Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое задание №8**»

Выполнил: Зиятдинова Алина Ленаровна

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Задание:**

**Входные и выходные данные**

**num\_poezd –** номер поезда, целочисленный тип

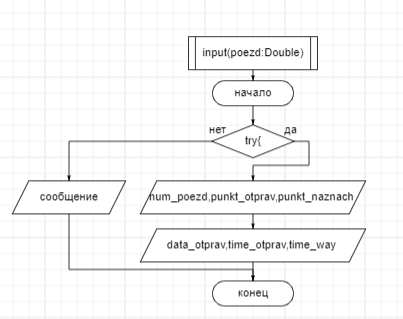
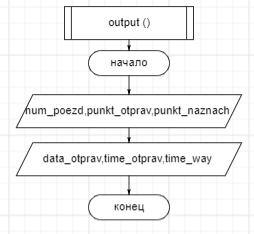
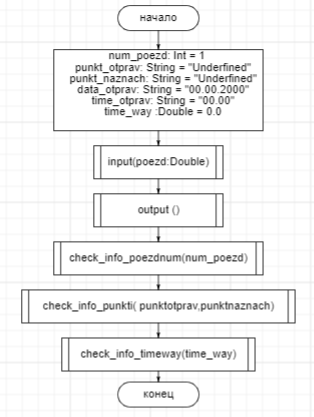
**punkt\_otprav-** пункт отправления, строковый тип

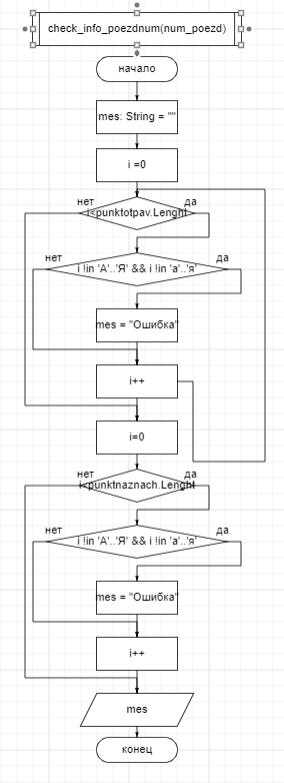
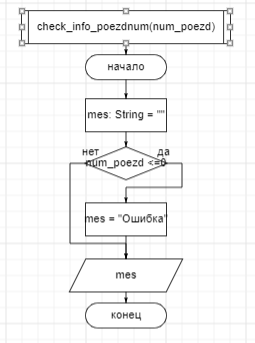
**punkt\_naznach –** пункт назначения, строковый тип

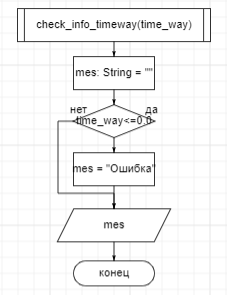
**data\_otprav** – дата отправления, строковый тип

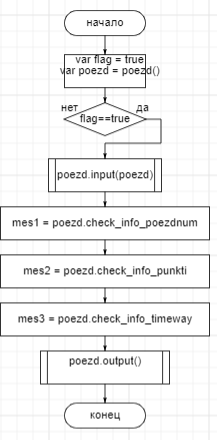
**time\_otprav –** время отправления, строковый тип

