

Факультет программной инженерии и

компьютерной техники

Лабораторная работа №3

**Основы объектно-ориентированного программирования**

Вариант №34082

Выполнила Громилова Мария Дмитриевна,

группа P3130

Преподаватель Блохина Елена Николаевна

г. Санкт-Петербург 2021 г.

**Оглавление:**

[Задание к лабораторной работе: 2](#_Toc90930106)

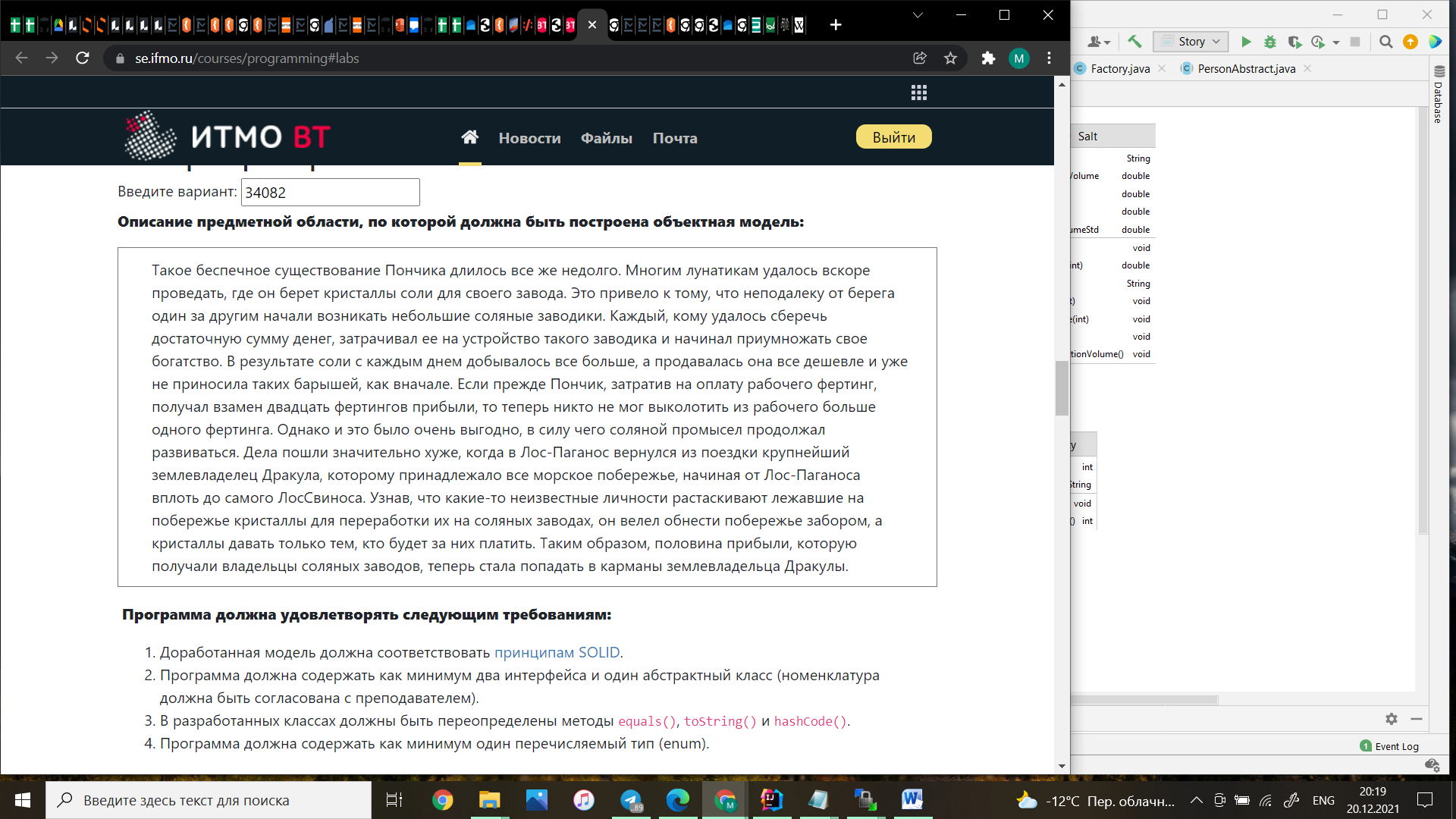
[Диаграмма классов реализованной объектной модели: 3](#_Toc90930107)

