Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практика 11**»

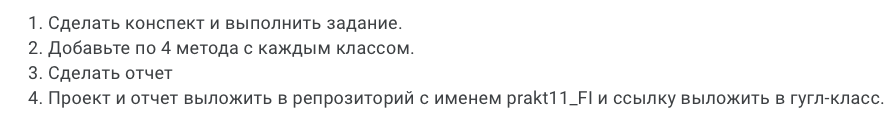
Выполнил: Молотова Виктория Анатольевна

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

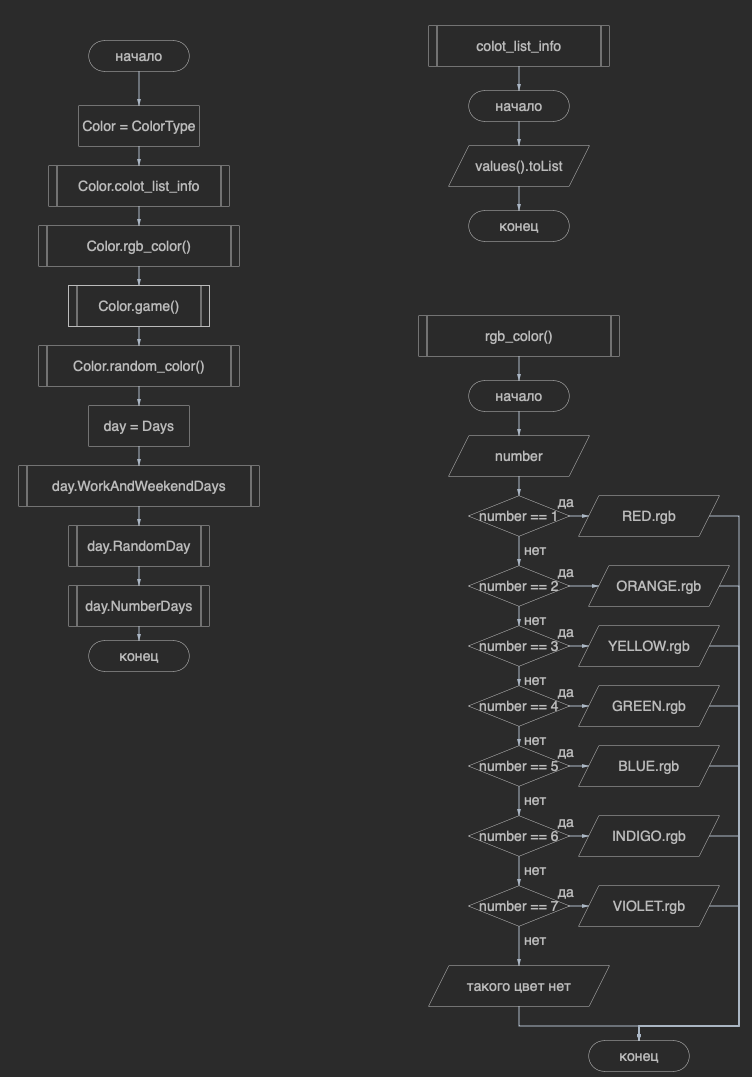
**Задание:**

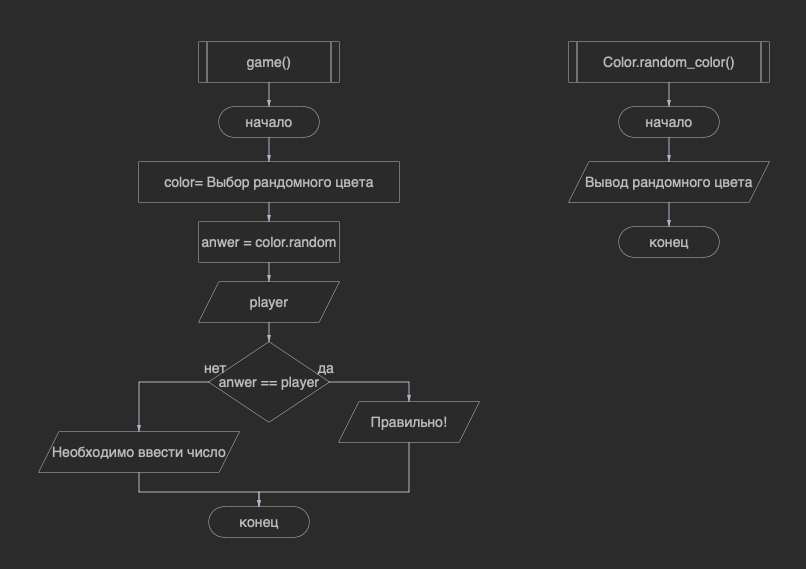
