Федеральное государственное автономное образовательное учреждение высшего образования **«Национальный исследовательский университет ИТМО»**

Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа по программированию №3**

вариант: 852

Преподаватель: Харитонова Анастасия Евгеньевна

Выполнил: Дагаев Даниил Сергеевич

Группа: Р3108

Санкт-Петербург, 2021г

Оглавление

[Задание 2](#_Toc89011785)

[Диаграмма 2](#_Toc89011786)

[Исходный код программы 2](#_Toc89011787)

[Результат работы программы 3](#_Toc89011788)

[Вывод 3](#_Toc89011789)

# Задание

**Описание предметной области, по которой должна быть построена объектная модель:**

"Вот те раз!" -- подумал Незнайка, который умел писать только печатными буквами и очень стеснялся показать свою необразованность. Скоро стемнело. Вокруг загорелись сотни разноцветных фонариков. Они сверкали и на деревьях и на палатках. Кое-где они были спрятаны в траве под деревьями, и от этого казалось, что сама трава светится каким-то волшебным светом. Нижняя часть беседки, над которой помещался оркестр, была задернута красивым голубым занавесом. Неожиданно занавес открылся, и все увидели за ним сцену.

**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).

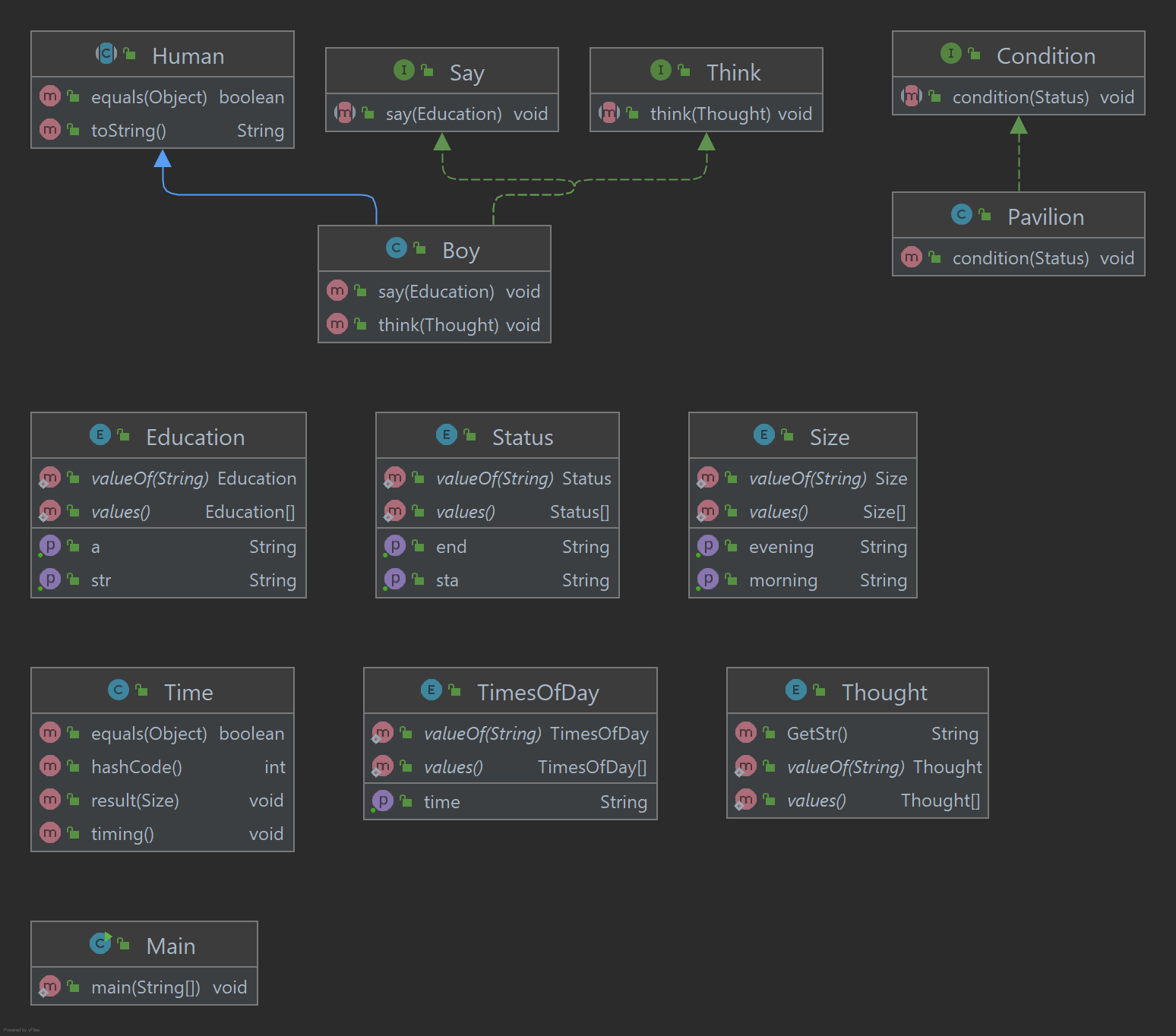
**Порядок выполнения работы:**

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.

