Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Название практического занятия**»

Выполнил: Плоских Александр Павлович

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Задание:**

1. Создать абстрактный класс и интерфейс в соответствии с вариантом.

2. На их основе описать базовый класс-наследник с указанными в

задании полями и методами:  2 Конструктора;

• Функция, которая определяет качество объекта – Q по заданной формуле;

• Вывод информации об объекте.

3.Построить класс потомок, который содержит:

• Дополнительное поле Р;

• Функция, которая определяет «качество» объекта класса потомка – Qp, которая

перекрывает функцию качества базового класса, выполняя вычисления по

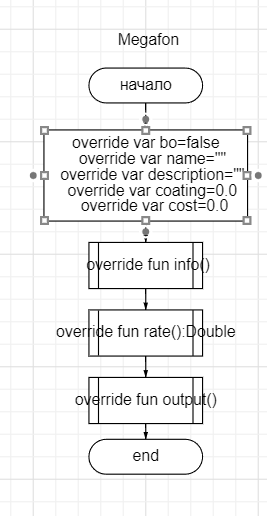
новой формуле.

3. Добавить еще 1 класс, который будет использовать функции

интерфейса.

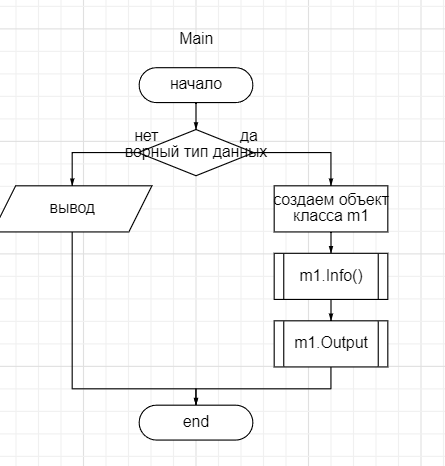
4. Создать проект для демонстрации работы: ввод-вывод информации об

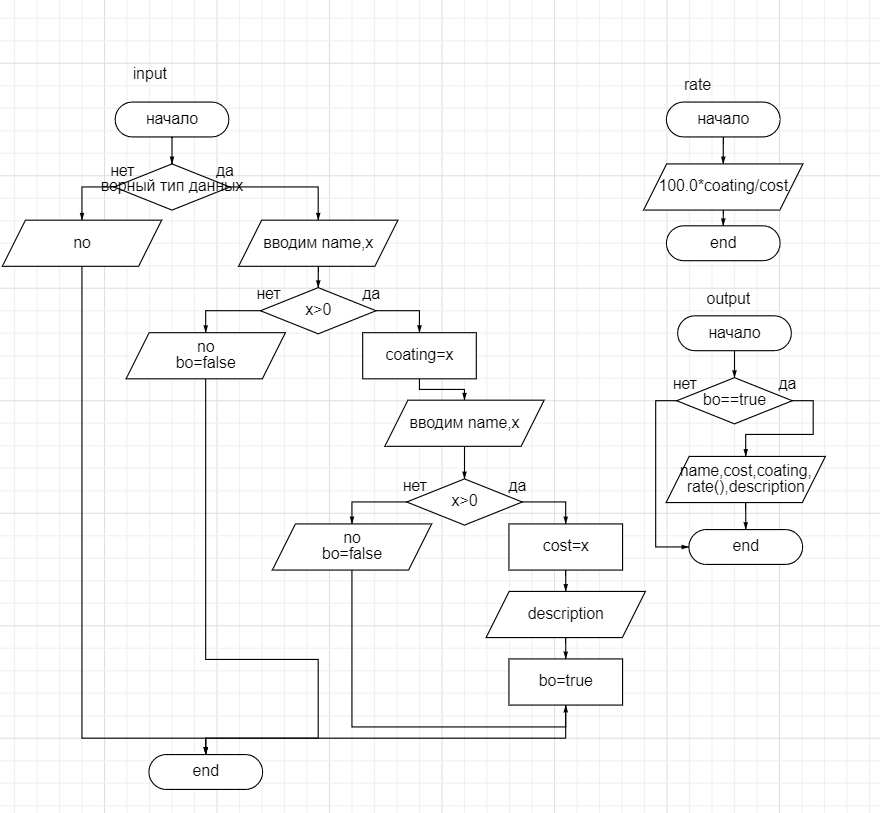
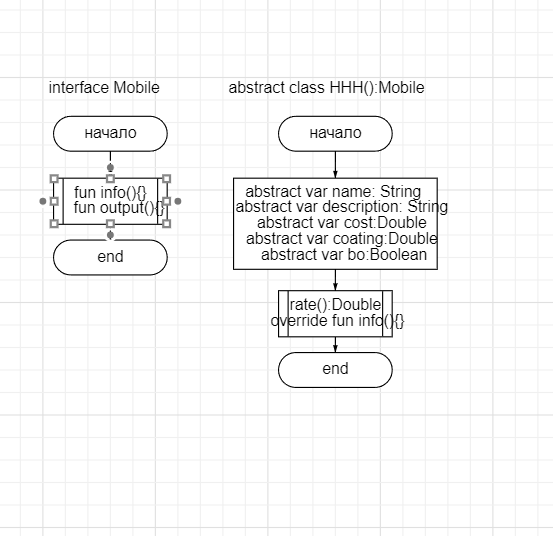
объектах базового класса и класса потомка.