****

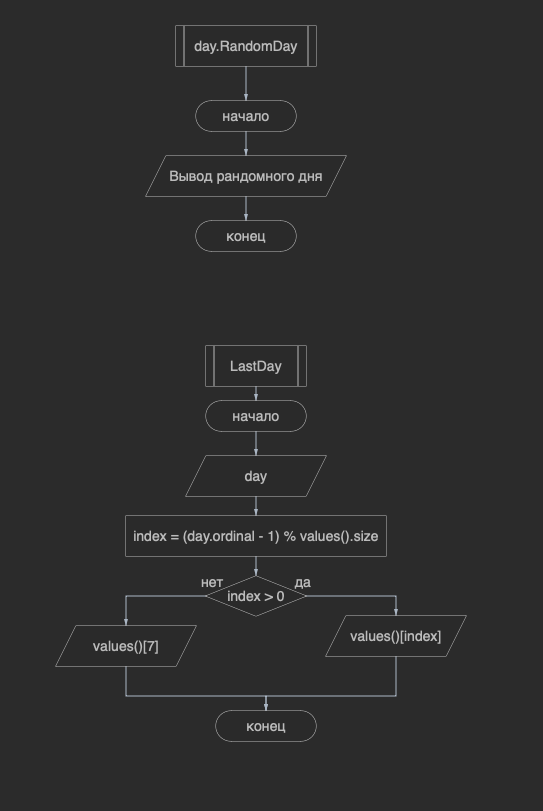
**Входные и выходные данные**

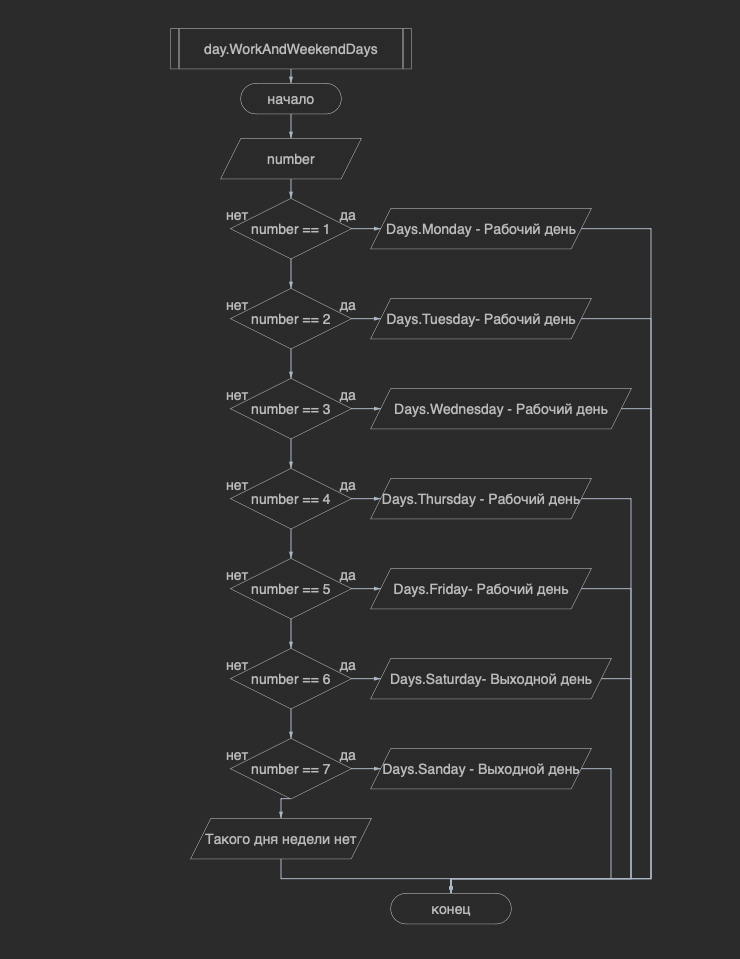
* *Входные данные:* val number :Int, val player :Int, val number: Int,
* *Выходные данные* val day: Days, val index: Int, text:String

