Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое заданятие №9-Б**»

Выполнил: ФИО

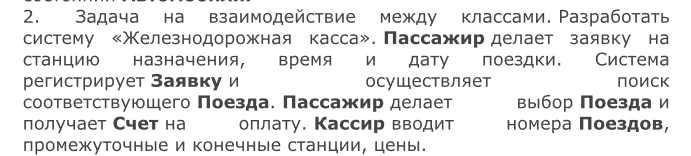
Группа: ПР-22

Преподаватель: Мирошниченко Г.В

2023

**Вариант 2.**

**Задание:**

****

**Входные и выходные данные**

name-имя пассажира, строковый тип

surname-фамилия пассажира, строковый тип

age- возраст пассажира, целочисленный тип

check\_benefits- вспомогательная переменная для определения значения льготы, целочисленный тип

benefits – значение для льготы (да/нет), строковый тип

st\_nazn\_pas- вспомогательная переменная для определения значения станции назначения

timeway-время в пути, вещественный тип

day-день отправления, целочисленный тип

month-месяц оправления, целочисленный тип

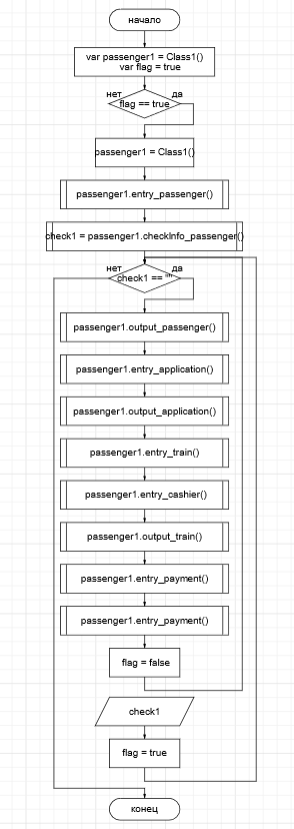
choice1 – вспомогательная переменная для определения типа поезда, целочисленный тип

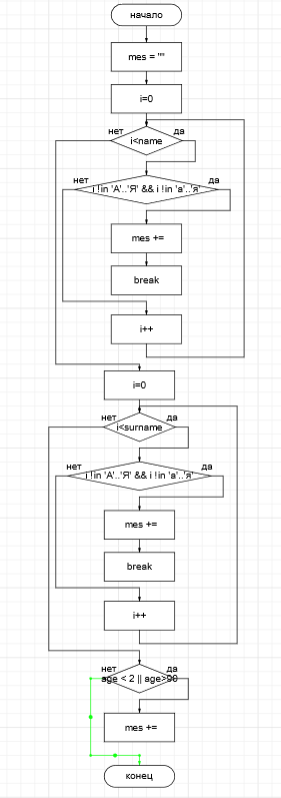
number – номер поезда, вещественный тип

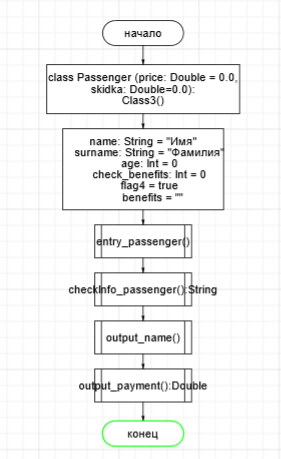
price\_ticket - цена билета, вещественный тип

