Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое заданятие №10**»

Выполнил: Зиятдинова Алина Ленаровна

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Вариант 1. Вариант 4**

**Задание:**

**Входные данные**

type-тип кабеля, строковый тип

countjil-количество жил, целочисленный тип

diametr – диаметр, вещественный тип

**Выходные данные**

**Q –** вещественный тип

**Qr –** вещественный тип

**Листинг программы**

package ex1  
abstract class Cabel: Opletka(){  
 abstract var type: String  
 abstract var countjil:Int  
 abstract var diametr: Double  
 abstract fun Q():Double  
 abstract fun Qr():Double  
 abstract fun output()  
}

package ex1  
abstract class Opletka {  
 abstract var avability: Int  
}

package ex1  
import ex1.Cabel  
class MainClass: Cabel() {  
 override var type = " "  
 override var countjil = 0  
 override var diametr = 0.0  
 override var avability = 0  
 override fun Q(): Double {  
 var q: Double = 0.0  
 q = diametr / countjil  
 return q  
 }  
 override fun Qr(): Double {  
 var qr:Double = 0.0  
 if (avability == 1)  
 {  
 qr = 2\*Q()  
 }  
 else if (avability == 2)  
 {  
 qr = 0.7\*Q()  
 }  
 return qr  
 }  
 override fun output() {  
 *println*("Тип кабеля: ${type}\nКоличество жил: ${countjil}\nДиаметр: ${diametr}\nНаличие оплетки: ${avability}\nQ: ${Q()}\nQr: ${Qr()}\n")  
 }  
}

import ex1.MainClass  
fun main()  
{  
 var cabel1 = MainClass()  
 while (true) {  
 try {  
 *println*("Введите тип кабеля:")  
 cabel1.type = *readLine*()!!.toString()  
 while (cabel1.type.length <= 0)  
 {  
 *println*("Повотрите ввод. Введите тип кабеля.")  
 cabel1.type = *readLine*()!!.toString()  
 }  
 *println*("Введите диаметр:")  
 cabel1.diametr = *readLine*()!!.*toDouble*()  
 while (cabel1.diametr<0)  
 {  
 *println*("Диаметр может быть меньше нуля. Повотрите ввод.")  
 cabel1.diametr = *readLine*()!!.*toDouble*()  
 }  
 *println*("Введите количество жил:")  
 cabel1.countjil = *readLine*()!!.*toInt*()  
 while (cabel1.countjil<0)  
 {  
 *println*("Количество жил не может быть меньше нуля. Повотрите ввод.")  
 cabel1.countjil = *readLine*()!!.*toInt*()  
 }  
 *println*("Есть оплетка? Введите 1- да, 2-нет:")  
 cabel1.avability = *readLine*()!!.*toInt*()  
 while (cabel1.avability!=1 && cabel1.avability!=2)  
 {  
 *println*("Введите 1 или 2.")  
 cabel1.avability = *readLine*()!!.*toInt*()  
 }  
 break  
 }  
 catch (e:Exception)  
 {  
 *println*("Возникло исключение! Повторите ввод.")  
 }  
 }  
 var q = cabel1.Q()  
 var qr = cabel1.Qr()  
 cabel1.output()  
}

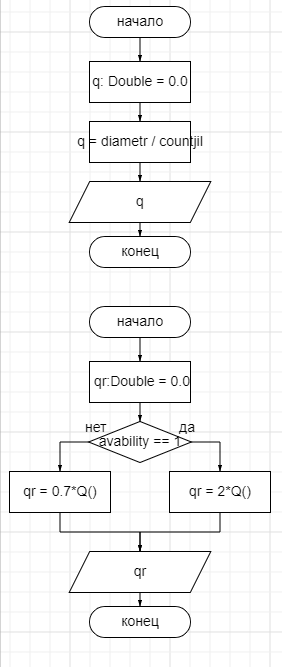
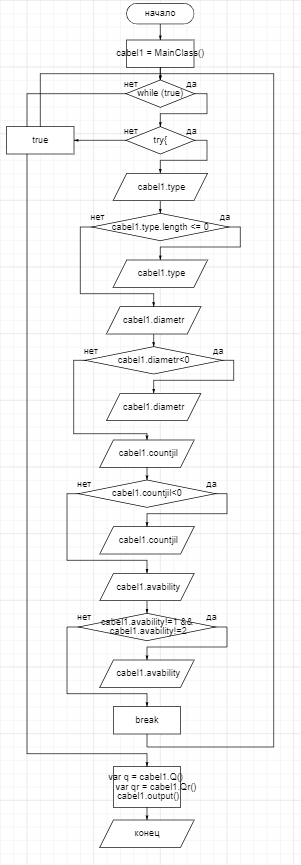
package ex2  
interface Cabel: Opletka{  
 var type: String  
 var countjil:Int  
 var diametr: Double  
 fun Q():Double  
 fun Qr():Double  
 fun output()  
}

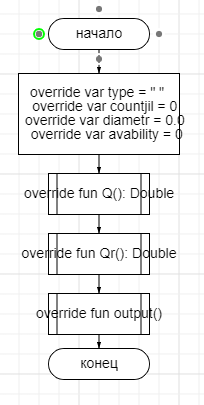
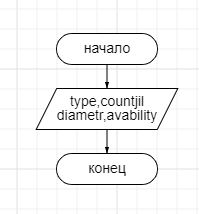
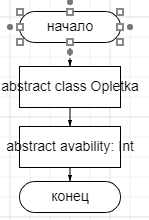
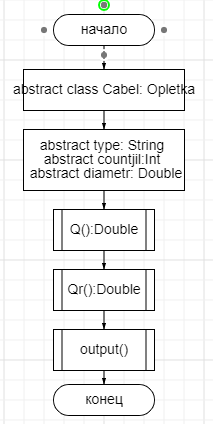
package ex2  
interface Opletka{  
var avability: Int  
}