[Исходный код: 3](#_Toc90930108)

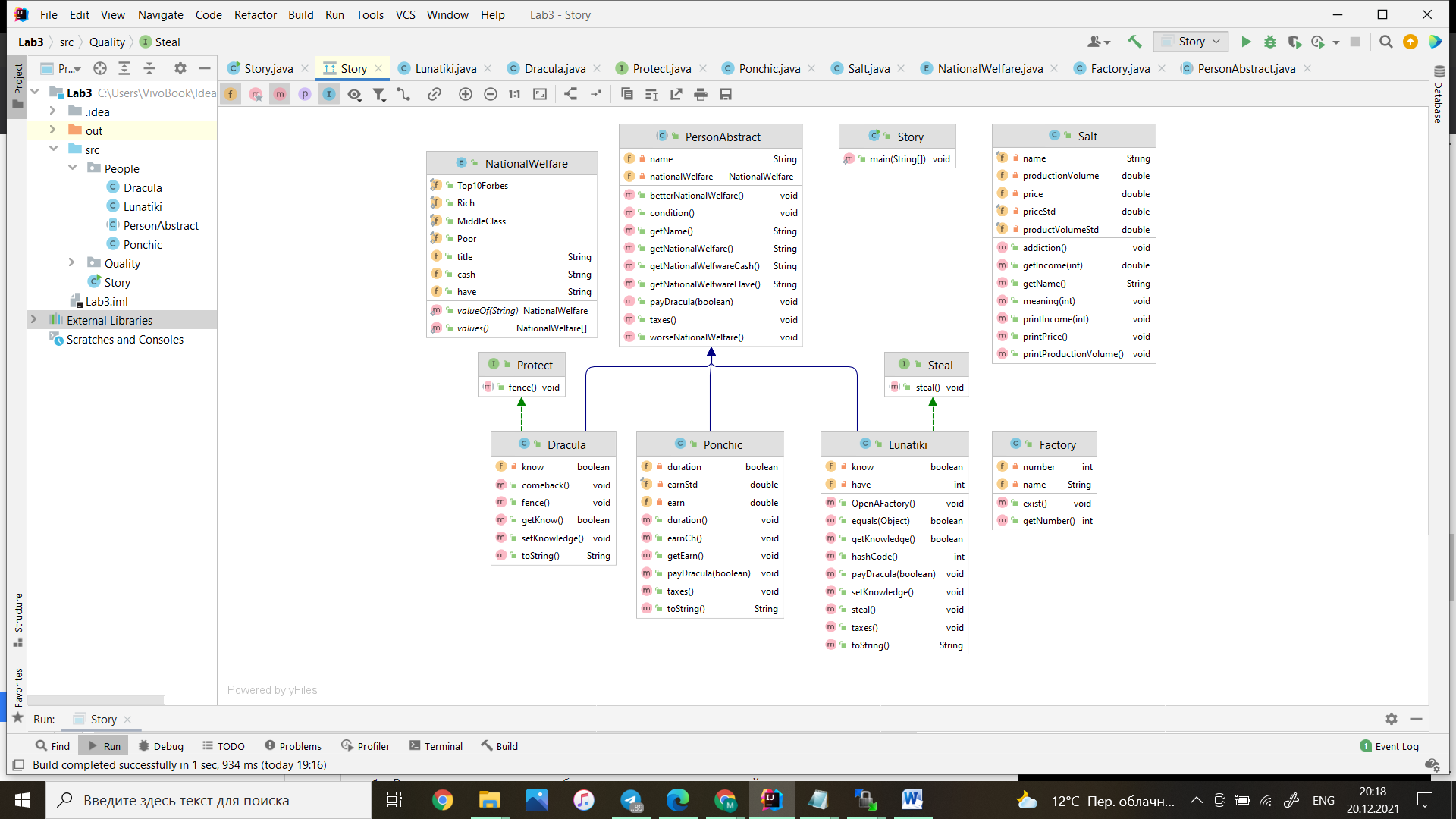
[Результаты работы программы: 10](#_Toc90930109)