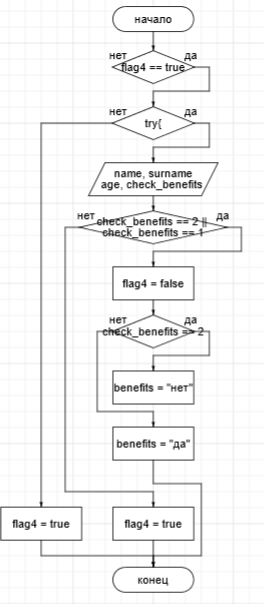
**Блок-схема**

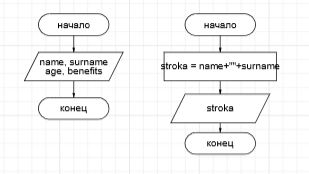
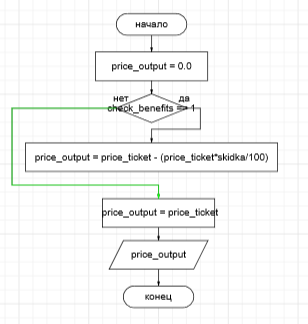
Главная программа

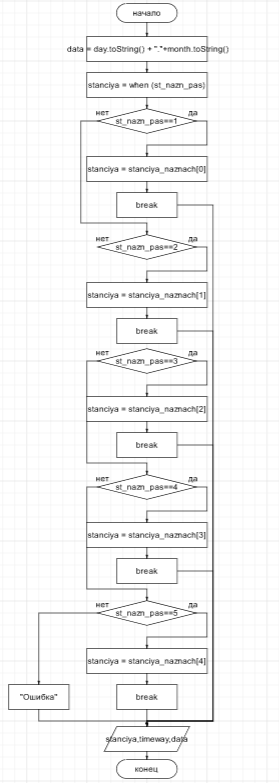


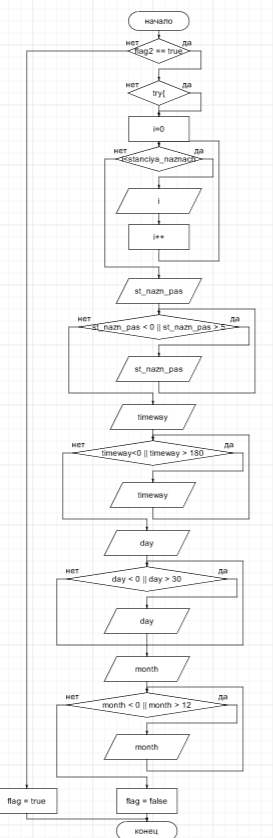
Класс Пассажир

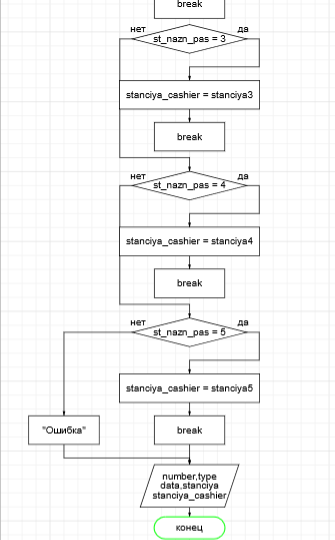


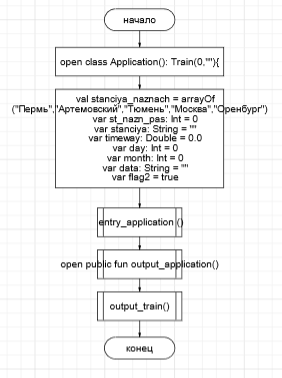




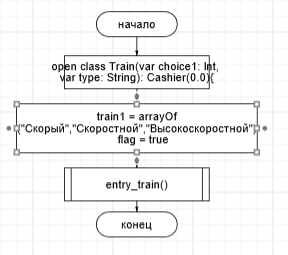
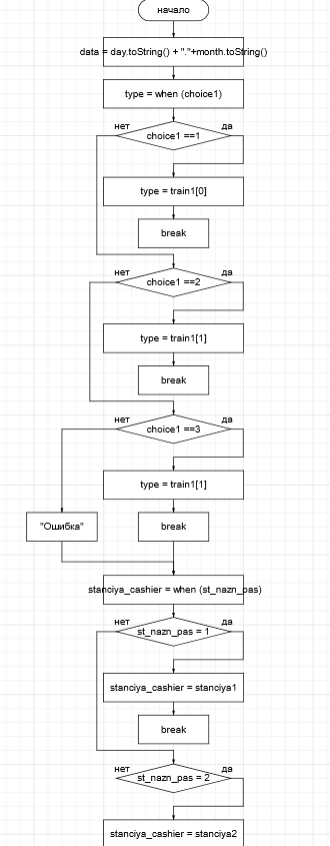
Класс Заявка



