Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

**Отчёт по программе «Практическое задание №11»**

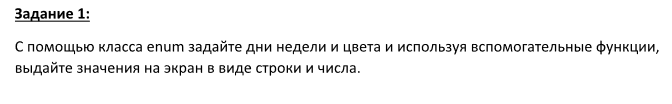
Выполнил: Зиятдинова Алина Ленаровна

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Задание:**

**Листинг программы**

enum class ColorType {  
 *YELLOW* {  
 override fun ColorName() = "Желтый";  
 override val rgb = "0xFFFF00";  
 override fun ranbow() = "Да"  
 override fun NearColor():String  
 {  
 var bl\_color = "Оранжевый"  
 return bl\_color  
 }  
 override fun CompareColor():String  
 {  
 var color = "Зеленый + красный"  
 return color  
 }  
 },  
 *GREEN* {  
 override fun ColorName() = "Зеленый";  
 override val rgb = "0x00FF00"  
 override fun ranbow() = "Да"  
 override fun NearColor():String  
 {  
 var bl\_color = "Изумрудный"  
 return bl\_color  
 }  
 override fun CompareColor():String  
 {  
 var color = "Синий + желтый"  
 return color  
 }  
 },  
 *RED* {  
 override fun ColorName() = "Красный";  
 override val rgb = "0x00FF00"  
 override fun ranbow() = "Да"  
 override fun NearColor():String  
 {  
 var bl\_color = "Алый"  
 return bl\_color  
 }  
 override fun CompareColor():String  
 {  
 var color = "Розовый + желтый"  
 return color  
 }  
 },  
 *BLUE* {  
 override fun ColorName() = "Синий";  
 override val rgb = "0x0000FF"  
 override fun ranbow() = "Да"  
 override fun NearColor():String  
 {  
 var bl\_color = "Голубой"  
 return bl\_color  
 }  
 override fun CompareColor():String  
 {  
 var color = "Желтый + зеленый"  
 return color  
 }  
 },  
 *PINK* {  
 override fun ColorName() = "Розовый";  
 override val rgb = "0xFF00A2"  
 override fun ranbow() = "Нет"  
 override fun NearColor():String  
 {  
 var bl\_color = "Малиновый"  
 return bl\_color  
 }  
 override fun CompareColor():String  
 {  
 var color = "Красный + белый"  
 return color  
 }  
 };  
 abstract val rgb: String  
 abstract fun ColorName():String  
 abstract fun ranbow(): String  
 abstract fun NearColor():String  
 abstract fun CompareColor():String  
}

enum class Day {  
 *Monday* {  
 override fun DayName() = "Понедельник";  
 override fun Budni() = "Да";  
 override fun Planeta() = "Луна"  
 override fun Chet() = "Нет"  
  
 },  
 *Tuesday* {  
 override fun DayName() = "Вторник";  
 override fun Budni() = "Да";  
 override fun Planeta() = "Марс"  
 override fun Chet() = "Да"  
 },  
 *Wednesday* {  
 override fun DayName() = "Среда";  
 override fun Budni() = "Да";  
 override fun Planeta() = "Меркурий"  
 override fun Chet() = "Нет"  
  
 },  
 *Thursday* {  
 override fun DayName() = "Четверг";  
 override fun Budni() = "Да";  
 override fun Planeta() = "Юпитер"  
 override fun Chet() = "Да"  
  
 },  
 *Friday* {  
 override fun DayName() = "Пятница";  
 override fun Budni() = "Да";  
 override fun Planeta() = "Венера"  
 override fun Chet() = "Нет"  
  
 },  
 *Saturday* {  
 override fun DayName() = "Суббота";  
 override fun Budni() = "Нет";  
 override fun Planeta() = "Сатурн"  
 override fun Chet() = "Да"  
  
 },  
 *Sunday* {  
 override fun DayName() = "Воскресенье";  
 override fun Budni() = "Нет";  
 override fun Planeta() = "Солнце"  
 override fun Chet() = "Нет"  
 };  
 abstract fun DayName():String  
 abstract fun Budni():String  
 abstract fun Planeta():String  
 abstract fun Chet():String  
}

