Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Классы и объекты в Kotlin. Практическое занятие 8 (задание 1)**»

Выполнил: Максимов Кирилл Эдуардович

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Задание:**

**Входные и выходные данные**

**Блок-схема**

**Листинг программы (если есть)**

**Тестовые ситуации**

**Вывод**

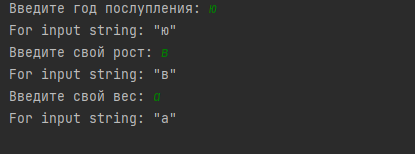
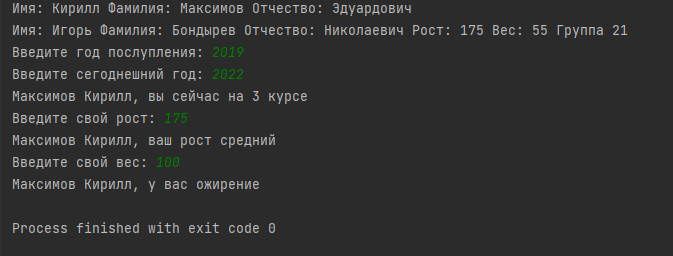
Вариант 9. Класс «Студент»: имя, отчество, фамилия, пол, группа, дата рождения, рост, вес, вид спорта.

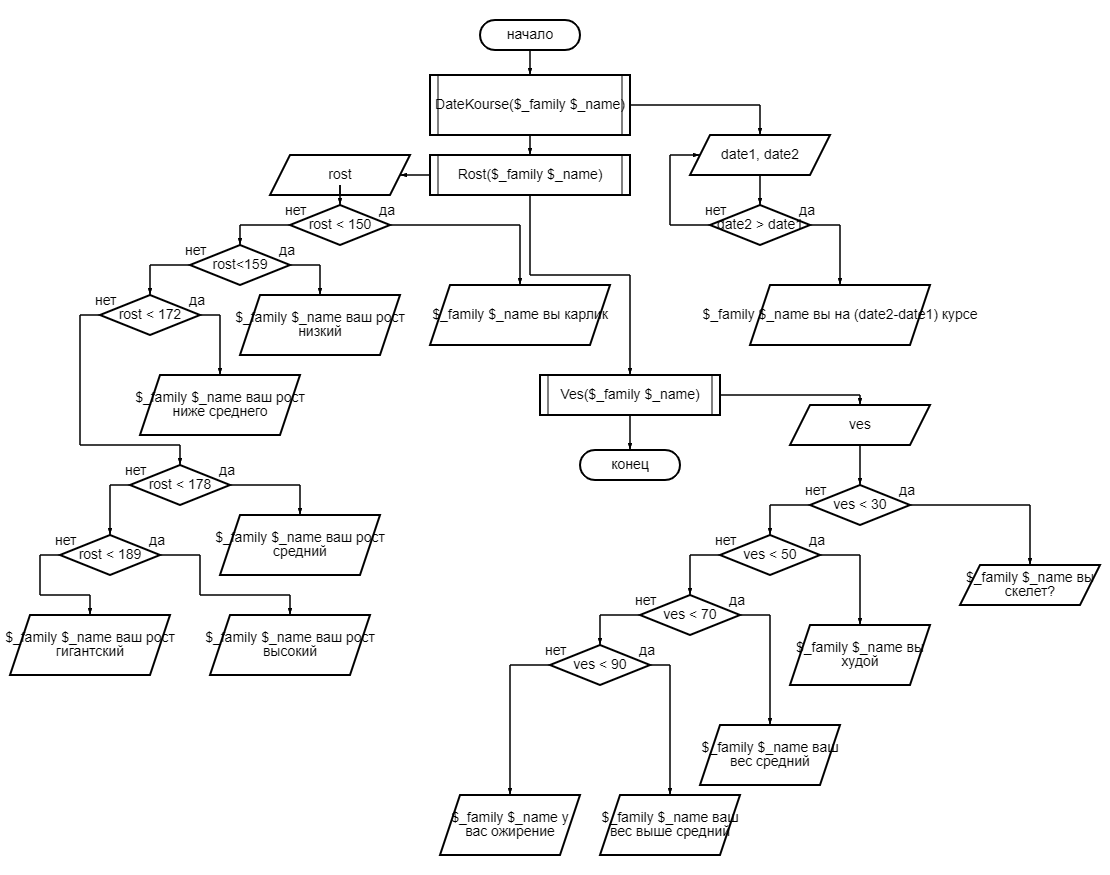
Входные данные: name, family, otchestvo, pol, data, sport – string  
 group, rost, ves – int