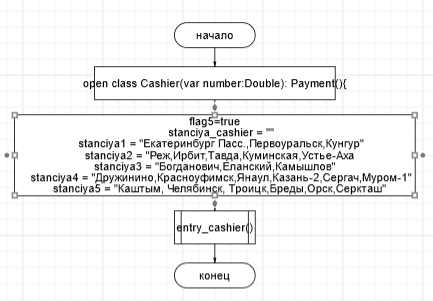
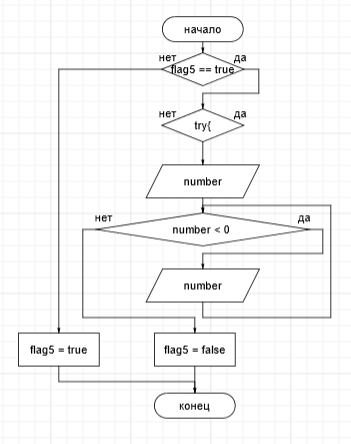




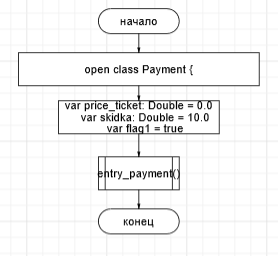
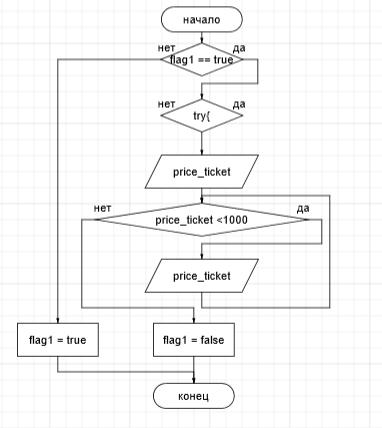
Класс Поезд



Класс Кассир



Класс Счет

****

**Листинг программы (если есть)**

Главная программа

import Passenger as Class1  
fun main()  
{  
 var passenger1 = Class1()  
 var flag = true  
 while (flag == true) {  
 passenger1 = Class1()  
 *println*("Заполнение информации о пассажире:")  
 passenger1.entry\_passenger()  
 var check1 = passenger1.checkInfo\_passenger()  
 if (check1 == "") {  
 *println*("\nИнформация о пассажире:")  
 passenger1.output\_passenger()  
 *println*("\nЗаполение заявки:")  
 passenger1.entry\_application()  
 *println*("\nЗаявка пассажира: ${passenger1.output\_name()}")  
 passenger1.output\_application()  
 *println*("Выбор поезда:")  
 passenger1.entry\_train()  
 passenger1.entry\_cashier()  
 *println*("\nИнформация о поезде:")  
 passenger1.output\_train()  
 *println*("Заполение чека на оплату билета:")  
 passenger1.entry\_payment()  
 passenger1.entry\_payment()  
 *println*("\nИнформация о счете на оплату:")  
 *println*("Стоимость билета, с учетом скидки: "+passenger1.output\_payment())  
 flag = false  
 } else {  
 *println*("Ошибка!${check1}")  
 flag = true  
 }  
 }  
}