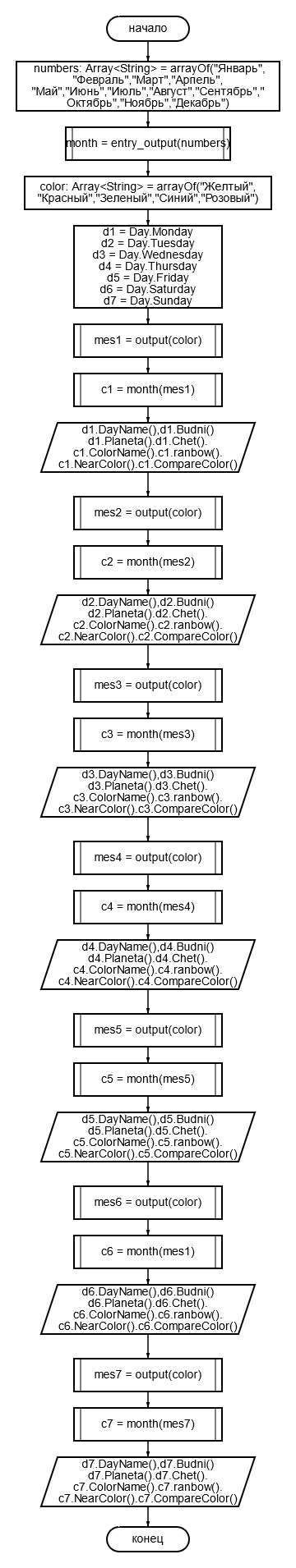
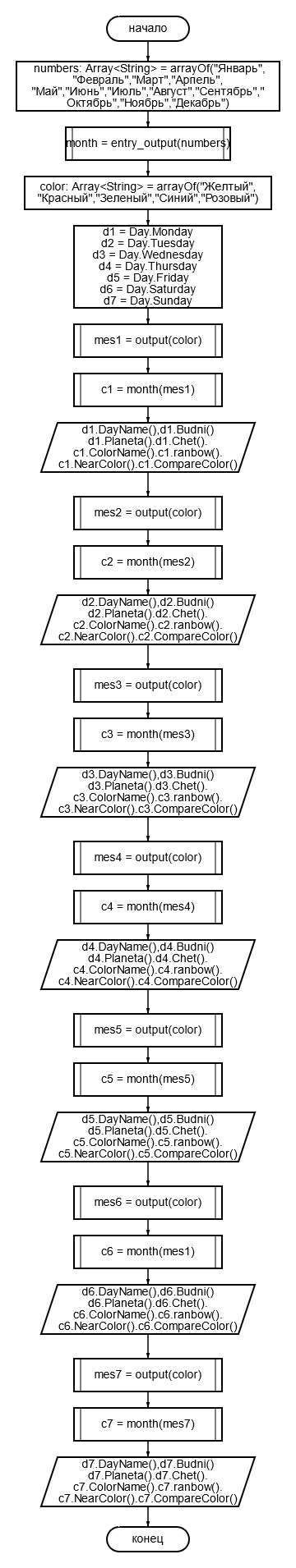
import ColorType  
import kotlin.random.Random  
import Day  
import java.lang.Exception  
import java.security.spec.ECField  
fun main() {  
 fun entry\_output(numbers:Array<String>):String  
 {  
 var mes: String = ""  
 var num:Int = 0  
 while (true) {  
 *println*("Выберите месяц. Введите число от 1 до 12")  
 for (i in numbers) {  
 *print*(i + " ")  
 }  
 *println*()  
 try  
 {  
 num = *readLine*()!!.*toInt*()  
 while (num<0 && num >12)  
 {  
 *println*("Число не может быть меньше нуля и больше 12. Повторите ввод.")  
 num = *readLine*()!!.*toInt*()  
 }  
 mes = numbers[num-1]  
 break  
 }  
 catch (e: Exception)  
 {  
 *println*("Возникло исключение! Повторите ввод.")  
 }  
 }  
 return mes  
 }  
 fun output(color:Array<String>):String  
 {  
 var num: Int = 0  
 while (true) {  
 for (i in color) {  
 *print*(i + " ")  
 }  
 *println*()  
 *println*("Введите число от 1 до 5.")  
 try {  
 num = *readLine*()!!.*toInt*()  
 while (num>5)  
 {  
 *println*("Вы вели число больше 5. Повторите ввод.")  
 num = *readLine*()!!.*toInt*()  
 }