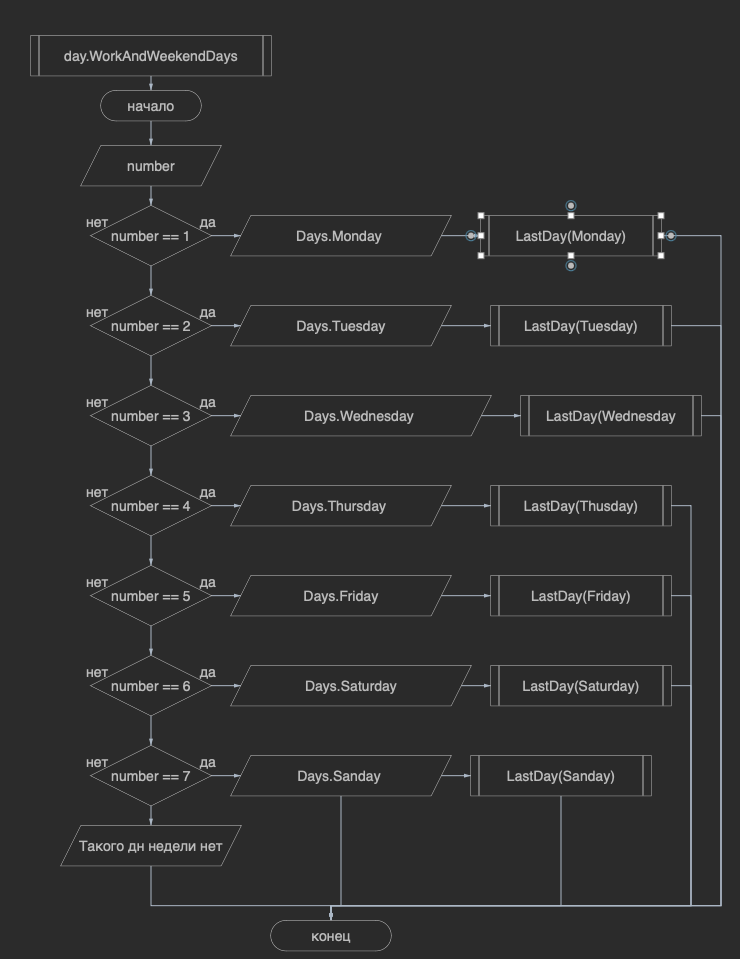
**Блок-схема**

****

****

****

****

****

**Листинг программы**

//main.kt  
fun main() {  
 val Color: ColorType = ColorType.*RED  
 println*(Color.colot\_list\_info())  
 Color.rgb\_color()  
 Color.game()  
 Color.random\_color()  
  
 val day: Days = Days.*Monday* day.WorkAndWeekendDays()  
 day.RandomDay()  
 day.NumberDays()  
  
}

//ColorType.kt  
import java.util.\*  
  
enum class ColorType(val rgb: String, val number:Int) {  
  
 *RED*("255, 0, 0", 1),  
 *ORANGE*("255, 127, 0", 2),  
 *YELLOW*("255, 255, 0", 3),  
 *GREEN*("0, 255, 0", 4),  
 *BLUE*("0, 0, 255", 5),  
 *INDOGO*("75, 0, 130", 6),  
 *VIOLET*("143, 0, 255", 7);  
  
 fun colot\_list\_info() : List<ColorType>{  
 *println*("Вот все возможные цвета, которые добавлены в программу")  
 return values().*toList*()  
 }  
  
 fun rgb\_color(){  
 *println*("Введите номер цвета и узнайте как он защифрован в ргб")  
 *println*("1 - красный\n2 - оранжевый\n3 - желтый\n4- зеленый\n5- синий\n6-голубой\n7-фиолетовый")  
  