Класс Пассажир

import Application as Class3  
class Passenger (price: Double = 0.0, skidka: Double=0.0): Class3(){  
 var name: String = "Имя"  
 var surname: String = "Фамилия"  
 var age: Int = 0  
 var check\_benefits: Int = 0  
 var flag4 = true  
 var benefits = ""  
 public fun entry\_passenger()  
 {  
 while (flag4 == true) {  
 try {  
 *println*("Введите имя (на русском языке):")  
 name = *readLine*()!!.toString()  
 *println*("Введите фамилию (на русском языке):")  
 surname = *readLine*()!!.toString()  
 *println*("Введите количество полных лет пассажира:")  
 age = *readLine*()!!.*toInt*()  
 *println*("Имеется льгота? Введите 1-да, 2-нет")  
 check\_benefits = *readLine*()!!.*toInt*()  
 if (check\_benefits == 2 || check\_benefits == 1) {  
 flag4 = false  
 if (check\_benefits == 2)  
 benefits = "нет"  
 else  
 {  
 benefits = "да"  
 }  
 }  
 else  
 {  
 *println*("Ошибка! Повторите ввод.")  
 flag4 = true  
 }  
 }  
 catch (e:Exception)  
 {  
 *println*("Возникло исключение! Повторите ввод.")  
 flag4 = true  
 }  
 }  
 }  
 public fun checkInfo\_passenger():String  
 {  
 var mes = ""  
 for (i in name) {  
 if (i !in 'А'..'Я' && i !in 'а'..'я') {  
 mes += "\nИмя пассажира должно содержать только буквы русского языка\n"  
 break  
 }  
 }  
 for (i in surname) {  
 if (i !in 'А'..'Я' && i !in 'а'..'я') {  
 mes += "Фамилия пассажира должно содержать только буквы русского языка\n"  
 break  
 }  
 }  
 if (age < 2 || age>90) {  
 mes += "Возраст пассажира поезда не может быть: меньше 2 лет и больше 90 \n"  
 }  
 return mes  
 }  
 public fun output\_passenger()  
 {  
 *println*("Имя: ${name}\nФамилия: ${surname}\nВозраст: ${age}\nЛьгота: ${benefits}")  
 }  
 public fun output\_name():String  
 {  
 var stroka = name+""+surname  
 return stroka  
 }  
 open public fun output\_payment():Double  
 {  
 var price\_output = 0.0  
 if (check\_benefits == 1)  
 {  
 price\_output = price\_ticket - (price\_ticket\*skidka/100)  
 }  
 else  
 price\_output = price\_ticket  
 return price\_output  
 }  
}

Класс Заявка

import Train  
open class Application(): Train(0,""){  
 val stanciya\_naznach = *arrayOf*("Пермь","Артемовский","Тюмень","Москва","Оренбург")  
 var st\_nazn\_pas: Int = 0  
 var stanciya: String = ""  
 var timeway: Double = 0.0  
 var day: Int = 0  
 var month: Int = 0  
 var data: String = ""  
 var flag2 = true  
 open public fun entry\_application ()  
 {  
 while (flag2 == true) {  
 try {  
 *println*("Выберите нужную станцию (Введите число от 1 до 5)")  
 for (i in stanciya\_naznach)  
 {  
 *print*(i+" ")  
 }  
 *println*()  
 st\_nazn\_pas = *readLine*()!!.*toInt*()  
 while (st\_nazn\_pas < 0 || st\_nazn\_pas > 5)  
 {  
 *println*("Ошибка ввода. Повторите попытку:")  
 st\_nazn\_pas = *readLine*()!!.*toInt*()  
 }  
 *println*("Введите время в дороге (в минутах)")  
 timeway = *readLine*()!!.*toDouble*()  
 while (timeway<0 || timeway > 180)  
 {  
 *println*("Ошибка ввода. Повторите попытку.")  
 timeway = *readLine*()!!.*toDouble*()  
 }  
 *println*("Введите день отправления.")  
 day = *readLine*()!!.*toInt*()  
 while (day < 0 || day > 30)  
 {  
 *println*("Ошибка ввода. Повторите попытку.")  
 day = *readLine*()!!.*toInt*()  
 }  
 *println*("Введите месяц отправления.")  
 month = *readLine*()!!.*toInt*()  
 while (month < 0 || month > 12)  
 {  
 *println*("Ошибка ввода. Повторите попытку.")  
 month = *readLine*()!!.*toInt*()  
 }  
 flag2 = false  
 }  
 catch (e:Exception)  
 {  
 *println*("Возникло исключение! Повторите ввод.")  
 flag2 = true  
 }  
 }  
 }  
 open public fun output\_application() {  
 data = day.toString() + "."+month.toString()  
 stanciya = when (st\_nazn\_pas)  
 {  
 1 -> stanciya\_naznach[0]  
 2 -> stanciya\_naznach[1]  
 3 -> stanciya\_naznach[2]  
 4 -> stanciya\_naznach[3]  
 5 -> stanciya\_naznach[4]  
 else  
 -> "Ошибка"  
 }  
  