var mes = ""  
 mes = color[num-1]  
 break  
 } catch (e: Exception) {  
 *println*("Возникло исключение! Повотрите ввод.")  
 }  
 }  
  
 return mes  
 }  
 fun month(color:String):ColorType  
 {  
 var c = ColorType.*YELLOW* if (color == "Желтый")  
 c = ColorType.*YELLOW* else if (color == "Красный")  
 c = ColorType.*RED* else if (color == "Зеленый")  
 c = ColorType.*GREEN* else if (color == "Синий")  
 c = ColorType.*BLUE* else if (color == "Розовый")  
 c = ColorType.*PINK* return c  
 }  
 val numbers: Array<String> = *arrayOf*("Январь","Февраль","Март","Арпель","Май","Июнь","Июль","Август","Сентябрь","Октябрь","Ноябрь","Декабрь")  
 var month = entry\_output(numbers)  
 val color: Array<String> = *arrayOf*("Желтый","Красный","Зеленый","Синий","Розовый")  
 *println*("Месяц: "+ month)  
 *println*()  
 val d1 = Day.*Monday* val d2 = Day.*Tuesday* val d3 = Day.*Wednesday* val d4 = Day.*Thursday* val d5 = Day.*Friday* val d6 = Day.*Saturday* val d7 = Day.*Sunday  
 println*("День недели: ${d1.DayName()}. Как отметить этот день?")  
 var mes1 = output(color)  
 var c1 = month(mes1)  
 *println*()  
 *println*("День недели: ${d1.DayName()}\nБудни: ${d1.Budni()}\nВедущая ланета: ${d1.Planeta()}\nЧетный день: ${d1.Chet()}\nЦвет: ${c1.ColorName()}\n" +  
 "Цвет есть в радуге?: ${c1.ranbow()}\nБлизкий цвет: ${c1.NearColor()}\nКак получить цвет: ${c1.CompareColor()}\n")  
 *println*("День недели: ${d2.DayName()}. Как отметить этот день?")  
 var mes2 = output(color)  
 var c2 = month(mes2)  
 *println*()  
 *println*("День недели: ${d2.DayName()}\nБудни: ${d2.Budni()}\nВедущая ланета: ${d2.Planeta()}\nЧетный день: ${d2.Chet()}\n" +  
 "Цвет есть в радуге?: ${c2.ranbow()}\nБлизкий цвет: ${c2.NearColor()}\nКак получить цвет: ${c2.CompareColor()}\n")  
 var mes3 = output(color)  
 var c3 = month(mes3)  
 *println*()  
 *println*("День недели: ${d3.DayName()}\nБудни: ${d3.Budni()}\nВедущая ланета: ${d3.Planeta()}\nЧетный день: ${d3.Chet()}\n" +  
 "Цвет есть в радуге?: ${c3.ranbow()}\n" +  
 "Близкий цвет: ${c3.NearColor()}\n" +  
 "Как получить цвет: ${c3.CompareColor()}\n")  
  
 var mes4 = output(color)  
 var c4 = month(mes4)  
 *println*()  
 *println*("День недели: ${d4.DayName()}\nБудни: ${d4.Budni()}\nВедущая ланета: ${d4.Planeta()}\nЧетный день: ${d4.Chet()}\n" +  
 "Цвет есть в радуге?: ${c4.ranbow()}\n" +  
 "Близкий цвет: ${c4.NearColor()}\n" +  
 "Как получить цвет: ${c4.CompareColor()}\n")  
  
 var mes5 = output(color)  
 var c5 = month(mes5)  
 *println*()  
 *println*("День недели: ${d5.DayName()}\nБудни: ${d5.Budni()}\nВедущая ланета: ${d5.Planeta()}\nЧетный день: ${d5.Chet()}\n" +  
 "Цвет есть в радуге?: ${c5.ranbow()}\n" +  
 "Близкий цвет: ${c5.NearColor()}\n" +  
 "Как получить цвет: ${c5.CompareColor()}\n")  
  
