Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическая работа 10**»

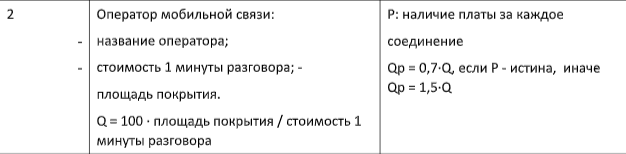
Выполнил: Малюгин Кирилл Алексеевич

Группа: ПР-22

Преподаватель: Мирошниченко Г.В

2023

**Задание:**

****

**Входные данные**

***Operator:***

**Plata1 – присутствие платы за соединение, логический тип (boolean)**

**Nameoper1 – наименование оператора связи, строковый тип (string)**

**S1 – площадь покрытия, целый тип (int)**

**Onemin1 – стоимость за одну минуту разговора, вещественный тип (double)**

***Plata:***

**Plata1 – присутствие платы за соединение, логический тип (boolean)**

**Nameoper1 – наименование оператора связи, строковый тип (string)**

**S1 – площадь покрытия, целый тип (int)**

**Onemin1 – стоимость за одну минуту разговора, вещественный тип (double)**

**Pris – ввод числа, для проверки условия на присутсвие платы за соединение, целый тип (int)**

**Выходные данные**

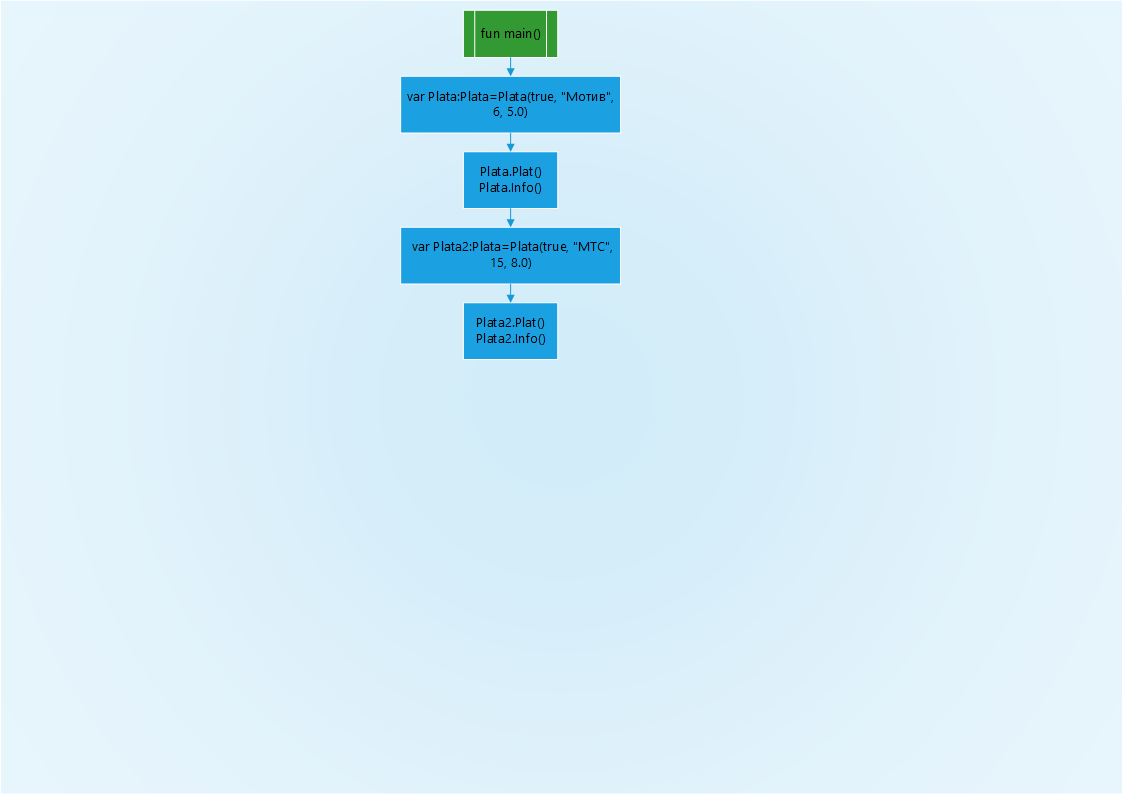
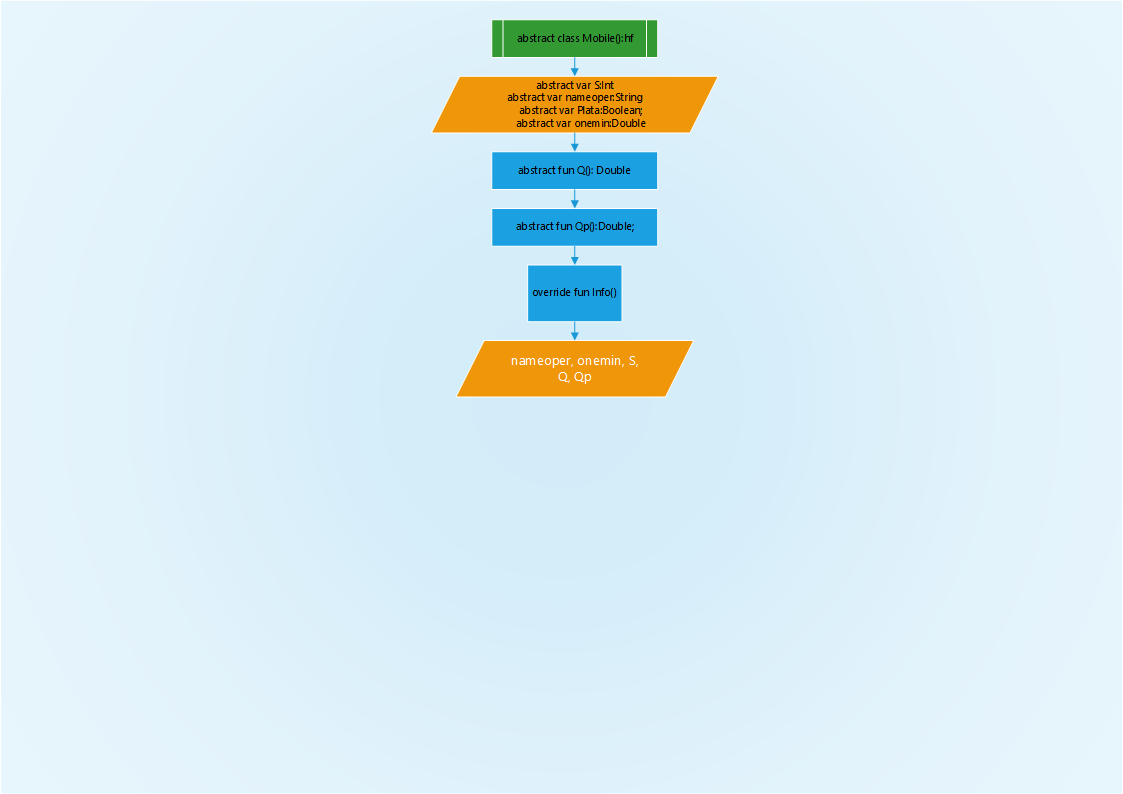
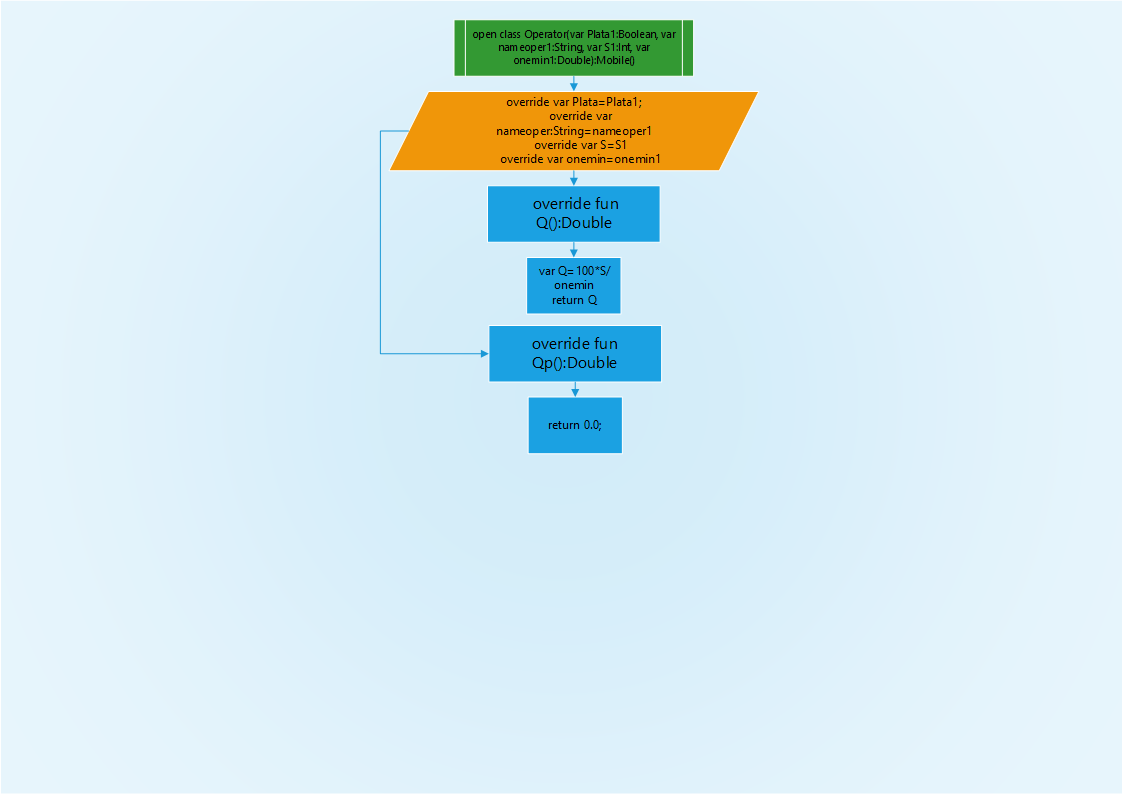
**Nameoper - наименование оператора связи, строковый тип (string)**