 *println*("Станция назначения: ${stanciya}\nВремя в пути: ${timeway}\nДата: ${data}\n")  
 }  
 public fun output\_train()  
 {  
 data = day.toString() + "."+month.toString()  
 type = when (choice1)  
 {  
 1 -> train1[0]  
 2 -> train1[1]  
 3 -> train1[2]  
 else  
 -> "Ошибка"  
 }  
 stanciya\_cashier = when (st\_nazn\_pas)  
 {  
 1 -> stanciya1  
 2 -> stanciya2  
 3 -> stanciya3  
 4 -> stanciya4  
 5 -> stanciya5  
 else  
 -> "Ошибка"  
 }  
 *println*("Номер поезда: ${number}\nТип поезда: ${type}\nДата: ${data}\nКонечная станция: ${stanciya}\nПромежуточные станции:\n${stanciya\_cashier}\n")  
 }  
}

Класс Поезд

open class Train(var choice1: Int, var type: String): Cashier(0.0,0.0){  
 val train1 = *arrayOf*("Скорый","Скоростной","Высокоскоростной")  
 var flag = true  
  
 open public fun entry\_train() {  
 while (flag == true) {  
 try {  
 *println*("Введите нужный тип поезда (число от 1 до 3)")  
 for (i in train1)  
 {  
 *print*(i+" ")  
 }  
 *println*()  
 choice1 = *readLine*()!!.*toInt*()  
 while (choice1 < 1 && choice1>3)  
 {  
 *println*("Возникла ошибка! Повторите ввод.")  
 choice1 = *readLine*()!!.*toInt*()  
 }  
 flag= false  
 } catch (e: Exception) {  
 *println*("Возникло исключение! Повторите ввод.")  
 flag= true  
 }  
 }  
 }  
}

