Факультет программной инженерии и компьютерной техники

Программирование

Лабораторная работа № 3

Вариант 126589

Выполнил Калабухов Максим

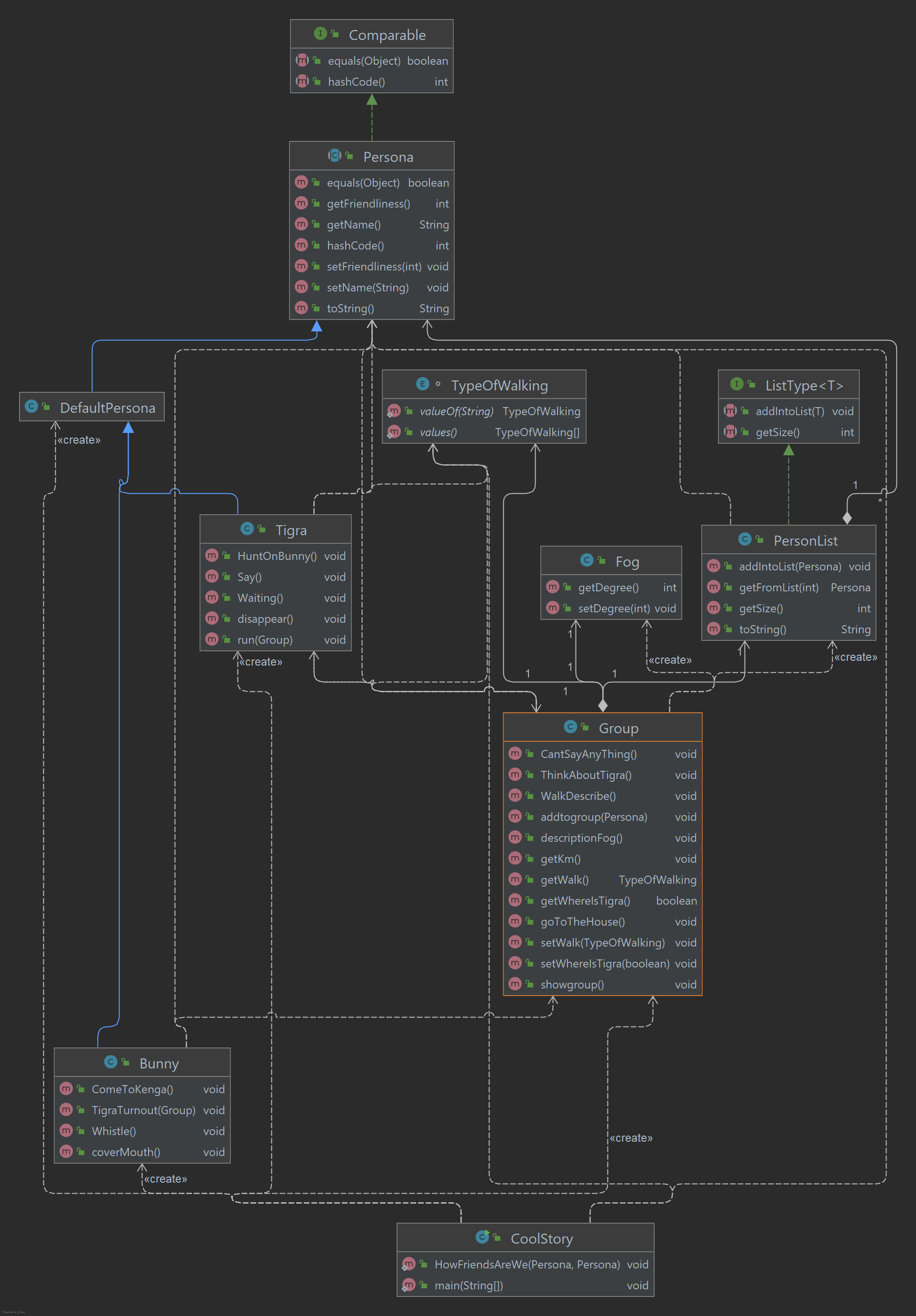
Группа P3112

Г.Санкт-Петербург

2021 г.

