Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практика 10**»

Выполнил: Деспижек Булат

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Задание 1:**

**Входные данные:**

**Число – Целого типа**

**Выходные данные:**

**Информация о днях -Строкового типа**

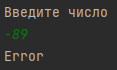
**Листинг-Кода:**

import WeekDays.\*  
fun main(){  
 try {  
 *println*("Введите число")  
 var num: Int = *readLine*()!!.*toInt*()  
 when {  
 (num<=0 || num >31)->throw Exception("Error")  
 }  
 //if(num==0)throw Exception("Error")  
 num = num % 7  
 val day:Day  
 when (num) {  
 1 -> {  
 day = Day.*Monday* }  
 2 -> {  
 day = Day.*Tuesday* }  
 3 -> {  
 day = Day.*Wednesday* }  
 4 -> {  
 day = Day.*Thursday* }  
 5 -> {  
 day = Day.*Friday* }  
 6 -> {  
 day = Day.*Saturday* }  
 0 -> {  
 day = Day.*Sunday* }  
 else ->throw Exception("Error")  
 }  
 *println*(day.Getday(day))  
 day.Weekend(day)  
 *println*("${day.Yesterday(day)}")  
 *println*("${day.Tomorrow(day)}")  
  
 val colorRed = ColorType.*RED  
 println*(colorRed.name)  
 *println*(colorRed.ordinal)  
  
 }catch (ex:Exception){  
 *println*(ex.message)  
 }  
}

package WeekDays  
enum class Day {  
 *Monday*,  
 *Tuesday*,  
 *Wednesday*,  
 *Thursday*,  
 *Friday*,  
 *Saturday*,  
 *Sunday*;  
  
 public fun Getday(day: Day): String {  
 return when (day) {  
 Day.*Monday* -> "Понедельник"  
 Day.*Tuesday* -> "Вторник"  
 Day.*Wednesday* -> "Среда"  
 Day.*Thursday* -> "Четверг"  
 Day.*Friday* -> "Пятница"  
 Day.*Saturday* -> "Суббота"  
 Day.*Sunday* -> "Воскресенье"  
 }  
 }  
  
 public fun Weekend(day: Day) {  
 when (day.ordinal) {  
 in 0..4 -> *println*("Рабочий день")  
 else -> *println*("Выходной день")  
 }  
 }  
  
 public fun Yesterday(day:Day):String{  
 val yesterday:Day  
 var vyvod:String  
 when(day){  
 Day.*Monday* -> {  
 yesterday=Day.*Sunday* }  
 Day.*Tuesday* -> {  
 yesterday=Day.*Monday* }  
 Day.*Wednesday* -> {  
 yesterday=Day.*Tuesday* }  
 Day.*Thursday* -> {  
 yesterday=Day.*Wednesday* }  
 Day.*Friday* -> {  
 yesterday=Day.*Thursday* }  
 Day.*Saturday* -> {  
 yesterday=Day.*Friday* }  
 Day.*Sunday* -> {  
 yesterday=Day.*Saturday* }  
 }  
  
 when(yesterday.ordinal){  
 in 0..1->vyvod="Вчера был ${yesterday.Getday(yesterday)}"  
 2->vyvod="Вчера былa ${yesterday.Getday(yesterday)}"  
 3->vyvod="Вчера был ${yesterday.Getday(yesterday)}"  
 4->vyvod="Вчера была ${yesterday.Getday(yesterday)}"  
 5->vyvod="Вчера была ${yesterday.Getday(yesterday)}"  
 6->vyvod="Вчера было ${yesterday.Getday(yesterday)}"  
 else->throw Exception("Error")  
 }  
 return vyvod  
 }  
  
 public fun Tomorrow(day:Day):String{  
 val tomorrow:Day  
 var vyvod:String  
 when(day){  
 Day.*Monday* -> {  
 tomorrow=Day.*Tuesday* }  
 Day.*Tuesday* -> {  
 tomorrow=Day.*Wednesday* }  
 Day.*Wednesday* -> {  
 tomorrow=Day.*Thursday* }  
 Day.*Thursday* -> {  
 tomorrow=Day.*Friday* }  
 Day.*Friday* -> {  
 tomorrow=Day.*Saturday* }  
 Day.*Saturday* -> {  
 tomorrow=Day.*Sunday* }  
 Day.*Sunday* -> {  
 tomorrow=Day.*Monday* }  
 }  
  
 when(tomorrow.ordinal){  
 in 0..6->vyvod="Завтра будет ${tomorrow.Getday(tomorrow)}"  
 else->throw Exception("Error")  
 }  
 return vyvod  
 }  
}

package WeekDays  
enum class ColorType/\*(val r:Int, val g:Int, val b:Int)\*/ {  
 /\*Red(255,0,0),  
 Blue(0,0,255),  
 Yellow( 255,255,0),  
 Purple( 128,0,128),  
 Green(0,255,0),  
 Pink(255,192,203),  
 Orange(255,165,0);\*/  
 *RED*{  
 override fun colorName() = "Красный"  
 override val rgb = "0xFF0000"  
 },  
 *BLUE* {  
 override fun colorName() = "Голубой"  
 override val rgb = "0x0000FF"  
 },  
 *GREEN* {  
 override fun colorName() = "Зелёный"  
 override val rgb = "0x00FF00"  
 };  
 abstract fun colorName(): String  
 abstract val rgb: String  
}

**Тестовые ситуации:**

****

**Блок-Схема:**