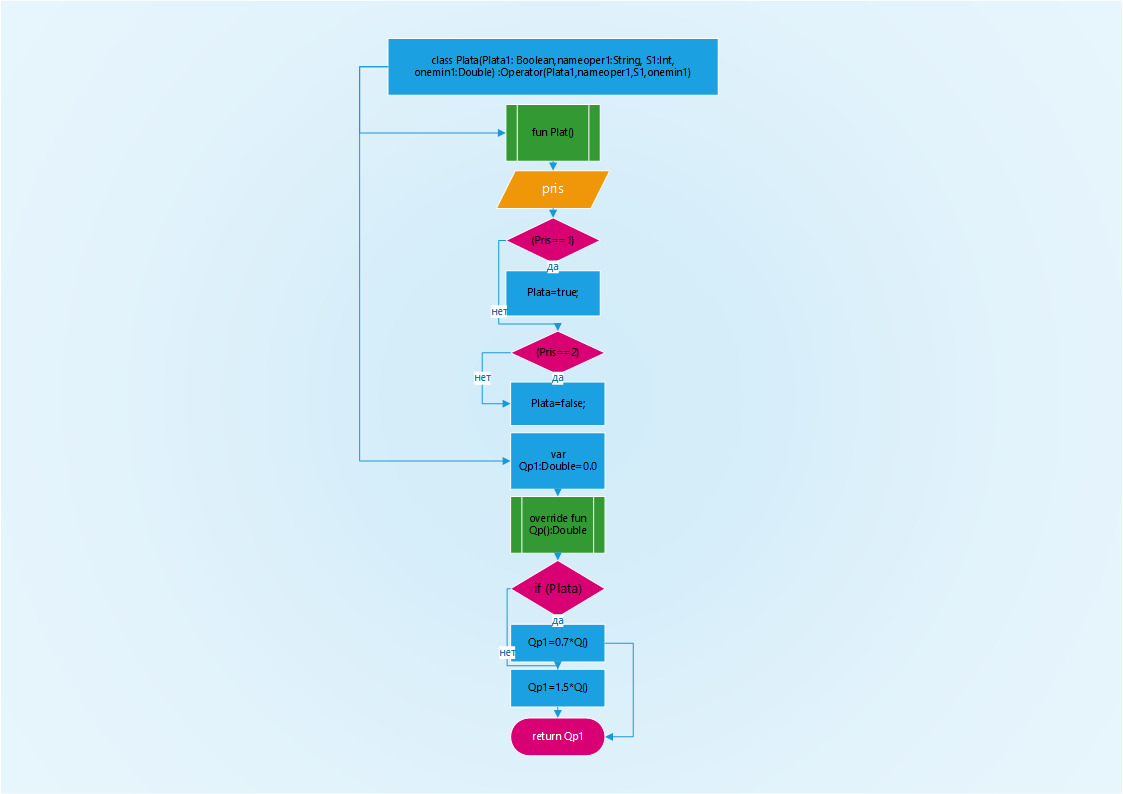
**Onemin – стоимость за одну минуту разговора, вещественный тип (double)**