Изображение выглядит как текст

Автоматически созданное описание



Исходный код:

CoolStory.java

**public** **class** **CoolStory** {

**public** **static** **void** **main**(String[] args) {

Tigra Tigra = **new** Tigra("Тигра", **1**);

Bunny bunny = **new** Bunny("Кролик", **2**);

DefaultPersona vinni = **new** DefaultPersona("Винни-пух", **3**);

DefaultPersona ru = **new** DefaultPersona("Крошка Ру", **3**);

DefaultPersona pig = **new** DefaultPersona("Пятачок", **4**);

Group friends1 = **new** Group();

Group tigran = **new** Group();

tigran.addtogroup(Tigra);

tigran.addtogroup(ru);

friends1.addtogroup(pig);

friends1.addtogroup(bunny);

friends1.addtogroup(vinni);

tigran.goToTheHouse();

HowFriendsAreWe(Tigra,ru);

friends1.goToTheHouse();

Tigra.Waiting();

bunny.TigraTurnout(tigran);

friends1.setWalk(TypeOfWalking.WalkingAlongside);

Tigra.run(friends1);

friends1.showgroup();

friends1.getKm();

friends1.setWalk(TypeOfWalking.inSingleFile);

friends1.descriptionFog();

Tigra.run(friends1);

friends1.showgroup();

friends1.setWalk(TypeOfWalking.onBothSides);

Tigra.run(friends1);

friends1.showgroup();

friends1.descriptionFog();

Tigra.disappear();

Tigra.HuntOnBunny();

Tigra.HuntOnBunny();

Tigra.HuntOnBunny();

friends1.getKm();

friends1.ThinkAboutTigra();

Tigra.Say();

friends1.CantSayAnyThing();

Tigra.disappear();

}

**public** **static** **void** **HowFriendsAreWe**(Persona a, Persona b) {

**if** ((a.getName().equals("Тигра") && b.getName().equals("Крошка Ру")) || (b.getName().equals("Тигра") && a.getName().equals("Крошка Ру"))) {

System.out.println(a.getName() + " и " + b.getName() + " близкие друзья");

}

**else** **if** ((a.getFriendliness() + b.getFriendliness() < **10**) && (a.getFriendliness() + b.getFriendliness() > **5**)) {

System.out.println(a.getName() + " и " + b.getName() + " хорошие друзья");

} **else** **if** ((a.getFriendliness() + b.getFriendliness() > **0**) && (a.getFriendliness() + b.getFriendliness() <= **5**)) {

System.out.println(a.getName() + " и " + b.getName() + " знакомые");

}

}

}

Persona.java

**import** **java.util.Objects**;

**public** **abstract** **class** **Persona** **implements** Comparable {

**private** String name;

**private** **int** friendliness;

**public** **Persona**(String name, **int** friendliness) {

**this**.name = name;

**this**.friendliness = friendliness;

System.out.println(getName() + ": присоеденился к истории");

}

**public** **int** **getFriendliness**() {

**return** friendliness;

}

**public** **void** **setFriendliness**(**int** friendliness) {

**this**.friendliness = friendliness;

}

**public** String **getName**() {

**return** name;

}

**public** **void** **setName**(String name) {

**this**.name = name;

}

@Override

**public** **boolean** **equals**(Object o) {

**if** (**this** == o) **return** **true**;

**if** (o == **null** || getClass() != o.getClass()) **return** **false**;

Persona persona = (Persona) o;

**return** friendliness == persona.friendliness && Objects.equals(name, persona.name);

}

@Override

**public** **int** **hashCode**() {

**return** Objects.hash(name, friendliness);

}

@Override

**public** String **toString**() {

**return** "Persona{" +

"name='" + name + '\'' +

", friendliness=" + friendliness +

'}';

}

}

DefaultPersona.java

**public** **class** **DefaultPersona** **extends** Persona {

**public** **DefaultPersona**(String name, **int** friendless) {

**super**(name,friendless);

}

**public** **void** **go**() {

System.out.println(getName() + "идёт");

}

}

Tigra.java

**public** **class** **Tigra** **extends** DefaultPersona{

**private** **int** howmanytimesdisappear = **0**;

**public** **Tigra**(String name, **int** friendless) {

**super**(name,friendless);

}

**public** **void** **Waiting**() {

System.out.println( getName() + " их ждёт");

}

**public** **void** **run**(Group p) {

**if** (p.getWalk().equals(TypeOfWalking.WalkingAlongside)) {

System.out.print(getName() + " носится и описывает большие круги вокруг ");

}

**else** **if** (p.getWalk().equals(TypeOfWalking.inSingleFile)) {

System.out.print(getName() + " описывает петли вокруг ");

}

**else** **if** (p.getWalk().equals(TypeOfWalking.onBothSides)) {

System.out.print(getName() + " убегает то далеко вперед, то возвращается к группе ");

}

}

**public** **void** **HuntOnBunny**() {

**double** chance = Math.random() \* **100**;

**if** (chance >= Math.random() \* **100**) {

System.out.print("\n");

