

Лабораторная работа №2

Задание №1

Тут слишком много написано! Упростите код.

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        String buttonName = "Button";
        String bottomMenu = "bottomMenu";
        MenuButton button1 = new MenuButton(buttonName,buttonName);
        button1.click();
    }
}
interface ClickHandler{
    void click();
}
abstract class Button implements ClickHandler{
    private String name;
    @Override
    public void click() {
        System.out.printf("Нажал на кнопку %s \n",name);
    }
    public String getName() {
        return name;
    }

    public void setName(String name) {
        if(name.equals("")){
            System.out.println("Название Кнопки неопределено");
            return;
        }
        this.name = name;
    }
}
class MenuButton extends Button implements ClickHandler{
    private String buttonMenu;

    public MenuButton(String buttonName,String buttonMenu) {
        setButtonMenu(buttonMenu);
        setName(buttonName);
    }
}
```

```

@Override
public void click() {
    System.out.printf("Нажал на кнопку %s в меню %s\n", getName(), buttonMenu);
}

public String getButtonMenu() {
    return buttonMenu;
}

public void setButtonMenu(String buttonMenu) {
    this.buttonMenu = buttonMenu;
}
}

```

Задание №2

Данная система классов сделана плохо! Переработайте её.

```

class Print{
    void print(){
        System.out.println("Вывод");
    };
}
class CountID extends Print{
    void count(){

    }
    @Override
    void print() {
        super.print();
    }
}
class Account extends CountID{
    private String firstName;
    private int age;
    static int idCounter = 0;
    private int id;
    @Override
    void count() {

```

```

        idCounter++;
    }
    @Override
    void print() {
        super.print();
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public int getId() {
        return age;
    }

    public Account(String firstName, int age) {
        this.firstName = firstName;
        this.age = age;
        Id = idCounter;
        count();
    }
}

class BankAccount extends Account{
    public double balance;
    public BankAccount(String firstName, int age) {
        super(firstName, age);
        balance = 0;
    }
}

```