**S1 – площадь покрытия, целый тип (int)**

**Q , Qp**

**Блок схема**

****

****

**Листинг программы**

interface hf {  
 fun Info()  
}

abstract class Mobile():hf {  
 abstract var S:Int  
 abstract var nameoper:String  
 abstract var Plata:Boolean;  
 abstract var onemin:Double  
 abstract fun Q(): Double  
 abstract fun Qp():Double;  
 override fun Info()  
 {  
 *println*("Оператор: $nameoper, Стоимость минуты: $onemin, Площадь покрытия: $S, Q=${Q()}, Qp=${Qp()}")  
 }  
  
}

open class Operator(var Plata1:Boolean, var nameoper1:String, var S1:Int, var onemin1:Double):Mobile(){  
 override var Plata=Plata1;  
 override var nameoper:String=nameoper1  
 override var S=S1  
 override var onemin=onemin1  
 override fun Q():Double  
 {  
 var Q=100\*S/onemin  
 return Q  
 }  
 override fun Qp():Double  
 {  
 return 0.0;  
 }  
}

class Plata(Plata1: Boolean,nameoper1:String, S1:Int, onemin1:Double) :Operator(Plata1,nameoper1,S1,onemin1)  
{  
 fun Plat()  
 {  
 *print*("Присутствует ли плата за соединение? 1-Да, 2-Нет: ")  
 var Pris= *readLine*()!!.*toInt*()  
 if (Pris==1)  
 Plata=true;  
 else if (Pris==2)  
 Plata=false  
 else  
 {*println*("Ошибочный ввод данных, выставлен режим отсутствия платы за соединение"); Plata=false;}  
  
 }  
 var Qp1:Double=0.0  
 override fun Qp():Double  
 {  
 if (Plata)  
 Qp1=0.7\*Q()  
 else Qp1=1.5\*Q()  
 return Qp1  
 }  
}

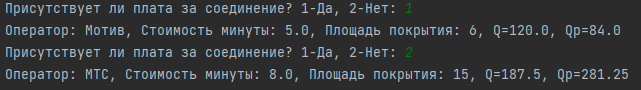
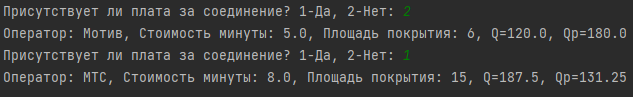
fun main()  
{  
 try {  
 var Plata: Plata = Plata(true, "Мотив", 6, 5.0)  
 Plata.Plat()  
 Plata.Info()  
 var Plata2: Plata = Plata(true, "МТС", 15, 8.0)  
 Plata2.Plat()  
 Plata2.Info()  
 }  
 catch (e:Exception)  
 {  
 *println*("Ошибка при вводе данных")  
 }  
}

**Тестовые ситуации**

****

****

**Вывод**

**** ****