Выходные данные: println - string

class Student(\_name: String)  
{  
 var name = \_name  
 var family = "Undefiend"  
 var otchestvo = "Undefiend"  
 var pol = "Undefiend"  
 var group = 0  
 var data = "Undefiend"  
 var rost = 0  
 var ves = 0  
 var sport = "Undefiend"  
  
 constructor(\_name: String, \_family: String, \_otchestvo: String): this(\_name) {  
 family = \_family  
 otchestvo = \_otchestvo  
 }  
  
 constructor(\_name: String, \_family: String, \_otchestvo: String, \_rost: Int, \_ves: Int, \_group: Int) : this(\_name) {  
 family = \_family  
 otchestvo = \_otchestvo  
 rost = \_rost  
 ves = \_ves  
 group = \_group  
 }  
  
 fun DateKourse(\_family: String, \_name: String)  
 {  
 try {  
 var data1 : Int  
 var data2 : Int  
 do {  
 *print*("Введите год поступления: ")  
 data1 = *readLine*()!!.*toInt*()  
 *print*("Введите сегоднешний год: ")  
 data2 = *readLine*()!!.*toInt*()  
 }while(data1 > data2)  
  
 *println*("$\_family $\_name, вы сейчас на ${data2-data1} курсе")  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
 }  
  
 fun Rost(\_family: String, \_name: String)  
 {  
 try {  
 *print*("Введите свой рост: ")  
 var rost = *readLine*()!!.*toInt*()  
  
 when{  
 (rost < 150) -> *println*("$\_family $\_name, вы карлик")  
 (rost < 159) -> *println*("$\_family $\_name, ваш рост низкий")  
 (rost < 172) -> *println*("$\_family $\_name, ваш рост ниже среднего")  
 (rost < 178) -> *println*("$\_family $\_name, ваш рост средний")  
 (rost < 189) -> *println*("$\_family $\_name, ваш рост высокий")  
 else -> *println*("$\_family $\_name, ваш рост гигантский")  
 }  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
  
 }  
  
 fun Ves(\_family: String, \_name: String)  
 {  
 try {  
 *print*("Введите свой вес: ")  
 var ves = *readLine*()!!.*toInt*()  
  
 when{  
 (ves < 30) -> *println*("$\_family $\_name, вы скелет?")  
 (ves < 50) -> *println*("$\_family $\_name, вы худой")  
 (ves < 70) -> *println*("$\_family $\_name, ваш вес средний")  
 (ves < 90) -> *println*("$\_family $\_name, ваш вес выше среднего")  
 else -> *println*("$\_family $\_name, у вас ожирение")  
 }  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
  
 }  
}

fun main()  
{  
 try {  
 val studentone = Student("Кирилл", "Максимов", "Эдуардович")  
  
 val studenttwo = Student("Игорь", "Бондырев", "Николаевич", 175, 55, 21)  
  
 *println*("Имя: ${studentone.name} Фамилия: ${studentone.family} Отчество: ${studentone.otchestvo}")  
  
 *println*("Имя: ${studenttwo.name} Фамилия: ${studenttwo.family} Отчество: ${studenttwo.otchestvo} " +  
 "Рост: ${studenttwo.rost} Вес: ${studenttwo.ves} Группа ${studenttwo.group}")  
  
 studentone.DateKourse(studentone.family, studentone.name)  
 studentone.Rost(studentone.family, studentone.name)  
 studentone.Ves(studentone.family, studentone.name)  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
  
}



Вариант 12. Класс «Кинотеатр»: название кинофильма, сеанс, стоимость билета, зал, количество зрителей.

