Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Классы перечислений (enum).**»

Выполнил: Плоских Александр Павлович

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2024

**Задание:**

1. Сделать конспект и выполнить задание (создать по 2 класса enum).  
2. Добавьте по 4 метода с каждым классом.  
3. Сделать отчет  
4. Проект и отчет выложить в репрозиторий с именем prakt11\_FI и ссылку выложить в гугл-класс.

**Входные и выходные данные**

Класс Fruits

s-String,переменная в которую вводится какой фрукт хотят купить

Класс Season

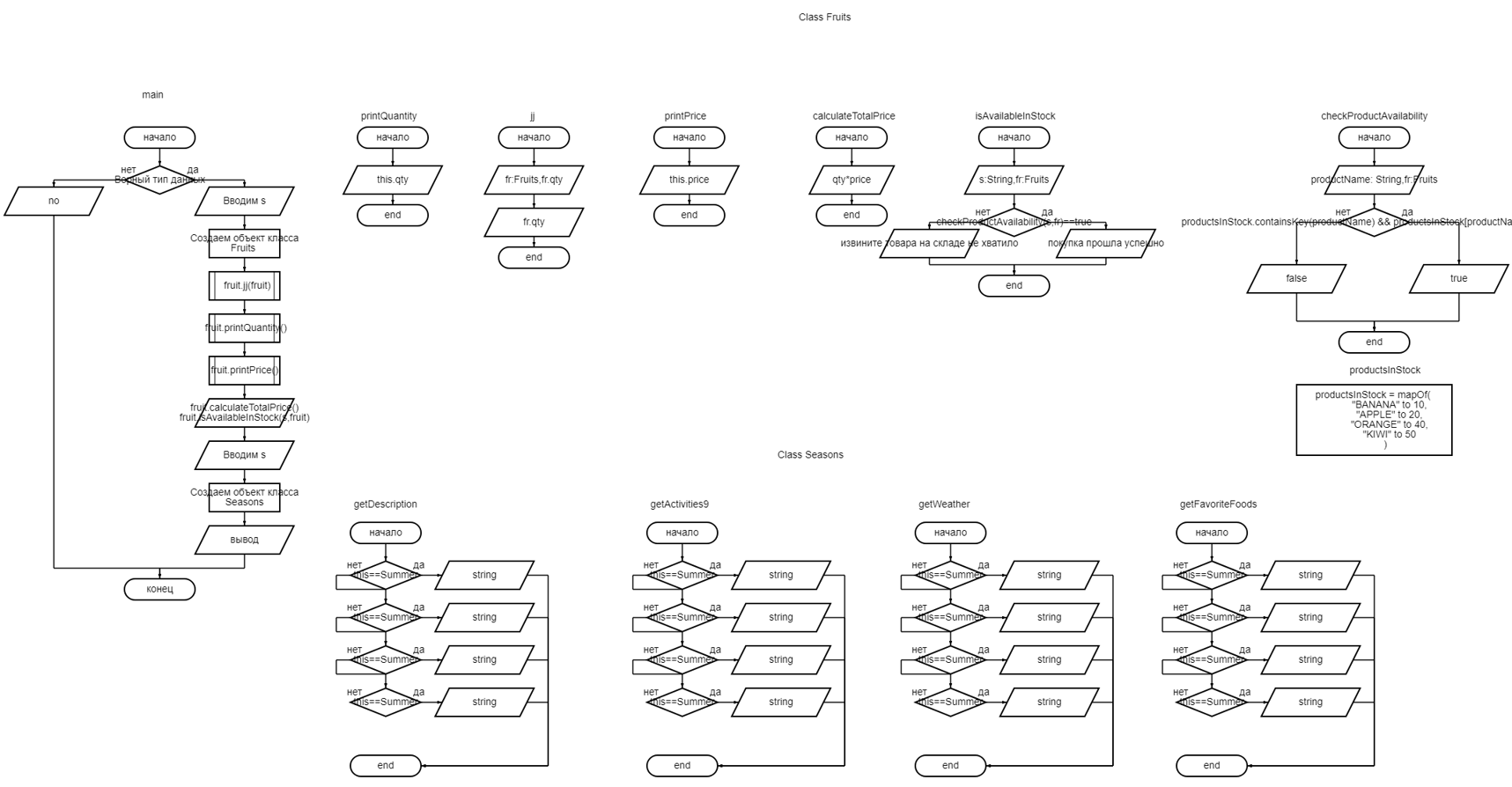
s-String,переменная в которую вводится про какой месяц хотят узнать информацию

