Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое занятие 10»**

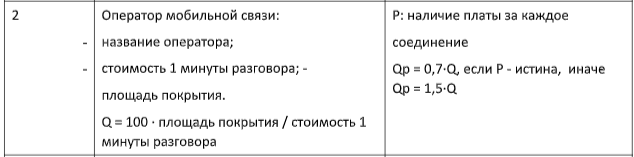
Выполнил: Рявкин Лев Алексеевич

Группа: ПР-22

Преподаватель: Мирошниченко Г.В

2023

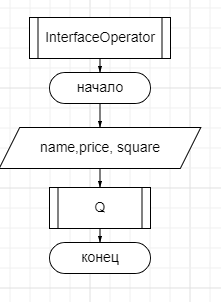
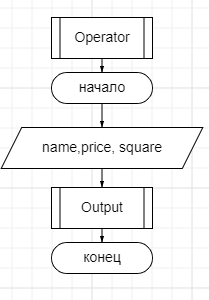
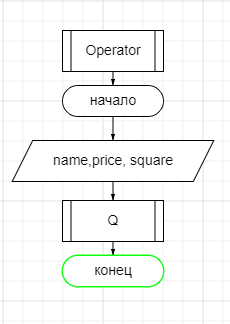
**Задание:**

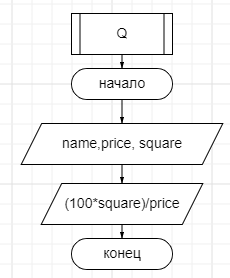
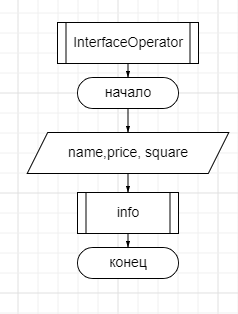


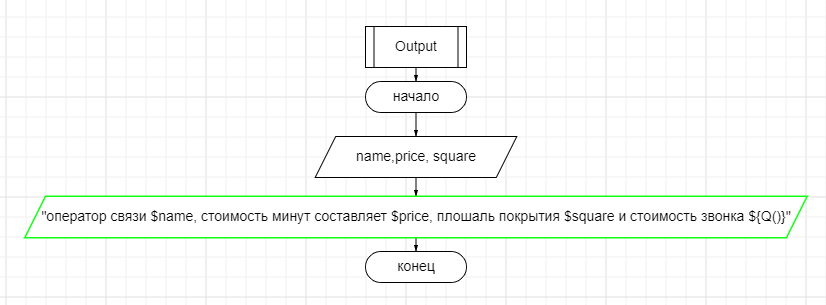
**Входные и выходные данные**

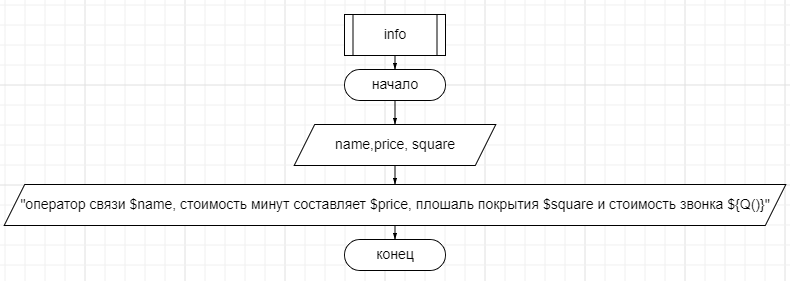
name,price, square,P,price2, square2

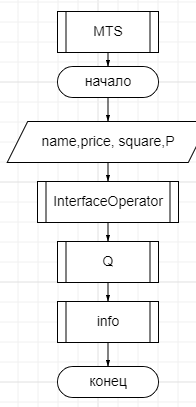
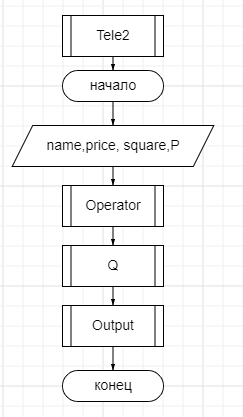
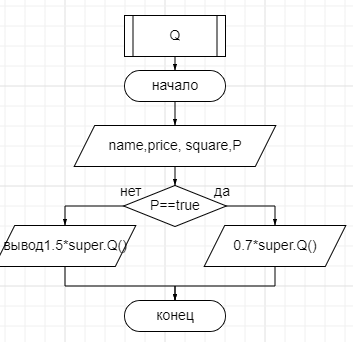
**Блок-схема**













**Листинг программы**

interface InterfaceOperator  
{  
 var name:String  
 var price:Double  
 var square:Double  
  
  
 fun q(): Double  
  
 fun info():String

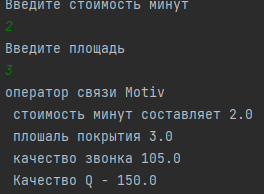
fun main()  
{  
 *println*("Введите стоимость минут")  
 var price= *readLine*()!!.*toDouble*()  
 *println*("Введите площадь")  
 var square= *readLine*()!!.*toDouble*()  
 if(price>0&&square>0)  
 {  
 var motiv=Motiv("Motiv",price,square,true)  
 *println*(motiv.Info())  
 }  
 else *println*("Введены отрицательные значения")  
  
 *println*("Введите стоимость минут")  
 var price2= *readLine*()!!.*toDouble*()  
 *println*("Введите площадь")  
 var square2= *readLine*()!!.*toDouble*()  
 if (price2>0&&square2>0)  
 {  
 var mts=MTS("MTS",price2,square2,false)  
 *println*(mts.info())  
 }  
 else *println*("Введены отрицательные значения")  
  
}

abstract class Operator(\_name:String,\_price:Double,\_square:Double):InterfaceOperator  
{  
 override var name: String=\_name  
 override var price: Double=\_price  
 override var square: Double=\_square  
 constructor(price:Double,square: Double):this("",price,square)  
 abstract fun Q():Double  
  
  
 override fun q(): Double  
 {  
 var q=100\*square/price  
 return q  
 }  
  
 override fun info(): String {  
 return "оператор связи $name\n стоимость минут составляет $price\n плошаль покрытия $square\n качество звонка ${Q()}\n Качество Q - ${q()}"  
 }

class Motiv(name:String,price:Double,square:Double,public var P:Boolean):Operator(name,price,square)  
{  
 constructor(name:String,price:Double,square: Double):this(name,price,square,true)  
 override fun Q():Double {  
 if (P==true)  
 return 0.7\*super.q()  
 else  
 return 1.5\*super.q()  
 }  
  
 override fun q(): Double  
 {  
 var q=100\*square/price  
 return q  
 }  
 fun Info(): String  
 {  
 return "оператор связи $name\n стоимость минут составляет $price\n плошаль покрытия $square\n качество звонка ${Q()}\n Качество Q - ${q()}"  
 }  
}

class MTS( name:String,price:Double, square:Double,public var P:Boolean):Operator(name,price,square)  
{  
 override fun Q():Double {  
 if (P==true)  
 return 0.7\*super.q()  
 else  
 return 1.5\*super.q()  
 }  
 override fun q(): Double  
 {  
 var q=100\*square/price  
 return q  
 }  
 override fun info(): String  
 {  
 return "оператор связи $name\n стоимость минут составляет $price\n плошаль покрытия $square\n качество звонка ${Q()}\n Качество Q - ${q()}"  
 }  
}

**Вывод данных**

****