Входные данные: namefilm, seans – string  
 moneyticket, numberzal, kolvoWatcher – int

Выходные данные: println - string

class KinoTeatr(\_namefilm: String)  
{  
 var namefilm = \_namefilm  
 var seans = "Undefiend"  
 var moneyticket = 0  
 var numberzal = 0  
 var kolvoWatcher = 0  
  
 constructor(\_namefilm: String, \_seans: String, \_moneyticket: Int): this(\_namefilm) {  
 seans = \_seans  
 moneyticket = \_moneyticket  
 }  
  
 constructor(\_namefilm: String, \_seans: String, \_moneyticket: Int, \_numberzal : Int, \_kolvoWatcher: Int) : this(\_namefilm) {  
 seans = \_seans  
 moneyticket = \_moneyticket  
 numberzal = \_numberzal  
 kolvoWatcher = \_kolvoWatcher  
 }  
  
 fun Seans()  
 {  
 try {  
 var time : Int  
 do {  
 *print*("Введите час сеанса: ")  
 time = *readLine*()!!.*toInt*()  
 }while (time < 0 || time > 24)  
  
  
 when{  
 (time > 0 && time <= 4) -> *println*("Ночной сеанс")  
 (time > 4 && time <= 11) -> *println*("Утренний сеанс")  
 (time > 11 && time <= 16) -> *println*("Дневной сеанс")  
 (time > 16 && time <= 24) -> *println*("Вечерний сеанс")  
 }  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
 }  
  
 fun Zal()  
 {  
 try {  
 var mestall : Int  
 var zanyat: Int  
 do{  
 *print*("Кол-во занятых мест должно быть меньше мест в зале")  
 *print*("Введите мест в зале: ")  
 mestall = *readLine*()!!.*toInt*()  
 *print*("Введите кол-во занятых мест: ")  
 zanyat = *readLine*()!!.*toInt*()  
 }while (zanyat > mestall)  
  
 *println*("В зале свобожных мест ${mestall - zanyat} ")  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
 }  
  
 fun Money()  
 {  
 try {  
 *print*("Введите сколько у вас денег: ")  
 var MyMoney = *readLine*()!!.*toInt*()  
 *print*("Введите цену билета: ")  
 var moneyTicket = *readLine*()!!.*toInt*()  
 *print*("Введите кол-во билетов: ")  
 var kolvoTicket = *readLine*()!!.*toInt*()  
  
 when{  
 ((moneyTicket \* kolvoTicket) < MyMoney) -> *println*("Вы успешно купили билеты")  
 else -> *println*("Ваш не хватает денег")  
 }  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
 }  
  
  
}

fun main()  
{  
 try {  
 val kinofilmOne = KinoTeatr("Человек паук: возвращение домой", "Вечерний", 300)  
  
 val kinofilmTwo = KinoTeatr("Перевозчик", "Ночной", 500, 4, 50)  
  
 *println*("Название: ${kinofilmOne.namefilm} Сеанс: ${kinofilmOne.seans} Стоимость: ${kinofilmOne.moneyticket}")  
  
 *println*("Название: ${kinofilmTwo.namefilm} Сеанс: ${kinofilmTwo.seans} Стоимость: ${kinofilmTwo.moneyticket}" +  
 " Номер зала: ${kinofilmTwo.numberzal} Кол-во зрителей: ${kinofilmTwo.kolvoWatcher}")  
  
 kinofilmOne.Seans()  
 kinofilmOne.Zal()  
 kinofilmOne.Money()  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
}