System.out.format("Желание Тигры напасть на Кролика := %.2f" , chance );

System.out.println("%");

System.out.println("Тигра нападает на Кролика");

}

**else** {

System.out.print("\n");

System.out.format("Желание Тигры напасть на Кролика := %.2f", chance );

System.out.println("%");

System.out.println("Тигра не нападает на Кролика");

}

}

**public** **void** **Say**() {

System.out.println(getName() + " появляется и выпаливает:\"Чего же вы ? Ходу !\"");

}

**public** **void** **disappear**() {

**this**.howmanytimesdisappear += **1**;

**if** (howmanytimesdisappear == **1**) {

System.out.println(getName() + " исчез в тумане");

}

**else** **if** (howmanytimesdisappear > **1**) {

System.out.println(getName() + " снова исчез");

}

}

}

Comparable.java

**public** **interface** **Comparable** {

**boolean** **equals**(Object o);

**int** **hashCode**();

}

Group.java

**public** **class** **Group**{

Fog fog = **new** Fog();

PersonList group = **new** PersonList();

**private** **boolean** whereIsTigra = **false**;

**private** TypeOfWalking walk = TypeOfWalking.DefaultWalk;

**private** **int** km;

**public** **void** **showgroup**() {

**for** (**int** i = **0**; i < group.getSize(); i++) {

**if** (i == (group.getSize() - **1**)) {

System.out.print(group.getFromList(i).getName() + "a ");

} **else** {

System.out.print(group.getFromList(i).getName() + "a, ");

}

}

}

**public** **void** **addtogroup**(Persona a) {

group.addIntoList(a);

**if** (a.getClass().equals(Tigra.class)) {

setWhereIsTigra(**true**);

}

}

**public** **void** **goToTheHouse**() {

**for** (**int** i = **0**; i < group.getSize(); i++) {

**if** (i == (group.getSize() - **1**)) {

System.out.print(group.getFromList(i).getName() + " ");

km += **1**;

} **else** {

System.out.print(group.getFromList(i).getName() + ", ");

km += **1**;

}

}

System.out.println("идут к дому Кенги");

}

**public** **boolean** **getWhereIsTigra**() {

**return** whereIsTigra;

}

**public** **void** **setWhereIsTigra**(**boolean** whereIsTigra) {

**this**.whereIsTigra = whereIsTigra;

}

**public** TypeOfWalking **getWalk**() {

**return** walk;

}

**public** **void** **setWalk**(TypeOfWalking walk) {

**this**.walk = walk;

System.out.println("");

System.out.print("Теперь группа из ");

showgroup();

WalkDescribe();

}

**public** **void** **WalkDescribe**() {

**if** (walk.equals(TypeOfWalking.DefaultWalk)){

System.out.println("идут как обычно");

km += **1**;

fog.setDegree(**1**);

}

**else** **if** (walk.equals(TypeOfWalking.WalkingAlongside)){

System.out.println("идут рядом");

km += **1**;

fog.setDegree(**3**);

}

**else** **if** (walk.equals(TypeOfWalking.inSingleFile)){

System.out.println("идут гуськом, друг за другом");

km += **1**;

fog.setDegree(**5**);

}

**else** **if** (walk.equals(TypeOfWalking.onBothSides)){

System.out.println("видит что по обе стороны тропинки встала колючая стена чертополоха");

km += **1**;

fog.setDegree(**10**);

}

}

**public** **void** **getKm**() {

System.out.println("");

System.out.print("Группа ");

showgroup();

System.out.println("прошла " + km + " киллометров");

}

**public** **void** **ThinkAboutTigra**() {

**if** (fog.getDegree() == **10**) {

System.out.print("\*Группа из ");

showgroup();

System.out.println(" думает, что Тигра исчез\*");

}

}

**public** **void** **CantSayAnyThing**(){

System.out.print("Группа из ");

showgroup();

System.out.println(" не успевает ничего ему сказать");

}

**public** **void** **descriptionFog**(){

System.out.println("");

**if** (fog.getDegree() == **1**) {

System.out.println("Туман не наблюдается");

}

**else** **if** (fog.getDegree() == **3**) {

System.out.println("В воздухе можно было заметить нотки слабого тумана");

}

**else** **if** (fog.getDegree() == **5**) {

System.out.println("Туман начинал бросаться в глаза, но все попрежнему было видно");

}

**else** **if** (fog.getDegree() == **7**) {

System.out.println("Туман стоял густой, не всегда можно было разглядеть дальнее дерево, но все же было комфортно");