**time\_way –** время в пути. вещественный тип

**Блок-схема**



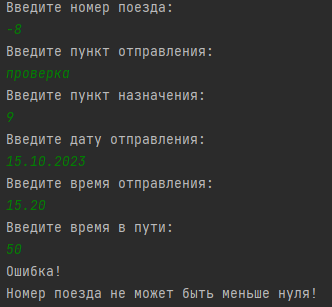


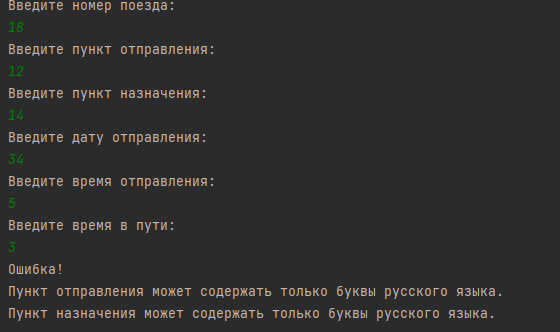


**Листинг программы (если есть)**

class poezd{  
 var num\_poezd: Int = 1  
 var punkt\_otprav: String = "Underfined"  
 var punkt\_naznach: String = "Underfined"  
 var data\_otprav: String = "00.00.2000"  
 var time\_otprav: String = "00.00"  
 var time\_way :Double = 0.0  
 fun input (poezd: poezd)  
 {  
 try {  
 *println*("Введите номер поезда:")  
 poezd.num\_poezd = *readLine*()!!.*toInt*()  
 *println*("Введите пункт отправления:")  
 poezd. punkt\_otprav = *readLine*()!!.toString()  
 *println*("Введите пункт назначения:")  
 poezd. punkt\_naznach = *readLine*()!!.toString()  
 *println*("Введите дату отправления:")  
 poezd.data\_otprav = *readLine*()!!.toString()  
 *println*("Введите время отправления:")  
 poezd.time\_otprav = *readLine*()!!.toString()  
 *println*("Введите время в пути:")  
 poezd.time\_way = *readLine*()!!.*toDouble*()  
 } catch (e: Exception) {  
 *println*("Возникло исключение! Повторите ввод данных.")  
 }  
 }  
 fun output ()  
 {  
 *println*("\nПункт назначения - ${punkt\_naznach}\nДата отправления - ${data\_otprav}\nВремя отправления - ${time\_otprav}\nВремя в пути - ${time\_way}")  
 }  
 fun check\_info\_poezdnum(num\_poezd:Int):String  
 {  
 var mes: String = ""  
 if (num\_poezd<=0) {  
 mes = "Номер поезда не может быть меньше нуля!\n"  
 }  
 return mes  
 }  
 fun check\_info\_punkti( punktotprav:String, punktnaznach: String):String  
 {  
 var mes: String = ""  
 for (i in punktotprav)  
 {  
 if ( i !in 'А'..'Я' && i !in 'а'..'я')  
 {  
 mes="Пункт отправления может содержать только буквы русского языка.\n"  
 break  
 }  
 }  
 for (i in punktnaznach)  
 {  
 if ( i !in 'А'..'Я' && i !in 'а'..'я')  
 {  
 mes +="Пункт назначения может содержать только буквы русского языка.\n"  
 break  
 }  
 }  
 return mes  
 }  
 fun check\_info\_timeway(time\_way:Double):String  
 {  
 var mes = ""  
 if (time\_way<=0.0)  
 {  
 mes = "Время в пути не может быть отрицательным."  
 }  
 return mes  
 }  
}

fun main()  
{  
 var flag = true  
 var poezd = poezd()  
 while (flag == true) {  
 poezd.input(poezd)  
 var mes1 = poezd.check\_info\_poezdnum(poezd.num\_poezd)  
 var mes2 = poezd.check\_info\_punkti(poezd.punkt\_otprav,poezd.punkt\_naznach)  
 var mes3 = poezd.check\_info\_timeway(poezd.time\_way)  
 if (mes1 != "") {  
 *println*("Ошибка!\n${mes1}\nПовторите ввод данных")  
  
 }  
 else if (mes2 != "")  
 {  
 *println*("Ошибка!\n${mes2}\nПовторите ввод данных")  
 }  
 else if (mes3 != "")  
 {  
 *println*("Ошибка!\n${mes3}\nПовторите ввод данных")  
 }  
 else  
 flag = false  
 }  
 poezd.output()  
}

**Тестовые ситуации**



**Задание**

**Входные и выходные данные**

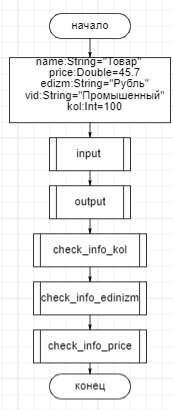
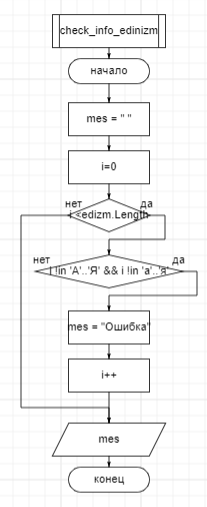
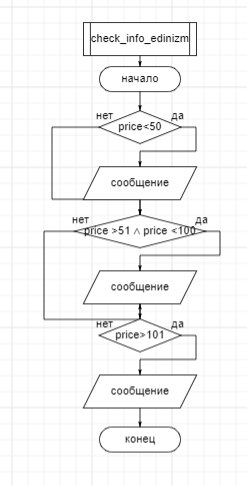
**name –** название товара, строковый тип