[Вывод: 11](#_Toc90930110)

# Задание к лабораторной работе:



# Диаграмма классов реализованной объектной модели:



# Исходный код:

**Story.java**

**import** People.\*;  
**import** Quality.\*;  
  
**public class** Story {  
 **public static void** main(String[] args) {  
 Ponchic ponchic = **new** Ponchic(**"Пончик"**, NationalWelfare.***Top10Forbes***, **false**);  
 Dracula dracula = **new** Dracula(**"Дракула"**, NationalWelfare.***Rich***, **false**);  
 Lunatiki lunatik1 = **new** Lunatiki(**"Первый лунатик"**, NationalWelfare.***Poor***);  
 Lunatiki lunatik2 = **new** Lunatiki(**"Второй лунатик"**, NationalWelfare.***MiddleClass***);  
 Lunatiki lunatik3 = **new** Lunatiki(**"Третий лунатик"**, NationalWelfare.***MiddleClass***);  
 Lunatiki lunatik4 = **new** Lunatiki(**"Четвертый лунатик"**, NationalWelfare.***MiddleClass***);  
 Lunatiki lunatikTest = **new** Lunatiki(**"Лунатики"**, NationalWelfare.***MiddleClass***);  
 Salt salt = **new** Salt(**"Соль"**);  
 Factory factory = **new** Factory(**"Завод"**, 2);  
 ponchic.condition();  
 ponchic.duration();  
  
 **if** (ponchic.getNationalWelfare().equals(NationalWelfare.***Top10Forbes***.**title**)) {  
 lunatikTest.setKnowledge();  
 lunatik1.setKnowledge();  
 lunatik2.setKnowledge();  
 lunatik3.setKnowledge();  
 }  
  
 **if** (lunatikTest.equals(lunatik1)) {  
 lunatik1.OpenAFactory();  
 factory.exist();  
 lunatik1.betterNationalWelfare();  
 salt.addiction();  
 ponchic.earnCh();  
 }  
 **if** (lunatikTest.equals(lunatik2)){  
 lunatik2.OpenAFactory();  
 factory.exist();  
 lunatik2.betterNationalWelfare();  
 salt.addiction();  
 ponchic.earnCh();  
 }  
 **if** (lunatikTest.equals(lunatik3)){  
 lunatik3.OpenAFactory();  
 factory.exist();  
 lunatik3.betterNationalWelfare();  
 salt.addiction();  
 ponchic.earnCh();  
 }  
 **if** (lunatikTest.equals(lunatik4)){  
 lunatik4.OpenAFactory();  
 factory.exist();  
 lunatik4.getNationalWelfwareCash();  
 salt.addiction();  
 ponchic.earnCh();  
 }  
 salt.printProductionVolume();  
 salt.printPrice();  
 salt.printIncome(factory.getNumber());  
 ponchic.getEarn();  
 salt.meaning(factory.getNumber());  
 dracula.comeback();  
 ponchic.worseNationalWelfare();  
 dracula.setKnowledge();  
 lunatik1.steal();  
 lunatik2.steal();  
 lunatik3.steal();  
 lunatik4.steal();  
 dracula.fence();  
  
 ponchic.payDracula(dracula.getKnow());  
 lunatik1.payDracula(dracula.getKnow());  
 lunatik2.payDracula(dracula.getKnow());  
 lunatik3.payDracula(dracula.getKnow());  
 lunatik4.payDracula(dracula.getKnow());  
  
 ponchic.taxes();  
 lunatik1.taxes();  
 lunatik2.taxes();  
 lunatik3.taxes();  
 lunatik4.taxes();  
  
 }  
}

**Factory.java**

**package** Quality;  
  
**public class** Factory {  
 **private int number**;  
 **private** String **name**;  
 **public** Factory(String name, **int** number){  
 **this**.**name** = name;  
 **this**.**number** = number;  
 }  
 **public void** exist(){  
 **number** = **number** += 1;  
 System.***out***.println(**"Появляется небольшой соляной заводик."**);  
  