  
 var mes6 = output(color)  
 var c6 = month(mes6)  
 *println*()  
 *println*("День недели: ${d6.DayName()}\nБудни: ${d6.Budni()}\nВедущая ланета: ${d6.Planeta()}\nЧетный день: ${d6.Chet()}\n" +  
 "Цвет есть в радуге?: ${c6.ranbow()}\n" +  
 "Близкий цвет: ${c6.NearColor()}\n" +  
 "Как получить цвет: ${c6.CompareColor()}\n")  
  
 var mes7 = output(color)  
 var c7 = month(mes7)  
 *println*()  
 *println*("День недели: ${d7.DayName()}\nБудни: ${d7.Budni()}\nВедущая ланета: ${d7.Planeta()}\nЧетный день: ${d7.Chet()}\n" +  
 "Цвет есть в радуге?: ${c7.ranbow()}\n" +  
 "Близкий цвет: ${c7.NearColor()}\n" +  
 "Как получить цвет: ${c7.CompareColor()}\n")  
}

**Блок схема**

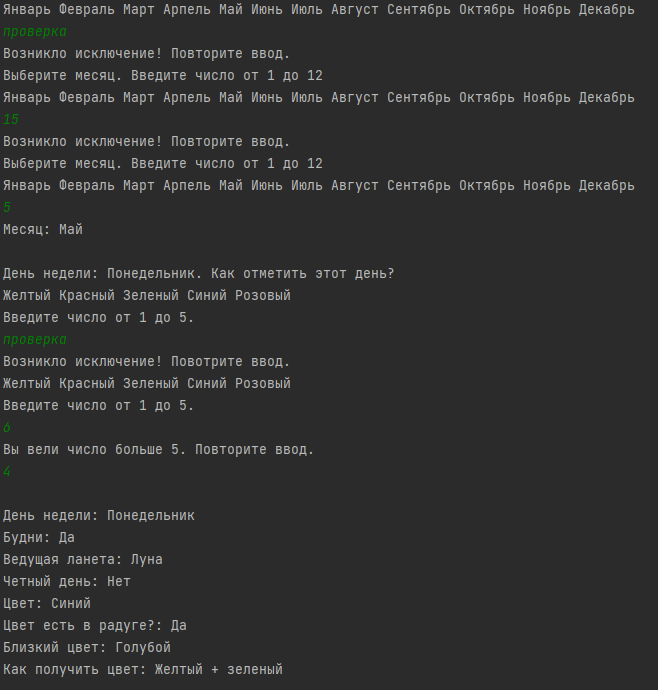
|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | | |

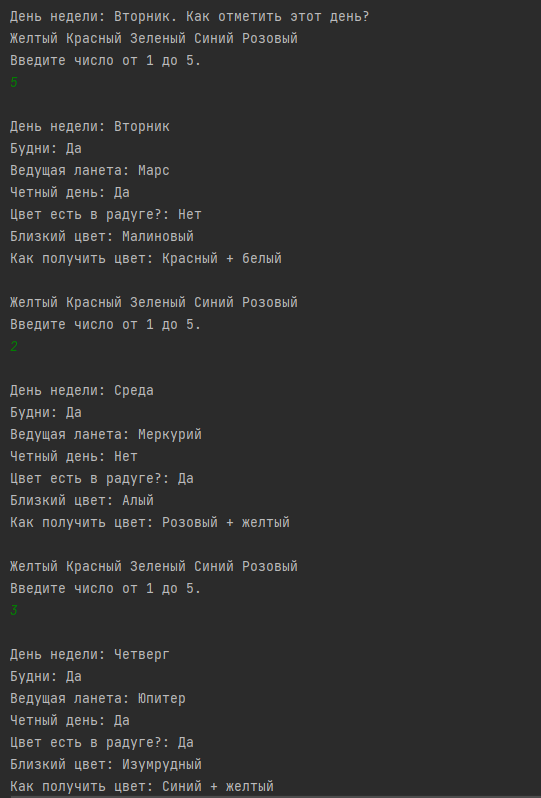
|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

****

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Тестовые ситуации**

****

****