**Отчёт по работе должен содержать:**

1. Текст задания.
2. Диаграмма классов объектной модели.
3. Исходный код программы.
4. Результат работы программы.
5. Выводы по работе.

# Диаграмма



# Исходный код программы

public class Main {  
 public static void main(String[] args) {  
 Boy neznaika = new Boy("Yeah boy");  
 Pavilion besedka = new Pavilion();  
 Time timeday = new Time(TimesOfDay.*NIGHT*);  
 neznaika.think(Thought.*VOT*);  
 neznaika.say(Education.*NO*);  
 timeday.timing();  
 timeday.result(Size.*LONG*);  
 besedka.condition(Status.*OPENED*);  
 }  
}

public class Boy extends Human implements Think,Say{  
 public void think(Thought th){  
 System.*out*.print(th.GetStr() +"-подумал " + name + ',');  
 }  
 public Boy(String nm)  
 {  
 name = nm;  
 }  
 public void say(Education b){  
 System.*out*.print("который " + b.getA() + " и");  
 System.*out*.print(b.getStr()+ ".");  
 }  
  
}

public abstract class Human{  
 String name;  
  
 @Override  
 public String toString() {  
 return this.name;  
 }  
  
 public boolean equals(Object obj) {  
 if (this.getClass() != obj.getClass()){  
 return false;  
 }  
 else{  
 return true;  
 }  
 }  
}

public class Pavilion implements Condition {  
 public void condition(Status a){  
 System.*out*.print("\nНижняя часть беседки, над которой помещался оркестр,"+ a.getSta() +"."+ "Неожиданно занавес " +a.getEnd()+".");  
 }  
}

import java.util.Objects;  
  
public class Time {  
 private TimesOfDay s1;  
 public Time(TimesOfDay a){  
 s1 = a;  
 }  
 public void timing(){  
 System.*out*.print('\n' + s1.getTime()+'.');  
 }  
 public void result(Size n){  
 if (s1.equals(TimesOfDay.*NIGHT*)){  
 System.*out*.print(n.getEvening());  
 }  
 else {System.*out*.print(n.getMorning());}  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 else {  
 return false;  
 }  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(s1);  
 }  
}

public interface Condition {  
 void condition(Status a);  
}

public interface Think {  
 void think(Thought tv);  
}

public interface Say {  
 void say(Education b);  
}

public enum Education {  
 *YES*(" имел 900 iq",""),*NO*(" очень стеснялся показать свою необразованность","умел читать только печатные буквы");  
 private String st;  
 private String a;  
 Education(String str,String h){  
 st = str;  
 a = h;  
 }  
 public String getStr(){  
 return st;  
 }  
 public String getA(){  
 return a;  
 }  
}

public enum Size {  
 *LONG*("Вокруг загорелись сотни разноцветных фонариков.Они сверкали и на деревьях и на палатках.\nКое-где они были спрятаны в траве под деревьями, и от этого казалось, что сама трава светится каким-то волшебным светом.","Солнце освещало что-то там, вот, даааа"),*SHORT*("Вокруг загорелись сотни разноцветных фонариков.","Солнце освещало что-то там"),*MEDIUM*("Вокруг загорелись сотни разноцветных фонариков.\nОни сверкали и на деревьях и на палатках.","Солнце освещало что-то там, вот");  
 private String s2;  
 private String s4;  
 Size(String s1,String s3){  
 s2 = s1;  
 s4 = s3;  
 }  
 public String getEvening(){  
 return s2;  
 }  
 public String getMorning(){  
 return s4;  
 }  
}

public enum Status {  
 *OPENED*("была задернута красивым голубым занавесом","открылся, и все увидели за ним сцену"),*CLOSED*("была открыта","закрылся и никто ничего не увидел");  
 private String a;  
 private String b;  
 Status(String h,String b){  
 a = h;  
 this.b = b;  
 }  
 public String getSta(){  
 return a;  
 }  
 public String getEnd(){  
 return b;  
 }  
}

public enum Thought {  
 *VOT*("Вот те раз!"),*NEVOT*("Вот не раз!");  
 private String val;  
 Thought(String str){  
 val = str;  
 }  
 public String GetStr(){  
 return val;  
 }  
}

public enum TimesOfDay{  
 *NIGHT*("Скоро стемнело"),*DAY*("Солнце взошло");  
 String st;  
 TimesOfDay(String str){  
 st = str;  
 }  
 public String getTime(){  
 return st;  
 }  
  
}

# Результат работы программы

Вот те раз!-подумал Yeah boy,который умел читать только печатные буквы и очень стеснялся показать свою необразованность.

Скоро стемнело.Вокруг загорелись сотни разноцветных фонариков.Они сверкали и на деревьях и на палатках.

Кое-где они были спрятаны в траве под деревьями, и от этого казалось, что сама трава светится каким-то волшебным светом.

Нижняя часть беседки, над которой помещался оркестр,была задернута красивым голубым занавесом.Неожиданно занавес открылся, и все увидели за ним сцену.

# Вывод

Во время выполнения данной лабораторной работы я научился применять принципы SOLID на практике, разобрал интерфейсы, перечисления enum, абстрактные классы и не выспался.