**Входные и выходные данные**

Класс Megafon

**Блок-схема**

****

****

**Листинг программы (если есть)**

Основная программа

fun main(){  
 try{  
 var m1:Megafon=Megafon()  
 m1.info()  
 m1.output()  
 }catch (e: Exception){  
 *println*("no")  
 }  
}

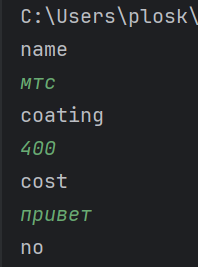
Класс Megafon

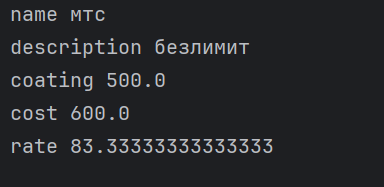
class Megafon():HHH() {  
 override var bo=false  
 override var name=""  
 override var description=""  
 override var coating=0.0  
 override var cost=0.0  
 override fun info() {  
 try{  
 var x:Double  
 *println*("name")  
 name= *readln*()  
 *println*("coating")  
 x= *readln*()!!.*toDouble*()  
 if(x>0){  
 coating= x  
 *println*("cost")  
 x= *readln*()!!.*toDouble*()  
 if(x>0){  
 cost= x  
 *println*("description")  
 description= *readln*()!!.toString()  
 bo=true  
 }  
 else{  
 *println*("no")  
 bo=false  
 }  
 }  
 else{  
 *println*("no")  
 bo=false  
 }  
  
 }catch (e:Exception){  
 *println*("no")  
 }  
 }  
  
  
  
 override fun rate():Double {  
 return (100.0\*coating/cost)  
 }  
 override fun output() {  
 if(bo==true){  
 *println*("name $name")  
 *println*("description $description")  
 *println*("coating $coating")  
 *println*("cost $cost")  
 *println*("rate ${rate()}")  
 }  
 }  
}

Интерфейс Mobile и абстрактный класс HHH

interface Mobile{  
 fun info(){}  
 fun output(){}  
}  
  
abstract class HHH():Mobile{  
 abstract var name: String  
 abstract var description: String  
 abstract var cost:Double  
 abstract var coating:Double  
 abstract var bo:Boolean  
 open fun rate():Double{  
 return 0.0  
 }  
 override fun info(){}  
  
}

**Тестовые ситуации**

****

****

**Вывод**

Выполняя это задание, я научился пользоваться интерфейсами, абстрактными классами на языке программирования, а также улучшил свои навыки в использование классов наследников и использование базовых конструкций