Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практика 11 (enum)**»

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**

1. **С помощью класса enum задайте дни недели и цвета и используя вспомогательные функции, выдайте значения на экран в виде строки и числа.**

**Входные данные: DayName – String, numberdays – Int,**

**DaysOfJob() – Bolean, ChenNeChet() – Bolean**

**Выходные данные: DayName – String, numberdays – Int,**

**DaysOfJob() – Bolean, ChenNeChet() – Bolean**

enum class DaysWeek {  
 *Monday*{  
 override fun DayName() = "Понедельник"  
 override val numberdays = 1  
 override fun DaysOfJob() = true  
 override fun ChenNeChet() = false  
 },  
 *Tuesday*{  
 override fun DayName() = "Вторник"  
 override val numberdays = 2  
 override fun DaysOfJob() = true  
 override fun ChenNeChet() = true  
 },  
 *Wednesday*{  
 override fun DayName() = "Среда"  
 override val numberdays = 3  
 override fun DaysOfJob() = true  
 override fun ChenNeChet() = false  
 },  
 *Thursday*{  
 override fun DayName() = "Четверг"  
 override val numberdays = 4  
 override fun DaysOfJob() = true  
 override fun ChenNeChet() = true  
 },  
 *Friday*{  
 override fun DayName() = "Пятница"  
 override val numberdays = 5  
 override fun DaysOfJob() = true  
 override fun ChenNeChet() = false  
 },  
 *Saturday*{  
 override fun DayName() = "Суббота"  
 override val numberdays = 6  
 override fun DaysOfJob() = false  
 override fun ChenNeChet() = true  
 },  
 *Sunday*{  
 override fun DayName() = "Воскресенье"  
 override val numberdays = 7  
 override fun DaysOfJob() = false  
 override fun ChenNeChet() = false  
 };  
  
 open fun DayName() = "Понедельник, Вторник, Среда, Четверг, Пятница, Суббота, Воскресенье"  
 open val numberdays = 0  
 abstract fun DaysOfJob(): Boolean  
 abstract fun ChenNeChet(): Boolean  
  
}

**Входные данные: colorName – String, rgb– Int,**

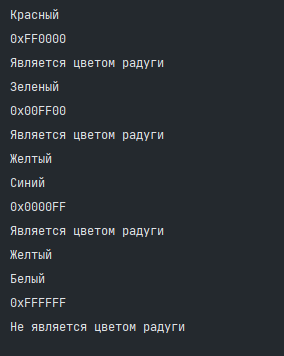
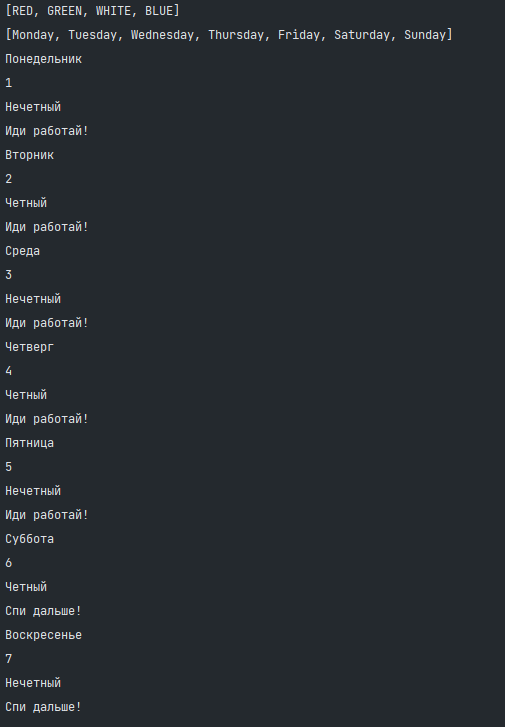
**Raduga() – Bolean, NumberColor() – Int**

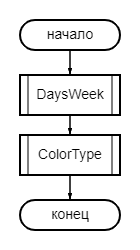
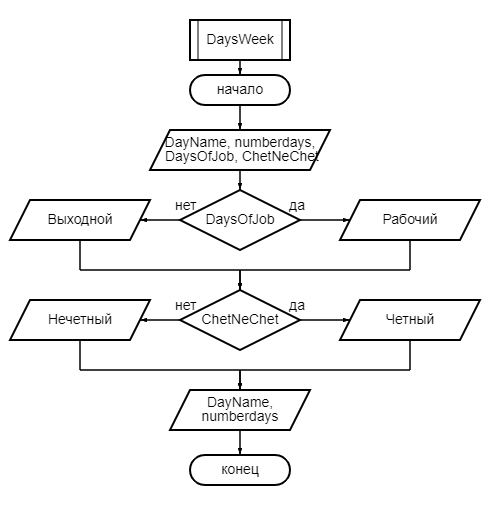
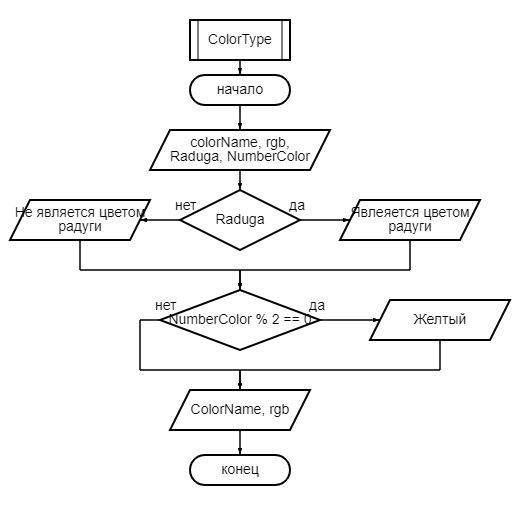
**Выходные данные: colorName – String, rgb– Int,**