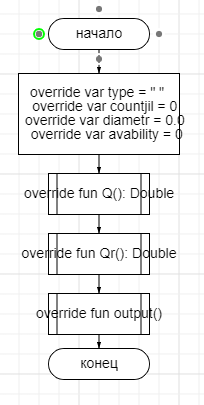
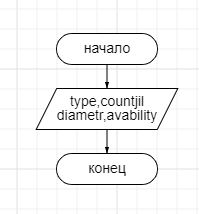
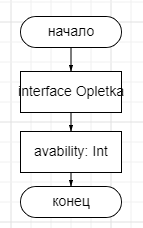
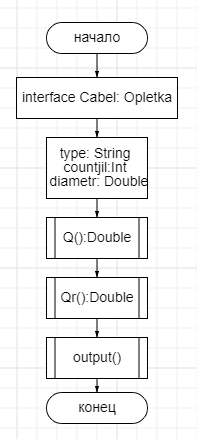
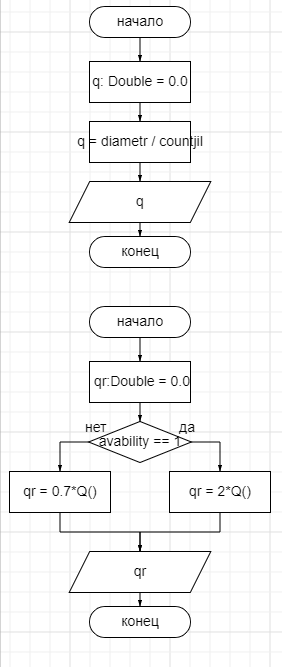
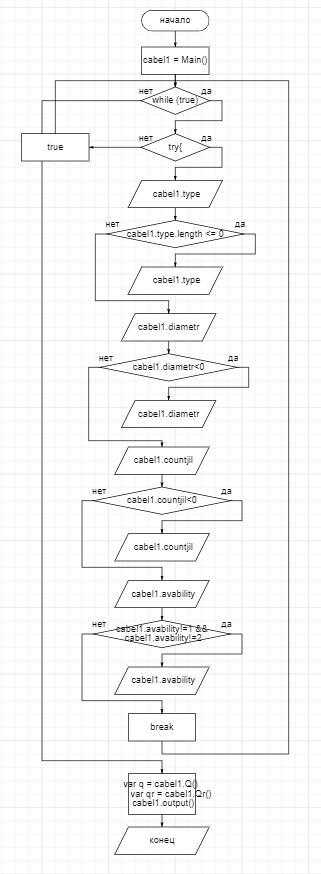
package ex2  
import ex2.Cabel  
class Main: ex2.Cabel{  
 override var countjil=0  
 override var diametr = 0.0  
 override var type = ""  
 override var avability = 0  
 override fun Q(): Double {  
 var q: Double = 0.0  
 q = diametr / countjil  
 return q  
 }  
 override fun Qr(): Double {  
 var qr:Double = 0.0  
 if (avability == 1)  
 {  
 qr = 2\*Q()  
 }  
 else  
 {  
 qr = 0.7\*Q()  
 }  
 return qr  
 }  
 override fun output() {  
 *println*("Тип кабеля: ${type}\nКоличество жил: ${countjil}\nДиаметр: ${diametr}\nНаличие оплетки: ${avability}\nQ: ${Q()}\nQr: ${Qr()}\n")  
 }

package ex2  
import ex2.Main  
fun main(){  
 var cabel1 = Main()  
 while (true) {  
 try {  
 *println*("Введите тип кабеля:")  
 cabel1.type = *readLine*()!!.toString()  
 while (cabel1.type.length <= 0)  
 {  
 *println*("Повотрите ввод. Введите тип кабеля.")  
 cabel1.type = *readLine*()!!.toString()  
 }  
 *println*("Введите диаметр:")  
 cabel1.diametr = *readLine*()!!.*toDouble*()  
 while (cabel1.diametr<0)  
 {  
 *println*("Диаметр может быть меньше нуля. Повотрите ввод.")  
 cabel1.diametr = *readLine*()!!.*toDouble*()  
 }  
 *println*("Введите количество жил:")  
 cabel1.countjil = *readLine*()!!.*toInt*()  
 while (cabel1.countjil<0)  
 {  
 *println*("Количество жил не может быть меньше нуля. Повотрите ввод.")  
 cabel1.countjil = *readLine*()!!.*toInt*()  
 }  
 *println*("Есть оплетка? Введите 1- да, 2-нет:")  
 cabel1.avability = *readLine*()!!.*toInt*()  
 while (cabel1.avability!=1 && cabel1.avability!=2)  
 {  
 *println*("Введите 1 или 2.")  
 cabel1.avability = *readLine*()!!.*toInt*()  
 }  
 break  
 }  
 catch (e:Exception)  
 {  
 *println*("Возникло исключение! Повторите ввод.")  
 }  
 }  
 var q = cabel1.Q()  
 var qr = cabel1.Qr()  
 cabel1.output()  
}

**Блок-схема**

**Абстракт.**



**Интерфейс.**

**Тестовые ситуации**

