Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Алтайский государственный технический университет

им. И.И. Ползунова»

Факультет (институт) Информационных технологий

Кафедра Прикладная математик

Отчет защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.М. Старолетов

(подпись преподавателя) (инициалы, фамилия)

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2017 г.

Отчет

по лабораторной (практической) работе № 5

по дисциплине Архитектурное проектирование и паттерны программирования

(наименование дисциплины)

ЛР 09.03.04.10.000 ОТ

(обозначение документа)

Студенты группы ПИ-42 Е.А. Колесников

(инициалы, фамилии)

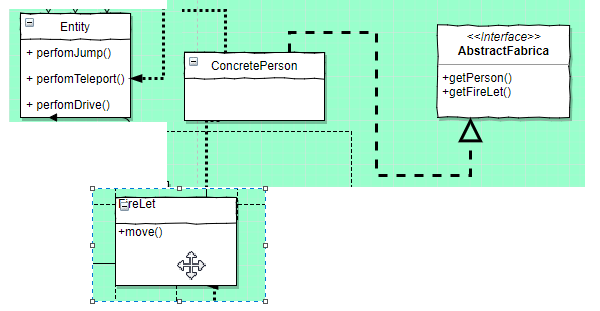
Преподаватель Доцент к. канд. физ.-мат. наук С.М.Старолетов

(должность, ученое звание) (инициалы, фамилия)

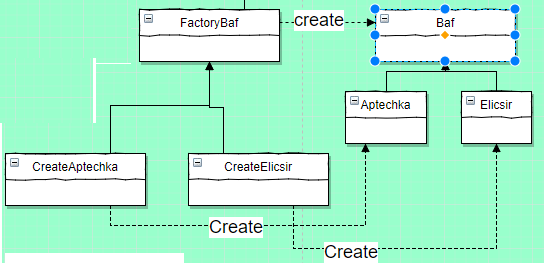
Барнаул 2017

**Задание:** реализовать паттерны: Фабрика, Абстрактная фабрика, Фабричный метод

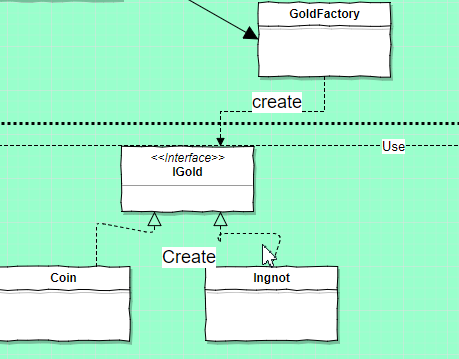
Абстрактная фабрика:



Фабричный метод:



Фабрика:



**Код:**

**Фабричный метод:**

public abstract class FabrikaBaf {  
 public abstract Baf CreateBaf(int baf);  
}

public abstract class Baf {  
 protected int baf;  
 protected Texture texturebaf;  
 protected Rectangle rectanglebaf;  
 public Baf(int baf,int x, int y) {  
 rectanglebaf=new Rectangle();  
 rectanglebaf.x=x;  
 rectanglebaf.y=y;  
 rectanglebaf.width=64;  
 rectanglebaf.height=64;  
 this.baf = baf;  
 }  
  
 public Texture getTexturebaf() {  
 return texturebaf;  
 }  
  
 public void setTexturebaf(Texture texturebaf) {  
 this.texturebaf = texturebaf;  
 }  
  
 public Rectangle getRectanglebaf() {  
 return rectanglebaf;  
 }  
  
 public void setRectanglebaf(Rectangle rectanglebaf) {  
 this.rectanglebaf = rectanglebaf;  
 }  
  
 public Baf() {  
 baf=0;  
 }  
  
 public abstract void toApply(Person p);  
}

public class CreateAptechka extends FabrikaBaf {  
 public CreateAptechka() {  
 }  
  
 @Override  
 public Baf CreateBaf(int baf) {  
 int x= MathUtils.*random*(0, GameScreen.*VIEWPORT\_WIDTH*);  
 Aptechka aptechka = new Aptechka(baf,x,20);  
 return aptechka;  
 }  
}

public class CreateEliksir extends FabrikaBaf {  
 public CreateEliksir() {  
  
 }  
  
 @Override  
 public Baf CreateBaf(int baf) {  
 int x= MathUtils.*random*(0, GameScreen.*VIEWPORT\_WIDTH*);  
 Eliksir eliksir = new Eliksir(baf,x,20);  
 return eliksir;  
 }  
}

public class Aptechka extends Baf {  
 public Aptechka(int baf, int x, int y) {  
 super(baf, x, y);  
 texturebaf= new Texture(Gdx.*files*.internal("aptech.png"));  
 }  
  
 public Aptechka() {  
 }  
  
 @Override  
 public void toApply(Person p) {  
 p.setHP(p.getHP()+baf);  
 }  
}

public class Eliksir extends Baf {  
  
 @Override  
 public void toApply(Person p) {  
 Vector3 vector3;  
 vector3=p.getVelocity();  
 vector3.x\*=baf;  
 p.setVelocity(vector3);  
 System.*out*.println("Ускорение: "+vector3.x);  
 }  
  