Класс Кассир

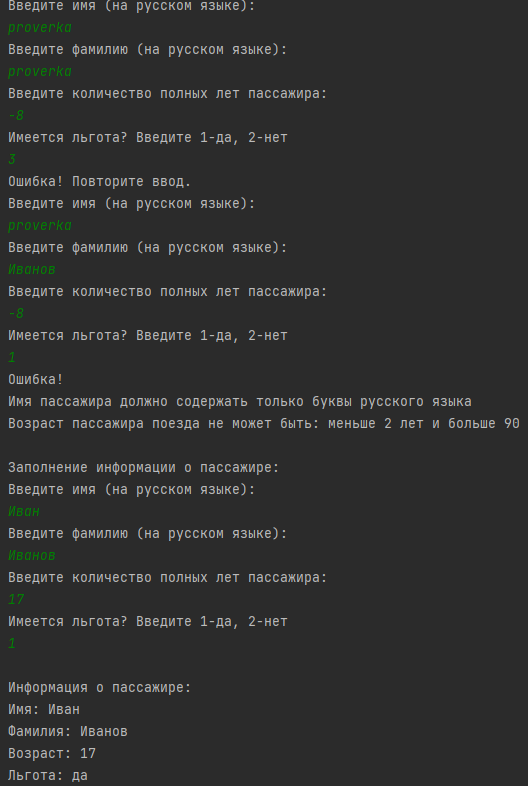
import Payment  
open class Cashier(var number:Double): Payment(){  
 var flag5=true  
 var stanciya\_cashier = ""  
 var stanciya1 = "Екатеринбург Пасс.,Первоуральск,Кунгур"  
 var stanciya2 = "Реж,Ирбит,Тавда,Куминская,Устье-Аха"  
 var stanciya3 = "Богданович,Еланский,Камышлов"  
 var stanciya4 = "Дружинино,Красноуфимск,Янаул,Казань-2,Сергач,Муром-1"  
 var stanciya5 = "Каштым, Челябинск, Троицк,Бреды,Орск,Серкташ"  
 open public fun entry\_cashier()  
 {  
 while (flag5 == true) {  
 try {  
 *println*("Введите номер поезда:")  
 number = *readLine*()!!.*toDouble*()  
 while (number < 0 ) {  
 *println*("Возникла ошибка! Повторите ввод.")  
 number = *readLine*()!!.*toDouble*()  
 }  
 flag5 = false  
 } catch (e: Exception) {  
 *println*("Возникло исключение! Повторите ввод.")  
 flag5 = true  
 }  
 }  
 }  
}

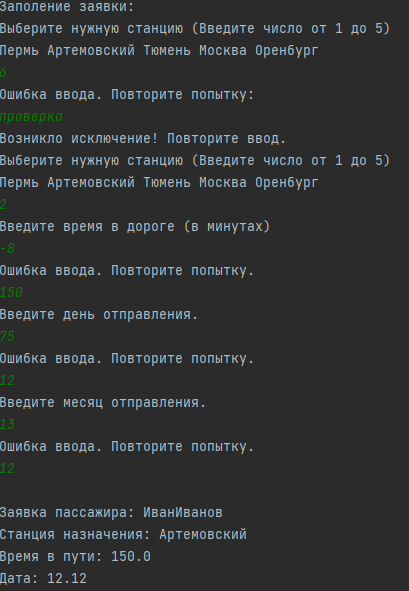
Класс Счет

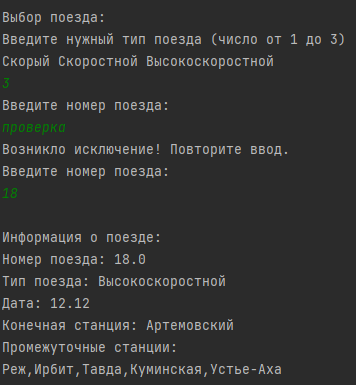
open class Payment {  
 var price\_ticket: Double = 0.0  
 var skidka: Double = 10.0  
 var flag1 = true  
 open public fun entry\_payment() {  
 while (flag1 == true) {  
 try {  
 *println*("Введите стоимость билета (больше 1000)")  
 price\_ticket = *readLine*()!!.*toDouble*()  
 while (price\_ticket <1000)  
 {  
 price\_ticket = *readLine*()!!.*toDouble*()  
 }  
 flag1 = false  
 } catch (e: Exception) {  
 *println*("Возникло исключение! Повторите ввод.")  
 flag1 = true  
 }  
 }  
 }  
}

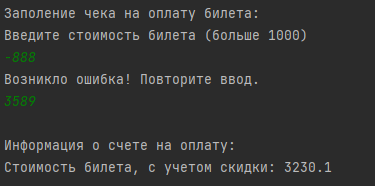
**Тестовые ситуации**

Выполнены проверки на корректность вводимый значений: строковые переменные должны содержать только буквы русского языка, определены максимальные и минимальные значения для вещественных и целочисленных переменных.







****Цена билета считается так, что если у пассажира есть льгота, то действует скидка 10%, если же льготы нет цена билета не меняется.