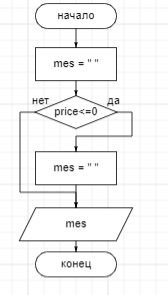
**price -** цена товара, вещественный тип

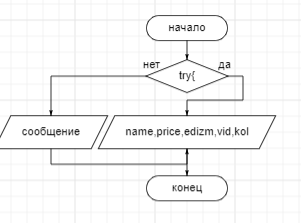
**edizm –** волюта**,** строковый тип

**vid –** вид товара, строковый тип

**kol –** количество товаров, целочисленный тип

**Блок-схема**

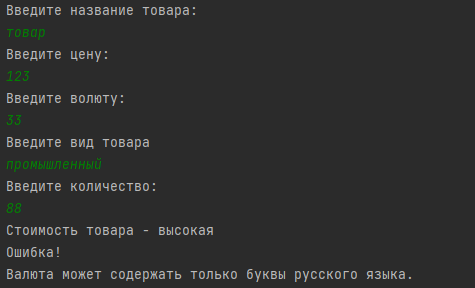
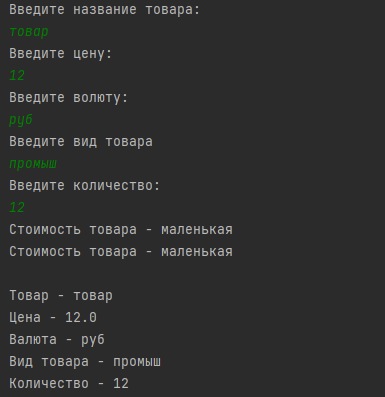




**Листинг программы**

class tovari {  
 var name:String="Товар"  
 var price:Double=45.7  
 var edizm:String="Рубль"  
 var vid:String="Промышенный"  
 var kol:Int=100  
 fun input (tovari: tovari)  
 {  
 try {  
 *println*("Введите название товара:")  
 tovari.name = *readLine*()!!.toString()  
 *println*("Введите цену:")  
 tovari.price = *readLine*()!!.*toDouble*()  
 *println*("Введите волюту:")  
 tovari.edizm = *readLine*()!!.toString()  
 *println*("Введите вид товара")  
 tovari.vid = *readLine*()!!.toString()  
 *println*("Введите количество:")  
 tovari.kol = *readLine*()!!.*toInt*()  
 } catch (e: Exception) {  
 *println*("Возникло исключение! Повторите ввод данных.")  
 }  
 }  
 fun output ()  
 {  
 *println*("\nТовар - ${name}\nЦена - ${price}\nВалюта - ${edizm}\nВид товара - ${vid}\nКоличество - ${kol}")  
 }  
 fun check\_info\_kol(price:Double):String  
 {  
 var mes: String = ""  
 if (price<=0) {  
 mes = "Количество не может быть меньше нуля!\n"  
 }  
 return mes  
 }  
 fun check\_info\_edinizm(edinizm: String):String {  
 var mes: String = ""  
 for (i in edinizm) {  
 if (i !in 'А'..'Я' && i !in 'а'..'я') {  
 mes = "Валюта может содержать только буквы русского языка.\n"  
 break  
 }  
 }  
 return mes  
 }  
 fun check\_info\_price(price: Double) {  
 var mes: String = ""  
 if(price < 50)  
 {  
 *println*("Стоимость товара - маленькая")  
 }  
 else if (price>51 && price<100)  
 *println*("Стоимость товара - средняя")  
 else if (price>101)  
 {  
 *println*("Стоимость товара - высокая")  
 }  
 }  
}

fun main()  
{  
 var flag = true  
 var tovar = tovari()  
 while (flag == true) {  
 tovar.input(tovar)  
 var mes1 = tovar.check\_info\_kol(tovar.price)  
 var mes2 = tovar.check\_info\_edinizm(tovar.edizm)  
 var mes3 = tovar.check\_info\_price(tovar.price)  
 if (mes1 != "") {  
 *println*("Ошибка!\n${mes1}\nПовторите ввод данных")  
  
 }  
 else if (mes2 != "")  
 {  
 *println*("Ошибка!\n${mes2}\nПовторите ввод данных")  
 }  
 else  
 flag = false  
 }  
 tovar.check\_info\_price(tovar.price)  
 tovar.output()  
}

**Тестовые ситуации**