**Выходные**

Класс Fruits

Класс Season

**Блок-схема**

****

**Листинг программы**

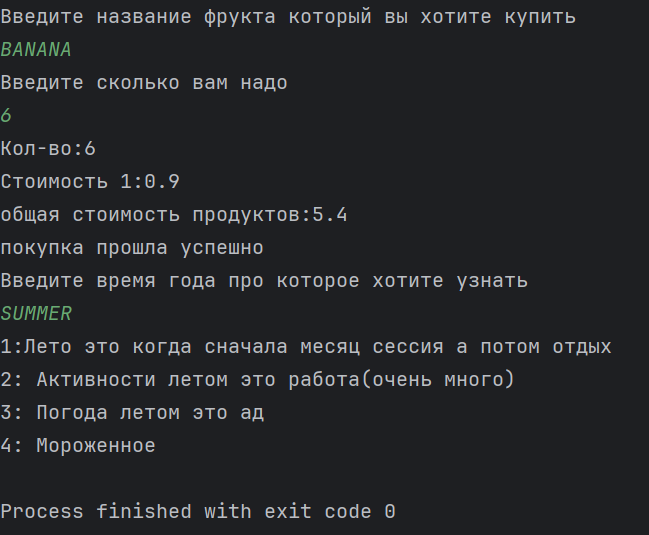
Main

//zadanie 2  
try {  
 *println*("Введите название фрукта который вы хотите купить")  
 var s= *readln*()  
 val fruit= Fruits.valueOf(s)  
 fruit.jj(fruit)  
 fruit.printQuantity()  
 fruit.printPrice()  
 *println*("общая стоимость продуктов:"+fruit.calculateTotalPrice())  
 *println*(fruit.isAvailableInStock(s,fruit))  
 //z2.2  
 *println*("Введите время года про которое хотите узнать")  
 s= *readln*()  
 *println*("1:${Season.valueOf(s).getDescription()}")  
 *println*("2: ${Season.valueOf(s).getActivities()}")  
 *println*("3: ${Season.valueOf(s).getWeather()}")  
 *println*("4: ${Season.valueOf(s).getFavoriteFoods()}")  
}catch (e:Exception){  
 *println*("no")  
}

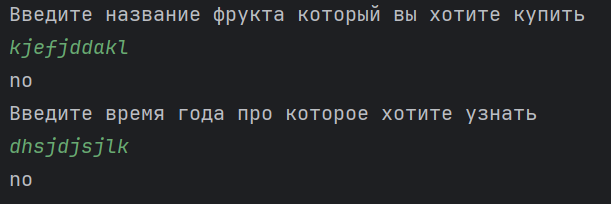
Классы

enum class Fruits(var qty: Int, var price: Double) {  
 *BANANA*(10, 0.9), *APPLE*(12, 1.1), *ORANGE*(5, 1.5), *KIWI*(15, 1.8);  
  
 fun printQuantity() {  
 *println*("Кол-во:"+this.qty)  
 }  
  
 fun jj(fr:Fruits):Int{  
 *println*("Введите сколько вам надо")  
 fr.qty= *readln*()!!.*toInt*()  
 return fr.qty  
 }  
  
 fun printPrice() {  
 *println*("Стоимость 1:"+this.price)  
 }  
  
 fun calculateTotalPrice(): Double {  
 return qty \* price  
 }  
  
 fun isAvailableInStock(s:String,fr:Fruits): String {  
 if(checkProductAvailability(s,fr)==true)  
 return "покупка прошла успешно"  
 else return "извините товара на складе не хватило"  
 }  
  
  
 companion object {  
 private var productsInStock = *mapOf*(  
 "BANANA" *to* 10,  
 "APPLE" *to* 20,  
 "ORANGE" *to* 40,  
 "KIWI" *to* 50  
 )  
  