 }  
 **public int** getNumber(){  
 **return number**;  
 }  
}

**NationalWeifware.java**

**package** Quality;  
  
**public enum** NationalWelfare{  
 ***Top10Forbes***(**"очень богат"**, **"огромная сумма"**, **"владеет всем"** ),  
 ***Rich***(**"богат"**, **"большая сумма"**, **"владеет всем побережьем"**),  
 ***MiddleClass***(**"средний класс"**, **"достаточная сумма"**, **"владеет несколькими заводами"**),  
 ***Poor***(**"бедный"**, **"маленькая сумма"**, **"ничем не владеет"**);  
  
 **public** String **title**;  
 **public** String **cash**;  
 **public** String **have**;  
  
 NationalWelfare(String title, String cash, String have){  
 **this**.**title** = title;  
 **this**.**cash** = cash;  
 **this**.**have** = have;  
 }  
}

**Salt.java**

**package** Quality;  
  
**public class** Salt {  
 **private final** String **name**;  
 **private double productionVolume** = 28900;  
 **private double price** = 785;  
 **private final double priceStd** = 785;  
 **private final double productVolumeStd** = 28900;  
  
 **public** Salt(String name){  
 **this**.**name** = name;  
 }  
  
 **public void** addiction(){  
 **this**.**productionVolume** = **productionVolume** \* 1.2;  
 **this**.**price** = **price** \* 0.75;  
 }  
  
 **public void** printIncome(**int** num){  
 **double** income = **price** \* **productionVolume** / num;  
 **if** (income > **priceStd** \* **productVolumeStd**){  
 System.***out***.println(getName() + **" приносит больше барышей, чем вначале."**);  
 } **else** {  
 System.***out***.println(getName() + **" уже не приносит таких барышей как вначале."**);  
 }  
 }  
  
 **public void** meaning(**int** num) {  
 **int** min = 100000;  
 **if** (**price**\***productionVolume**/ num > min){  
 System.***out***.println(**"Все еще выгодно, солевой промысел развивается"**);  
 } **else** {  
 System.***out***.println(**"Не выгодно, сворачиваемся"**);  
 }  
 }  
  
 **public void** printPrice() {  
 **if** (**price** > **priceStd**) {  
 System.***out***.println(getName() + **" продается все дороже."**);  
 } **else** {  
 System.***out***.println(getName() + **" продается все дешевле."**);  
 }  
 }  
  
 **public void** printProductionVolume() {  
 **if** (**productionVolume** > **productVolumeStd**) {  
 System.***out***.println(getName() + **" добывается все больше."**);  
 } **else** {  
 System.***out***.println(getName() + **"Соли добывается все меньше."**);  
 }  
 }  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public double** getIncome(**int** num){  
 **double** income = (**price** \* **productionVolume**) / num;  
 **return** income;  
 }  
}

**Protect.java**

**package** Quality;  
  
**public interface** Protect {  
 **void** fence();  
}

**Quality.java**

**package** Quality;  
  
**public interface** Steal {  
 **void** steal();  
  
}

**PersonAbstract.java**

**package** People;  
  
**import** Quality.NationalWelfare;  
  
**public abstract class** PersonAbstract {  
 **private** String **name**;  
 **private** NationalWelfare **nationalWelfare**;  
  
 **public** PersonAbstract(String name, NationalWelfare nationalWelfare) {  
 **this**.**name** = name;  
 **this**.**nationalWelfare** = nationalWelfare;  
 }  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public** String getNationalWelfare() {  
 String title = **nationalWelfare**.**title**;  
 **return** title;  
 }  
 **public** String getNationalWelfwareCash(){  
 **return nationalWelfare**.**cash**;  
 }  
 **public** String getNationalWelfwareHave(){  
 String have = **nationalWelfare**.**have**;  
 **return** have;  
 }  
  