```
}

@Override
void count() {
    super.count();
}

@Override
void print() {
    super.print();
}

public double getBalance() {
    return balance;
}

public void setBalance(double balance) {
    this.balance = balance;
}

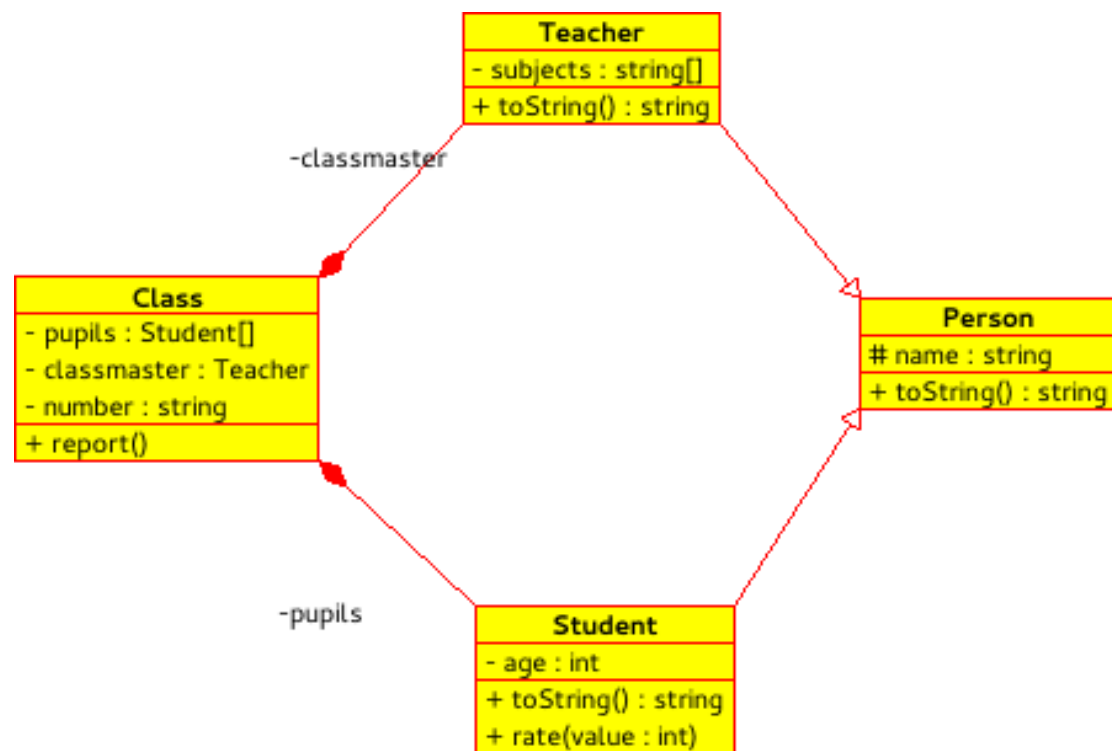
}
```

Задание №3

Напишите интерфейс Converter и классы, расширяющие его, для конвертации из градусов по Цельсию в Кельвины, Фаренгейты, и так далее. У интерфейса должен быть метод *convert*, который и делает конвертацию.

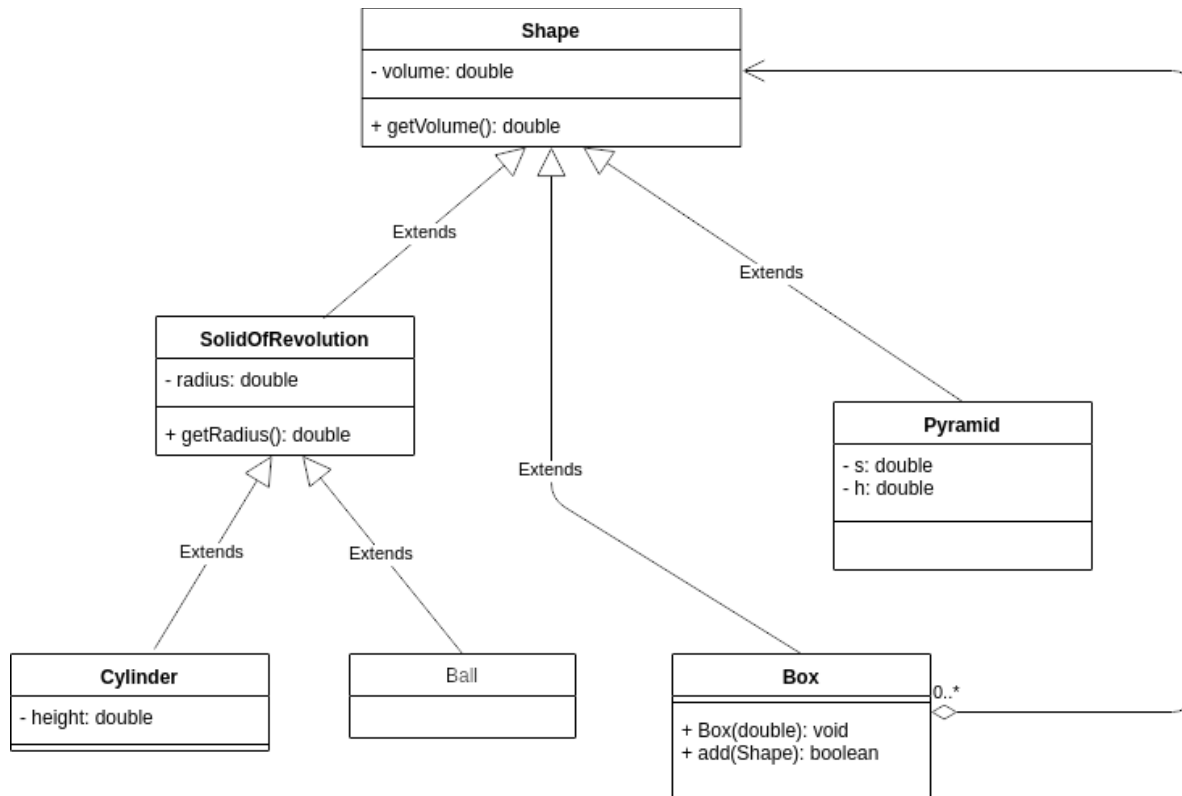
Задание №4

Реализуйте иерархию классов:



Задание №5

Реализуйте иерархию классов:



Класс **Box** является контейнером, он может содержать в себе другие фигуры. Метод `add()` принимает на вход `Shape`. Нужно добавлять новые фигуры до тех пор, пока для них хватает места в `Box` (будем считать только объём, игнорируя форму. Допустим, мы переливаем жидкость). Если места для добавления новой фигуры не хватает, то метод должен вернуть `false`.

Задание №6

Напишите метод, который на вход получает коллекцию объектов, а возвращает коллекцию уже без дубликатов.

Задание №7

Написать итератор по массиву

Задание №8

Напишите метод, который добавляет 1000000 элементов в `ArrayList` и `LinkedList`. Напишите еще один метод, который выбирает из заполненного списка элемент наугад 100000 раз. Замерьте время, которое потрачено на это. Сравните результаты и предположите, почему они именно такие.