 @JvmStatic  
 fun checkProductAvailability(productName: String,fr:Fruits): Boolean {  
 if(productsInStock.containsKey(productName) && productsInStock[productName]!! >fr.qty ){  
 return true  
 }  
 else return false  
 }  
 }  
}  
  
enum class Season {  
 *SPRING*, *SUMMER*, *AUTUMN*, *WINTER*;  
  
 fun getDescription(): String {  
 return when (this) {  
 *SPRING* -> "Весна это когда много грязи"  
 *SUMMER* -> "Лето это когда сначала месяц сессия а потом отдых"  
 *AUTUMN* -> "Осень это когда дети плачут"  
 *WINTER* -> "Зима это когда дед мороз вламывается в чужие дома"  
 }  
 }  
  
 fun getActivities(): String {  
 return when (this) {  
 *SPRING* -> "Активности весной это аллергия"  
 *SUMMER* -> "Активности летом это работа(очень много)"  
 *AUTUMN* -> "Активности осенью это очень много плакать"  
 *WINTER* -> "Активности зимой это убирать снег(но только тем кто живет в домах за городом)"  
 }  
 }  
  
 fun getWeather(): String {  
 return when (this) {  
 *SPRING* -> "Погода весной это потоп"  
 *SUMMER* -> "Погода летом это ад"  
 *AUTUMN* -> "Погода осенью это не понятно что "  
 *WINTER* -> "Погода зимой это 40 слоев одежды"  
 }  
 }  
  
 fun getFavoriteFoods(): String {  
 return when (this) {  
 *SPRING* -> "Я не знаю"  
 *SUMMER* -> "Мороженное"  
 *AUTUMN* -> "Я не знаю"  
 *WINTER* -> "Салатики"  
 }  
 }  
  
}

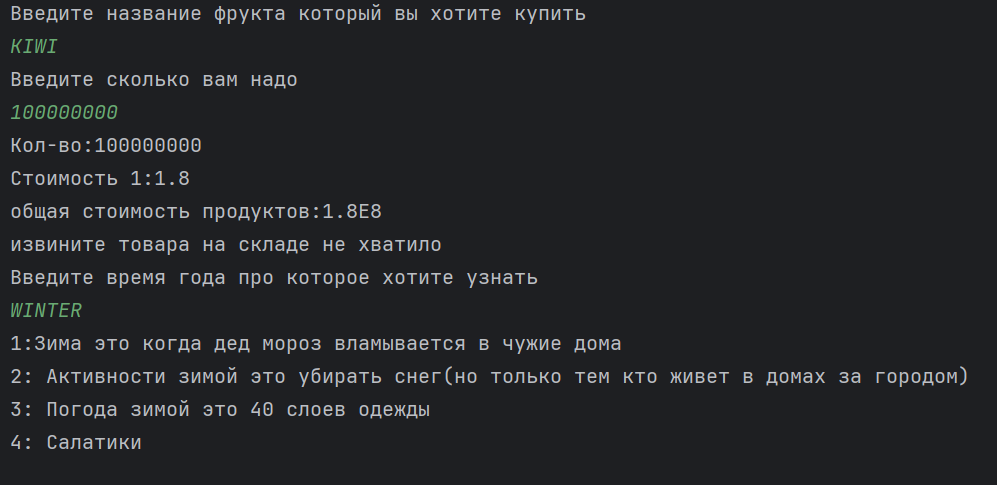
**Тестовые ситуации**

****

На этом скриншоте показано что происходит если неправильно ввести данные

****

На этом скриншоте показано что будет написано если товара на складе не хватит

****

**Вывод**

Выполняя эту работу, я изучил, как правильно использовать классы пересеслений в kotlin,

также обновил навыки использования классов в kotlin.