Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Котлин**»

Выполнил: Лобанов Глеб Артемович

Группа: ПР-22

Преподаватель: Мирошниченко Г.В

2023

**Задание: **

**Входные и выходные данные:**

**Marka(string)**

**Place\_otp(string)**

**Place\_nazn(string)**

**Date\_otpr(string)**

**Time\_otpr(string)**

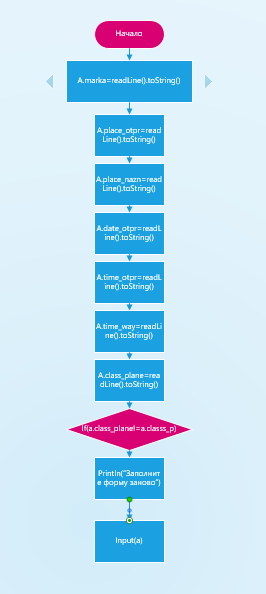
**Time\_way(string)**

**Class\_plane(string)**

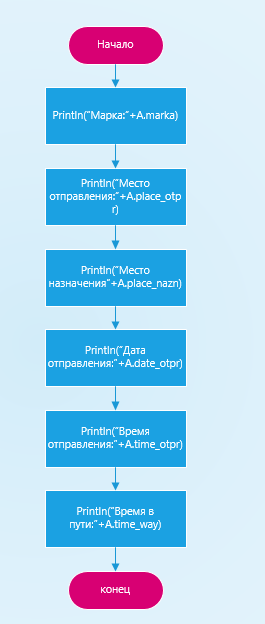
**Classs\_p(string)**

**Блок-схема:**

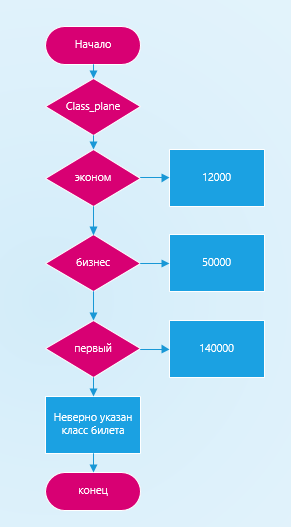
**Функция №1 Input**

****

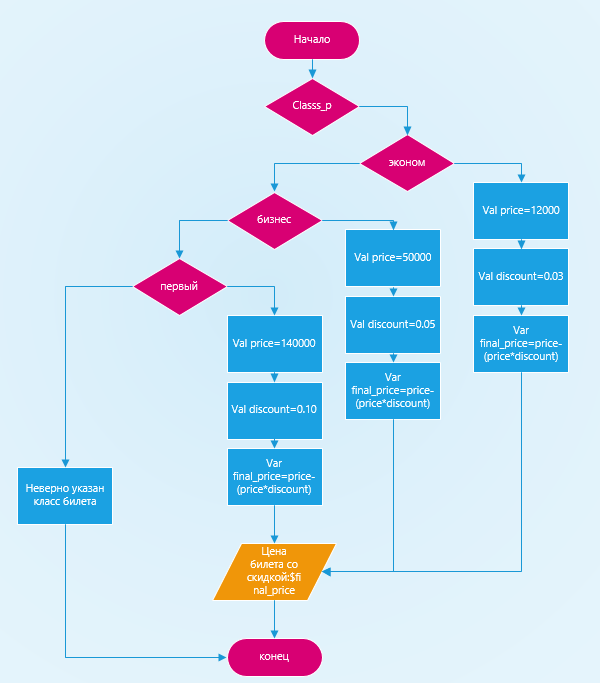
**Функция №2 Give\_Info**

****

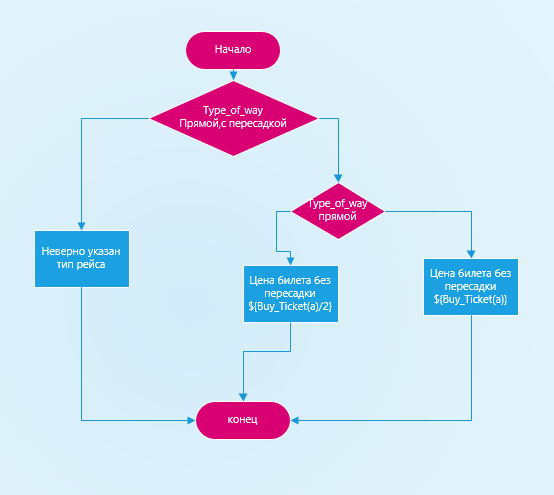
**Функция №3 Buy\_Ticket**

****

**Функция №4 Discount\_Ticket**

****

**Функция №5 Type\_Of\_Reys**

****

**Листинг программы:**

class Plane (var marka: String,  
 var place\_otpr: String,  
 var place\_nazn: String,  
 var date\_otpr: String,  
 var time\_otpr: String,  
 var time\_way: String,  
 var class\_plane:String,  
 var classs\_p: String,  
 var type\_of\_way: String  
 ){  
 fun Input(a:Plane){  
 a.marka= *readLine*().*toString*()  
 a.place\_otpr= *readLine*().*toString*()  
 a.place\_nazn= *readLine*().*toString*()  
 a.date\_otpr= *readLine*().*toString*()  
 a.time\_otpr= *readLine*().*toString*()  
 a.time\_way= *readLine*().*toString*()  
 a.class\_plane= *readLine*().*toString*()  
 *println*("Подтвердите класс, введя его еще раз")  
 a.classs\_p= *readLine*().*toString*()  
 if(a.class\_plane!=a.classs\_p) {  
 *println*("Заполните форму заново")  
 Input(a)  
 }  
 }  
  
