 }  
 **public void** endJourneyWith(Tigger tigger,Piglet piglet){  
 System.***out***.println(**name**+**" стучит в дверь"**);  
 Guest[] guests = {**this**, tigger};  
 piglet.openDoor(guests);  
 setState(State.***AT\_HOUSE***);  
 tigger.setState(State.***AT\_HOUSE***);  
 }  
}

// Tigger.java

**package** com.company;  
  
**public class** Tigger **extends** Entity **implements** FoodTaker, Guest{  
 Tigger(**final** String name,**final** State state){  
 **super**(name,state);  
 }  
 @Override  
 **public void** takeFood(){  
 System.***out***.println(**name**+**" берёт еду"**);  
 setVoice(Voice.***CLOGGED\_WITH\_ACORNS***);  
 }  
 @Override  
 **public void** enterInHouse(){  
 System.***out***.println(**name**+**" входит в дом, прекращая прыгать"**);  
 }  
 **public void** playWithShadow(Shadow shadow){  
 System.***out***.println(**name**+**" начинает прятаться за деревьями"**);  
 shadow.turnAway(**this**);  
 shadow.turnAway(**this**);  
 System.***out***.println(**name**+**" рассказывает, что Тигры наскакивают только до завтрака"**);  
 System.***out***.println(**"А как только они поедят желудей, то становятся Тихими и Вежливыми"**);  
 }  
 **public void** tryToCatchShadow(Shadow shadow){  
 System.***out***.println(**name** + **" пытается поймать "** + shadow);  
 shadow.setCaught(Math.*random*()<0.5);  
 }  
}

// Shadow.java

**package** com.company;  
  
**public class** Shadow **extends** Entity {  
 **private boolean isCaught** = **false**;  
 Shadow(**final** String name,**final** State state){  
 **super**(name,state);  
 }  
 **public void** turnAway(**final** Tigger tigger){  
 **if**(!**isCaught**) {  
 System.***out***.println(**name** + **" отвернулась от "** + tigger);  
 tigger.tryToCatchShadow(**this**);  
 }  
 }  
 **public void** setCaught(**boolean** caught) {  
 **isCaught** = caught;  
 **if**(**isCaught**){  
 System.***out***.println(**name**+**" была поймана!"**);  
 }  
 **else**{  
 System.***out***.println(**name**+**" не удалось поймать"**);  
 }  
 }  
}

Результат работы:

Результат 1.

Изображение выглядит как текст

Автоматически созданное описание

Вывод:

В ходе выполнения данной лабораторной работы я ознакомился с абстрактными классами и интерфейсами и научился работать с перечислениями в языке Java. Также я изучил принципы SOLID и STUPID, построил объектную модель, используя их. Полученные навыки пригодятся мне в дальнейшем.