**Raduga() – Bolean, NumberColor() - Int**

enum class ColorType {  
 *RED* {  
 override fun colorName() = "Красный"  
 override val rgb = "0xFF0000"  
 override fun Raduga() = true  
 override fun NumberColor() = 1  
 },  
 *GREEN*{  
 override fun colorName() = "Зеленый"  
 override val rgb = "0x00FF00"  
 override fun Raduga() = true  
 override fun NumberColor() = 2  
 },  
 *WHITE*{  
 override fun colorName() = "Белый"  
 override val rgb = "0xFFFFFF"  
 override fun Raduga() = false  
 override fun NumberColor() = 3  
 },  
 *BLUE* {  
 override fun colorName() = "Синий"  
 override val rgb = "0x0000FF"  
 override fun Raduga() = true  
 override fun NumberColor() = 4  
 };  
  
 abstract fun colorName(): String  
 abstract val rgb: String  
 abstract fun Raduga(): Boolean  
 abstract fun NumberColor(): Int  
}

fun main(){  
 *println*(ColorType.values().*asList*())  
 *println*(DaysWeek.values().*asList*())  
  
 val Monday = DaysWeek.*Monday  
 println*(Monday.DayName())  
 *println*(Monday.numberdays)  
 if(Monday.ChenNeChet()) *println*("Четный")  
 else *println*("Нечетный")  
  
 if(Monday.DaysOfJob()) *println*("Иди работай!")  
 else *println*("Спи дальше!")  
  
 val Tuesday = DaysWeek.*Tuesday  
 println*(Tuesday.DayName())  
 *println*(Tuesday.numberdays)  
 if(Tuesday.ChenNeChet()) *println*("Четный")  
 else *println*("Нечетный")  
  
 if(Tuesday.DaysOfJob()) *println*("Иди работай!")  
 else *println*("Спи дальше!")  
  
 val Wednesday = DaysWeek.*Wednesday  
 println*(Wednesday.DayName())  
 *println*(Wednesday.numberdays)  
 if(Wednesday.ChenNeChet()) *println*("Четный")  
 else *println*("Нечетный")  
  
 if(Wednesday.DaysOfJob()) *println*("Иди работай!")  
 else *println*("Спи дальше!")  
  
 val Thursday = DaysWeek.*Thursday  
 println*(Thursday.DayName())  
 *println*(Thursday.numberdays)  
 if(Thursday.ChenNeChet()) *println*("Четный")  
 else *println*("Нечетный")  
  
 if(Thursday.DaysOfJob()) *println*("Иди работай!")  
 else *println*("Спи дальше!")  
  
 val Friday = DaysWeek.*Friday  
 println*(Friday.DayName())  
 *println*(Friday.numberdays)  
 if(Friday.ChenNeChet()) *println*("Четный")  
 else *println*("Нечетный")  
  
 if(Friday.DaysOfJob()) *println*("Иди работай!")  
 else *println*("Спи дальше!")  
  
 val Saturday = DaysWeek.*Saturday  
 println*(Saturday.DayName())  
 *println*(Saturday.numberdays)  
 if(Saturday.ChenNeChet()) *println*("Четный")  
 else *println*("Нечетный")  
  
 if(Saturday.DaysOfJob()) *println*("Иди работай!")  
 else *println*("Спи дальше!")  
  
 val Sunday = DaysWeek.*Sunday  
 println*(Sunday.DayName())  
 *println*(Sunday.numberdays)  
 if(Sunday.ChenNeChet()) *println*("Четный")  
 else *println*("Нечетный")  
  
 if(Sunday.DaysOfJob()) *println*("Иди работай!")  
 else *println*("Спи дальше!")  
  
 val Red = ColorType.*RED  
 println*("\n${Red.colorName()}")  
 *println*(Red.rgb)  
 if(Red.Raduga()) *println*("Является цветом радуги")  
 else *println*("Не является цветом радуги")  
  
 if(Red.NumberColor() % 2 == 0) *println*("Желтый")  
  
  
 val Green = ColorType.*GREEN  
 println*(Green.colorName())  
 *println*(Green.rgb)  
 if(Green.Raduga()) *println*("Является цветом радуги")  
 else *println*("Не является цветом радуги")  
 if(Green.NumberColor() % 2 == 0) *println*("Желтый")  
  
 val Blue = ColorType.*BLUE  
 println*(Blue.colorName())  
 *println*(Blue.rgb)  
 if(Blue.Raduga()) *println*("Является цветом радуги")  
 else *println*("Не является цветом радуги")  
 if(Blue.NumberColor() % 2 == 0) *println*("Желтый")  
  
  
 val White = ColorType.*WHITE  
 println*(White.colorName())  
 *println*(White.rgb)  
 if(White.Raduga()) *println*("Является цветом радуги")  
 else *println*("Не является цветом радуги")  
 if(White.NumberColor() % 2 == 0) *println*("Желтый")  
}

****

****

**Вывод: Научился пользоваться enum class в Kotlin**