 **public void** betterNationalWelfare(){  
 System.***out***.println(getName() + **" приумножает свое богатство."**);  
 **if** (**nationalWelfare** == NationalWelfare.***Poor***){**nationalWelfare** = NationalWelfare.***MiddleClass***;}  
 **if** (**nationalWelfare** == NationalWelfare.***MiddleClass***){**nationalWelfare** = NationalWelfare.***Rich***;}  
 **if** (**nationalWelfare** == NationalWelfare.***Rich***){**nationalWelfare** = NationalWelfare.***Top10Forbes***;}  
 }  
 **public void** worseNationalWelfare(){  
 System.***out***.println(**"Дела "** + getName() + **"а идут хуже."**);  
 **if** (**nationalWelfare** == NationalWelfare.***Top10Forbes***){**nationalWelfare** = NationalWelfare.***Rich***;}  
 **if** (**nationalWelfare** == NationalWelfare.***Rich***){**nationalWelfare** = NationalWelfare.***MiddleClass***;}  
 **if** (**nationalWelfare** == NationalWelfare.***MiddleClass***){**nationalWelfare** = NationalWelfare.***Poor***;}  
 }  
  
 **public void** condition(){  
 System.***out***.print(getName()+ **" "**+getNationalWelfare());  
 }  
 **public void** payDracula(**boolean** n){}  
 **public void** taxes(){  
 }  
  
}

**Dracula.java**

**package** People;  
  
**import** Quality.NationalWelfare;  
**import** Quality.Protect;  
  
**public class** Dracula **extends** PersonAbstract **implements** Protect {  
 **private boolean know**;  
 **public** Dracula(String name, NationalWelfare rich, **boolean** know) {  
 **super**(name, rich);  
 **this**.**know** = know;  
 }  
  
 **public void** comeback(){  
 System.***out***.println(**"Вернулся из поездки "** + getName()+ **" который "**+ getNationalWelfwareHave()+ **"."**);  
 }  
  
 **public void** setKnowledge(){  
 **know** = **true**;  
 System.***out***.print(getName() + **" узнал, что "**);  
 }  
 **public boolean** getKnow(){  
 **return know**;  
 }  
 @Override  
 **public** String toString() {  
 **return "Имя:"** + getName() + **". Состояние: "** + getNationalWelfare()+ **". Знает о воровстве: "** + getKnow() + **". "**;  
 }  
 @Override  
 **public void** fence() {  
 **if** (**know**){  
 System.***out***.println(getName() + **" велит обнести побережье забором."**);  
 } **else** {  
 System.***out***.println(**" не замечает воровства."**);  
 }  
 }  
  
}

**Lunatiki.java**

**package** People;  
  
**import** Quality.NationalWelfare;  
**import** Quality.Steal;  
  
**public class** Lunatiki **extends** PersonAbstract **implements** Steal {  
 **private boolean know**;  
 **private int have**;  
 **public** Lunatiki(String name, NationalWelfare nationalWelfare) {  
 **super**(name, nationalWelfare);  
 }  
  
 **public void** setKnowledge(){  
 **know** = **true**;  
 System.***out***.println(getName() + **" теперь знает где брать соль. "**);  
  
 }  
**public void** OpenAFactory(){  
 System.***out***.println(getName() + **" сберег "** + getNationalWelfwareCash() + **" и строит завод."**);  
 **have** += 1;  
 }  
  
 **public boolean** getKnowledge(){  
 **return know**;  
 }  
  
  
 @Override  
 **public void** steal() {  
 **if** (**have** != 0){  
 System.***out***.println(getName() + **" ворует кристаллы для переработки на соляном заводе. "**);  
 }  
 }  
  
 @Override  
 **public void** payDracula(**boolean** n) {  
 **if** (n && **have**!= 0){  
 System.***out***.println(getName() + **" платит Дракуле за кристаллы. "**);  
 }  
 }  
  
 @Override  
 **public void** taxes() {  
 **if** (**have** != 0){  
 System.***out***.println(getName() + **" платит половину прибыли с завода в карман Дракулы. "**);  
 worseNationalWelfare();  
 }  
 }  
  
 @Override  
 **public** String toString() {  
 **return "Имя:"** + getName() + **". Состояние: "** + getNationalWelfare()+ **". Знает где брать соль: "** + getKnowledge() + **". "**;  
 }  
  