 fun Give\_Info(a: Plane){  
 *println*("Марка: "+a.marka)  
 *println*("Место отправления: "+a.place\_otpr)  
 *println*("Место назначения: "+a.place\_nazn)  
 *println*("Дата отправления: "+a.date\_otpr)  
 *println*("Время отправления: "+a.time\_otpr)  
 *println*("Время в пути: "+a.time\_way)  
 }  
  
 fun Buy\_Ticket(a:Plane): Int {  
 return when (class\_plane) {  
 "эконом" -> 12000  
 "бизнес" -> 50000  
 "первый" -> 140000  
 else -> {  
 *println*("Неверно указан класс билета")  
 1  
 }  
 }  
 }  
 fun Discount\_Ticket(a:Plane) {  
 when (classs\_p) {  
 "эконом" -> {  
 val price = 12000  
 val discount = 0.03  
 var final\_price = price - (price \* discount)  
 *println*("Цена билета со скидкой: $final\_price")  
 }  
  
 "бизнес" -> {  
 val price = 50000  
 val discount = 0.05  
 var final\_price = price - (price \* discount)  
 *println*("Цена билета со скидкой: $final\_price")  
 }  
  
 "первый" -> {  
 val price = 140000  
 val discount = 0.10  
 var final\_price = price - (price \* discount)  
 *println*("Цена билета со скидкой: $final\_price")  
 }  
 else -> *println*("Неверно указан класс билета")  
 }  
 }  
  
 fun Type\_Of\_Reys(a: Plane) {  
 when (type\_of\_way) {  
 "прямой", "с пересадкой" -> {  
 when (type\_of\_way) {  
 "прямой" -> *println*("Цена билета без пересадок: ${Buy\_Ticket(a)}")  
 "с пересадкой" -> *println*("Цена билета с пересадкой: ${Buy\_Ticket(a)/2}")  
 }  
 }  
 else -> {  
 *println*("Неверно указан тип рейса")  
 Input(a)  
 }  
 }  
 }  
}

**Тестовые ситуации:**

**Программа не выводит цену билета с пересадкой**