САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ.

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №3 ПО «ПРОГРАММИРОВАНИЮ»**

Вариант №3136

**Выполнила:**

Соколова Анастасия

Ефимова Катерина

Группа Р3101

**Преподаватель:**

Цопа Евгений Алексеевич

САНКТ-ПЕТЕРБУРГ, 2018

**Текст задания:**

Прежде чем Иа успел ответить, раздался громкий треск, и из камышей появился Тигра собственной персоной. Кристофер Робин шел по лесной дорожке в таком солнечном и безоблачном настроении, как будто бы, например, дважды девятнадцать-- это пустяки, и думал о том, что, если он в такой день встанет на нижнюю перекладину перил моста и наклонится над рекой, он вдруг узнает все-все на свете и тогда он расскажет все это Пуху, который пока еще знает не все на свете. Но когда он подошел к мосту и увидел всех своих друзей, он понял, что сегодня совсем не такой день. а совершенно другой-- день, когда нужно что-то сделать.

**Программа должна удовлетворять следующим требованиям:**

1. Доработанная модель должна соответствовать [принципам SOLID](https://en.wikipedia.org/wiki/SOLID_(object-oriented_design)).
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).

**Порядок выполнения работы:**

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.

..

|  |
| --- |
| +Existance |
| String name |
| +Forest |
| Public void setPlace(String place) public String getPlace() |
| +Place enum |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| +Person |  |  |  |  |
| Void go(String place)  Void mood()  Boolean dreams() |  |  |  |  |
| +Person(String name) |  |  |  |  |
|  |  |  |  |  |
| +Animals |  | +Tiger |  |  |
| Public void ask(String)  Public void answer()  Public void move() |  | Public void ask(String)  Public void answer()  Public void move() |  |  |
| Animals(String name, Place area) |  | +Tiger(String name, Place area) |  |  |
|  |  |  |  |  |
| +Politoed |  | +Poliwhirl |  | +Poliwag |
|  |  |  |  |  |
| +Politoed(String, int)  +Speech  Void ask(String)  Void answer() |  | +Poliwhirl(String, int )  +Acting  Void move() |  | +Poliwag(String,  int) |

**Исходный код**

public class Main {

public static void main(String[] args) {

Animals pooh = new Animals("Винни", Place.GLADE);

Animals ia = new Animals("Иа", Place.GLADE);

Tiger Tiger = new Tiger("Тигруля", Place.REEDS);

pooh.ask(ia.name);

ia.answer();

Tiger.move();

Person Chris = new Person("Кристофер Робин");

Chris.go("Дорожка");

Chris.mood();

Chris.dreams();

Chris.go("Мост");

Chris.dreams();

}

}

abstract class Existance {

String name;

@Override

public boolean equals(Object object) {

return this == object;

}

@Override

public String toString() {

return "Existance [x = " + name + "]";

}

@Override

public int hashCode() {

return super.hashCode();

}

public class Person extends Existance {

Forest field = new Forest();

Person(String name) {

super.name = name;

}

void go(String place) {

field.setPlace(place);

System.out.println(name + " пошагал на место: " + field.getPlace());

}

void mood() {

System.out.println("настроение : солнечное и безоблачное");

}

boolean dreams() {

if (field.getPlace().equals("Дорожка")) {

System.out.print("Если я встану на нижнюю перекладину перил моста и наклонюсь над рекой, ");

System.out.print("я узнаю все-все на свете и тогда я расскажу все это Пуху, который пока ");

System.out.println("еще знает не все на свете. - подумал " + name);

return true;

} else if (field.getPlace().equals("Мост") && (Animals.friends == true)) {

System.out.print("Сегодня день, когда нужно что-то сделать.");

System.out.println(" - подумал " + name + ", увидев своих друзей.");

return true;

} else {

System.out.println("никаких планов на день. - подумал " + name);

return false;

}

}

@Override

public boolean equals(Object object) {

return this == object;

}

@Override

public String toString() {

return "Person [x = " + name + "]";

}

@Override

public int hashCode() {

return super.hashCode();

}

}

public class Animals extends Existance implements Speech, Acting {

Place point;

public String type;

static boolean friends = false;

Animals() {

}

Animals(String name, Place area) {

super.name = name;

this.point = area;

switch (this.point) {

case GLADE:

this.type = "Поле";

this.friends = true;

break;

case REEDS:

this.type = "Камыши";

break;

case PATH:

this.type = "Дорожка";

break;

case BRIDGE:

this.type = "Мост";

break;

}

}

public void ask(String a) {

System.out.println(name + " спрашивает " + a);

}

public void answer() {

System.out.println(name + " отвечает");

}

public void move() {

System.out.println(name + "находится на месте: " + point);

}

@Override

public boolean equals(Object object) {

return this == object;

}

@Override

public String toString() {

return "Animals [x = " + name + "]";

}

@Override

public int hashCode() {

return super.hashCode();

}

}public class Tiger extends Animals implements Acting {

public Tiger(String name, Place area) {

super(name, area);

}

public void move() {

System.out.println(name + " появился из места : " + type);

}

public void ask(String a) {

}

public void answer() {

}

}public class Forest {

private String place;

public void setPlace(String place) {

this.place = place;

}

public String getPlace() {

return place;

}

@Override

public boolean equals(Object object) {

return this == object;

}

@Override

public String toString() {

return "Forest [x = " + place + "]";

}

@Override

public int hashCode() {

return super.hashCode();

}

}

public enum Place {

GLADE,

REEDS,

PATH,

BRIDGE;

}

public interface Speech {

void ask(String a);

void answer();

}

public interface Acting {

void move();

}

**Результат работы программы:**

Винни спрашивает Иа

Иа отвечает

Тигруля появился из места : Камыши

Кристофер Робин пошагал на место: Дорожка

настроение : солнечное и безоблачное

Если я встану на нижнюю перекладину перил моста и наклонюсь над рекой, я узнаю все-все на свете и тогда я расскажу все это Пуху, который пока еще знает не все на свете. - подумал Кристофер Робин

Кристофер Робин пошагал на место: Мост

Сегодня день, когда нужно что-то сделать. - подумал Кристофер Робин, увидев своих друзей.

**Вывод:**