 @Override  
 **public int** hashCode() {  
 **return** getNationalWelfare().hashCode() + Boolean.*toString*(getKnowledge()).hashCode();  
 }  
  
 @Override  
 **public boolean** equals(Object o) {  
 **if** (o **instanceof** Lunatiki){  
 **if** (**this**.hashCode() == o.hashCode()) {  
 **return true**;  
 }  
 }  
 **return false**;  
 }  
}

**Ponchic.java**

**package** People;  
  
**import** Quality.\*;  
  
**public class** Ponchic **extends** PersonAbstract{  
 **private boolean duration**;  
 **private final double earnStd** = 20;  
 **private double earn** = 20;  
 **public** Ponchic(String name, NationalWelfare nationalWelfare, **boolean** duration) {  
 **super**(name, nationalWelfare);  
 **this**.**duration** = duration;  
 }  
  
 **public void** duration(){  
 **if** (**duration**) {  
 System.***out***.println(**" долго. "**);  
 } **else** {  
 System.***out***.println(**" недолго. "**);  
 }  
  
 }  
 **public void** earnCh(){  
 **this**.**earn** = **earn** - 9.5;  
 }  
  
 **public void** getEarn() {  
 System.***out***.println(getName() + **" раньше, тратит на оплату рабочего фертинг и получает взамен "** + **earnStd** + **"."**);  
 **if** (**earn** > **earnStd**) {  
 System.***out***.println(**"Теперь рабочий приносит целых"** + **earn** + **"."**);  
 } **else**{  
 System.***out***.println(**"Теперь никто не может выколотить из рабочего и "** + **earn** + **" фертинга."**);  
 }  
  
 }  
 @Override  
 **public** String toString() {  
 **return "Имя:"** + getName() + **". Состояние: "** + getNationalWelfare()+ **". "**;  
 }  
 @Override  
 **public void** payDracula(**boolean** n){  
 **if** (n) {  
 System.***out***.println(getName() + **" платит Дракуле за кристаллы."**);  
 } **else** {System.***out***.println(getName() + **" ура налогов нет."**);}  
 }  
  
 @Override  
 **public void** taxes() {  
 System.***out***.println(getName() + **" платит половину прибыли с завода в карман Дракулы. "**);  
 worseNationalWelfare();  
 }  
}

# Результаты работы программы:

Пончик очень богат недолго.

Лунатики теперь знает где брать соль.

Первый лунатик теперь знает где брать соль.

Второй лунатик теперь знает где брать соль.

Третий лунатик теперь знает где брать соль.

Второй лунатик сберег достаточная сумма и строит завод.

Появляется небольшой соляной заводик.

Второй лунатик приумножает свое богатство.

Третий лунатик сберег достаточная сумма и строит завод.

Появляется небольшой соляной заводик.

Третий лунатик приумножает свое богатство.

Соль добывается все больше.

Соль продается все дешевле.

Соль уже не приносит таких барышей как вначале.

Пончик раньше, тратит на оплату рабочего фертинг и получает взамен 20.0.

Теперь никто не может выколотить из рабочего и 1.0 фертинга.

Все еще выгодно, солевой промысел развивается

Вернулся из поездки Дракула который владеет всем побережьем.

Дела Пончика идут хуже.

Дракула узнал, что Второй лунатик ворует кристаллы для переработки на соляном заводе.

Третий лунатик ворует кристаллы для переработки на соляном заводе.

Дракула велит обнести побережье забором.

Пончик платит Дракуле за кристаллы.

Второй лунатик платит Дракуле за кристаллы.

Третий лунатик платит Дракуле за кристаллы.

Пончик платит половину прибыли с завода в карман Дракулы.

Дела Пончика идут хуже.

Второй лунатик платит половину прибыли с завода в карман Дракулы.

Дела Второй лунатика идут хуже.

Третий лунатик платит половину прибыли с завода в карман Дракулы.

Дела Третий лунатика идут хуже.

# Вывод:

В ходе лабораторной работы я закрепила основные принципы ООП, изучила принципы SOLID и научилась применять их на практике.