 try {  
 val number :Int = *readln*()!!.*toInt*()  
 when(number){  
 1 -> *println*(*RED*.rgb)  
 2 -> *println*(*ORANGE*.rgb)  
 3 -> *println*(*YELLOW*.rgb)  
 4 -> *println*(*GREEN*.rgb)  
 5 -> *println*(*BLUE*.rgb)  
 6 -> *println*(*INDOGO*.rgb)  
 7 -> *println*(*VIOLET*.rgb)  
 else -> *println*("Такого цвета нет")  
 }  
 }  
 catch (e: NumberFormatException){  
 *println*("Необходимо ввести число")  
 }  
  
 }  
  
 fun game (){  
 val color : ColorType = values().*random*()  
 val anwers = color.number  
 *println*("Каким по счету в радуге является этот $color цвет?")  
 try {  
 val player :Int = *readln*()!!.*toInt*()  
 if(player == anwers) *println*("Правильно!")  
 else *println*("Вы ответили неверно. Этот цвет $anwers по счете в радуге")  
 }  
 catch (e: NumberFormatException){  
 *println*("Необходимо ввести число")  
 }  
 }  
  
 fun random\_color(){  
 *println*("Рандомный цвет: ${values().*random*()}")  
 }  
  
}

//Days.kt  
import java.util.\*  
  
enum class Days {  
 *Monday*, *Tuesday*, *Wednesday*, *Thursday*, *Friday*, *Saturday*, *Sunday*;  
  
 fun WorkAndWeekendDays() {  
 *println*("Введите номер дня недели")  
 try {  
 val number = *readLine*()!!.*toInt*()  
 when (number) {  
 1 -> *println*("${Days.*Monday*} - Рабочий день")  
 2 -> *println*("${Days.*Tuesday*} - Рабочий день")  
 3 -> *println*("${Days.*Wednesday*} - Рабочий день")  
 4 -> *println*("${Days.*Thursday*} - Рабочий день")  
 5 -> *println*("${Days.*Friday*} - Рабочий день")  
 6 -> *println*("${Days.*Saturday*} - Выходной день")  
 7 -> *println*("${Days.*Sunday*} - Выходной день")  
 else -> *println*("Такого дня недели нет")  
 }  
  
 } catch (e: NumberFormatException) {  
 *println*("Необходимо ввести число")  
 }  
 }  
  
 fun LastDay(day: Days){  
 val index = (day.ordinal - 1) % values().size  
 if(index > 0){  
 *println*("Предыдущий день: ${values()[index]}")  
 }  
 else {  
 *println*("Предыдущий день: ${values()[6]}")  
 }  
  
  
 }  
  
 fun RandomDay(): Days {  
 *println*("Случайный день недели:")  
 return values().*random*()  
 }  
  
 fun NumberDays() {  
 *println*("Введите номер дня недели")  
 try {  
 val number = *readLine*()!!.*toInt*()  
 when (number) {  
 1 -> {  
 *println*(Days.*Monday*)  
 LastDay(*Monday*)  
 }  
  
 2 -> {  
 *println*(Days.*Tuesday*)  
 LastDay(*Thursday*)  
 }  
  
 3 -> {  
 *println*(Days.*Wednesday*)  
 LastDay(*Wednesday*)  
 }  
  
 4 -> {  
 *println*(Days.*Thursday*)  
 LastDay(*Thursday*)  
 }  
  
 5 -> {  
 *println*(Days.*Friday*)  
 LastDay(*Friday*)  
 }  
  
 6 -> {  
 *println*(Days.*Saturday*)  
 LastDay(*Saturday*)  
 }  
  
 7 -> {  
 *println*(Days.*Sunday*)  
 LastDay(*Sunday*)  
 }  
 else -> *println*("Такого дня недели нет")  
 }  
  
 } catch (e: NumberFormatException) {  
 *println*("Необходимо ввести число")  
 }  
 }  
  
}

**Тестовые ситуации**

Если пользователь введет несуществующий день недели или цвета – программа сообщит об ошибке. Если программа требует ввести число, а пользователь введет буквы – программа тоже сообщит об ошибке.

**Вывод**

Я научилась работать с классами enum.