}

**else** **if** (fog.getDegree() == **10**) {

System.out.println("Туман был настолько густ, что Тигра стал временами пропадать");

}

}

}

Bunny.java

**public** **class** **Bunny** **extends** DefaultPersona{

**public** **Bunny**(String name, **int** friendless) {

**super**(name, friendless);

}

**public** **void** **ComeToKenga**(){

System.out.println(" и подошел к Кенге");

}

**public** **void** **TigraTurnout**(Group a){

**if** (a.getWhereIsTigra()){

System.out.print(getName() + " Увидев Тигру");

coverMouth();

Whistle();

ComeToKenga();

}

}

**public** **void** **Whistle**(){

System.out.print(" сказал:\"Положитесь на меня\"");

}

**public** **void** **coverMouth**(){

System.out.print(", прикрыл рот лапкой и");

}

}

fog.java

**public** **class** **Fog** {

**private** **int** degree;

**public** **int** **getDegree**() {

**return** degree;

}

**public** **void** **setDegree**(**int** degree) {

**this**.degree = degree;

}

}

ListType.java

**public** **interface** **ListType**<T>{

**void** **addIntoList**(T a);

**int** **getSize**();

}

PersonList.java

**import** **java.util.ArrayList**;

**public** **class** **PersonList** **implements** ListType<Persona> {

**private** ArrayList<Persona> personList;

**public** **PersonList**() {

personList = **new** ArrayList<Persona>();

}

@Override

**public** **void** **addIntoList**(Persona a) {

personList.add(a);

}

@Override

**public** **int** **getSize**() {

**return** personList.size();

}

**public** Persona **getFromList**(**int** i) {

**return** personList.get(i);

}

@Override

**public** String **toString**() {

**return** "PersonList{" +

"personList=" + personList +

'}';

}

}

TypeOfWalking.java

**enum** TypeOfWalking {

DefaultWalk,

WalkingAlongside, //идти рядом

inSingleFile, //гуськом

onBothSides; //по обе стороны встала колючая стена чертополоха

}

@Override

**public** **void** **addIntoList**(Persona a) {

personList.add(a);

}

@Override

**public** **int** **getSize**() {

**return** personList.size();

}

**public** Persona **getFromList**(**int** i) {

**return** personList.get(i);

}

@Override

**public** String **toString**() {

**return** "PersonList{" +

"personList=" + personList +

'}';

}

}

Результат работы:

Тигра: присоеденился к истории

Кролик: присоеденился к истории

Винни-пух: присоеденился к истории

Крошка Ру: присоеденился к истории

Пятачок: присоеденился к истории

Тигра, Крошка Ру идут к дому Кенги

Тигра и Крошка Ру близкие друзья

Пятачок, Кролик, Винни-пух идут к дому Кенги

Тигра их ждёт

Кролик Увидев Тигру, прикрыл рот лапкой и сказал:"Положитесь на меня" и подошел к Кенге

Теперь группа из Пятачокa, Кроликa, Винни-пухa идут рядом

Тигра носится и описывает большие круги вокруг Пятачокa, Кроликa, Винни-пухa

Группа Пятачокa, Кроликa, Винни-пухa прошла 4 киллометров

Теперь группа из Пятачокa, Кроликa, Винни-пухa идут гуськом, друг за другом

Туман начинал бросаться в глаза, но все попрежнему было видно

Тигра описывает петли вокруг Пятачокa, Кроликa, Винни-пухa

Теперь группа из Пятачокa, Кроликa, Винни-пухa видит что по обе стороны тропинки встала колючая стена чертополоха

Тигра убегает то далеко вперед, то возвращается к группе Пятачокa, Кроликa, Винни-пухa

Туман был настолько густ, что Тигра стал временами пропадать

Тигра исчез в тумане

Желание Тигры напасть на Кролика := 20,28%

Тигра не нападает на Кролика

Желание Тигры напасть на Кролика := 76,91%

Тигра нападает на Кролика

Желание Тигры напасть на Кролика := 2,44%

Тигра не нападает на Кролика

Группа Пятачокa, Кроликa, Винни-пухa прошла 6 киллометров

\*Группа из Пятачокa, Кроликa, Винни-пухa думает, что Тигра исчез\*

Тигра появляется и выпаливает:"Чего же вы ? Ходу !"

Группа из Пятачокa, Кроликa, Винни-пухa не успевает ничего ему сказать

Тигра снова исчез

Вывод:

Во время выполнения данной лабораторной работы я научился применять принципы SOLID на практике.