**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра САПР**

**ОТЧЕТ**

**по заданию на оценку «удовлетворительно»**

**по дисциплине «Программирование на языке Kotlin»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 9309  Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Корягин Е.А  Красильников А.В. |

Санкт-Петербург

2021

Содержание

[Задание на оценку «удовлетворительно» 3](#_Toc91203043)

[Задание 3](#_Toc91203044)

[Решение 3](#_Toc91203045)

[Выводы 3](#_Toc91203046)

# Задание на оценку «удовлетворительно»

# Задание

Реализовать иерархию классов приложения “заметки”. У каждой заметки должно быть названия (title), цвет(color) и содержимое (content). Контент должен быть реализован в виде sealed класса с следующими наследниками:

Text - контект, который хранит в себе список строк

Image - хранит одну строку, которая представляет собой ссылку или путь в системе к файлу с картинкой

ToDoList - список элементов из названия задачи и отметки о том, выполнена она или нет

В функции main создать список из нескольких заметок с разным типом контента и вывести его в консоль.

Опционально добавить валидацию данных при создании объектов.

# Решение

Color.kt:

|  |
| --- |
| enum class Color { |
|  | RED, |
|  | GREEN, |
|  | BLUE, |
|  | GRAY, |
|  | BLACK, |
|  | BROWN, |
|  | PURPLE |
|  | } |

NoteContent.kt:

|  |
| --- |
| import java.nio.file.Path |
|  |  |
|  | sealed class NoteContent { |
|  | data class Text(val strings: List<String>) : NoteContent() |
|  | data class Image(val imagePath: Path) : NoteContent() |
|  | data class ToDoList(val toDoItemList : List<ToDoListItem>) : NoteContent() |
|  | } |

ToDoListItem.kt

data class ToDoListItem(val checked: Boolean, val taskName: String)

Note.kt:

|  |
| --- |
| import java.nio.file.Path |
|  |  |
|  | data class Note (val title: String, val color: Color, val content: NoteContent) |
|  |  |
|  | fun main() { |
|  | val firstNote = Note("first", Color.RED, NoteContent.Text(listOf("firstString", "secondString"))) |
|  | val secondNote = Note("second", Color.BLUE, NoteContent.Image(Path.of("var2", "second"))) |
|  | val thirdNote = Note("third", |
|  | Color.GRAY, NoteContent.ToDoList(listOf(ToDoListItem(true, "firstTask"), ToDoListItem(false, "secondTask")))) |
|  | val fourthNote = Note("fourth", Color.GREEN, NoteContent.ToDoList(listOf(ToDoListItem(true,"firstTask"), ToDoListItem(false, "secondTask")))) |
|  |  |
|  | val notes: List<Note> = listOf(firstNote, secondNote, thirdNote, fourthNote) |
|  |  |
|  | println(notes) |
|  | } |

# Выводы

В процессе освоения курса и выполнения работ было проведено ознакомление с основными инструментами ЯП Kotlin, их сравнение с аналогами из ранее изученных ЯП. Среди особенностей инструментария Kotlin можно отметить удобство для парадигмы ООП, а также немногословность, обеспечиваемую сокращениями стандартных языковых конструкций.