 public Eliksir(int baf, int x, int y) {  
 super(baf, x, y);  
 texturebaf= new Texture(Gdx.*files*.internal("vodka.png"));  
 }  
  
 public Eliksir() {  
 }  
}

**Абстрактная фабрика:**

public interface AbstractPersonazhi {  
 Person getPerson(String path);  
 FireLet getFireLet();  
}

public class ConcretePerson implements AbstractPersonazhi {  
 @Override  
 public Person getPerson(String path) {  
 DirectorPerson directorPerson = new DirectorPerson();  
 if(path=="player2")  
 directorPerson.setPersonBuilder(new BuilderPlayer2());  
 else  
 directorPerson.setPersonBuilder(new BuilderPlayer1());  
  
 directorPerson.buildPerson();  
 return directorPerson.getPlayer();  
 }  
  
 @Override  
 public FireLet getFireLet() {  
 return new FireLet();  
 }  
}

**Фабрика:**

public interface IGold {  
 int getMoney();  
 int setPositionX();  
 Texture getTexture();  
 int getPositionX();  
 int setPositionY();  
 int getPositionY();  
}

public class GoldFactory {  
 public static IGold createGold() {  
 switch (MathUtils.*random*(0,1)) {  
 case 0:  
 return new Coin();  
 case 1:  
 return new Ingot();  
 default:return null;  
 }  
 }  
}

public class GoldInc {  
 private Array<IGold> gold;  
  
 public Array<IGold> getGold() {  
 return gold;  
 }  
  
 public void setGold(Array<IGold> gold) {  
 this.gold = gold;  
 }  
  
 public GoldInc() {  
 gold=new Array<IGold>();  
 }  
  
 public void createGold()  
 {  
 gold.add(GoldFactory.*createGold*());  
 }  
 public void SendToClient(IGold gold)  
 {  
  
 System.*out*.println("----------------------------");  
 System.*out*.println("Денег= "+gold.getMoney());  
 System.*out*.println("Х= "+gold.getPositionX());  
 System.*out*.println("Y= "+gold.getPositionY());  
 System.*out*.println("----------------------------");  
 }  
 public void update(Person e)  
 {  
 for(int i=0;i<gold.size;i++) {  
 if (e.getPlayer().overlaps(new Rectangle(gold.get(i).getPositionX(), gold.get(i).getPositionY(), 64, 64)))  
 {  
 e.setGold(e.getGold()+gold.get(i).getMoney());  
 gold.removeIndex(i);  
 }  
 }  
 }  
}

public class Coin implements IGold {  
 private static final int *money*=10;  
 private int positionX;  
 private int positionY;  
 private Texture coin;  
  
 @Override  
 public Texture getTexture() {  
 return coin;  
 }  
  
 @Override  
 public int getMoney() {  
 return *money*;  
 }  
  
 public Coin() {  
 coin = new Texture(Gdx.*files*.internal("gold.png"));  
 this.positionX = MathUtils.*random*(GameScreen.*STARTER\_POSITION*, GameScreen.*VIEWPORT\_WIDTH* - GameScreen.*WIDTH\_PLAYER*);  
 this.positionY = GameScreen.*STARTER\_POSITION*;  
 }  
  
 public Coin(int positionX, int positionY) {  
 this.positionX = positionX;  
 this.positionY = positionY;  
 }  
  
 @Override  
 public int setPositionX() {  
 return 0;  
 }  
  
 @Override  
 public int getPositionX() {  
 return positionX;  
 }  
  
 @Override  
 public int setPositionY() {  
 return 0;  
 }  
  
 @Override  
 public int getPositionY() {  
 return positionY;  
 }  
}

public class Ingot implements IGold {  
 private static final int *money*=50;  
 private Texture coin;  
 private int positionX;  
 private int positionY;  
  
 public Ingot() {  
 coin = new Texture(Gdx.*files*.internal("kuchaGold.png"));  
 this.positionX = MathUtils.*random*(GameScreen.*STARTER\_POSITION*, GameScreen.*VIEWPORT\_WIDTH* - GameScreen.*WIDTH\_PLAYER*);  
 this.positionY = GameScreen.*STARTER\_POSITION*;  
 }  
  
 @Override  
 public Texture getTexture() {  
 return coin;  
 }  
  
 public Ingot(int positionX, int positionY) {  
 this.positionX = positionX;  
 this.positionY = positionY;  
  
 }  
  
 @Override  
 public int getMoney() {  
  
 return 0;  
 }  
  
 @Override  
 public int setPositionX() {  
 return 0;  
 }  
  
 @Override  
 public int getPositionX() {  
 return positionX;  
 }  
  
 @Override  
 public int setPositionY() {  
 return 0;  
 }  
  
 @Override  
 public int getPositionY() {  
 return positionY;  
 }  
}

**ТЕСТЫ:**

**Абстрактная фабрика:**



**Фабричный метод и фабрика:**

