Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Кафедра инженерной психологии и эргономики

Современные языки программирования

Отчет по лабораторной работе №1

«Использование языка программирования Kotlin»

Выполнил: Фомина А.С.

Студент группы 310901

Преподаватель: Усенко Ф. В.

Минск 2024

Цель: Выполнить разработку приложения с использованием языка программирования Kotlin

Задание к лабораторной работе:

* согласно варианта задания написать программу (Вариант = номер в списке группы % 30 + 1);
* объекты и их взаимоотношения, имеющиеся в варианте задания, должны быть реализованы;
* функциональная часть приложения, представленная диаграммой последовательности, должна быть реализована;
* разбить функционал приложения на несколько пакетов придерживаясь логики.

Вариант 30

Добавить класс Плейлист, хранящий список видео. Класс должен выдавать следующее видео из списка при вызове метода Воспроизвести плейлист(). Добавить класс Музыкальное видео.

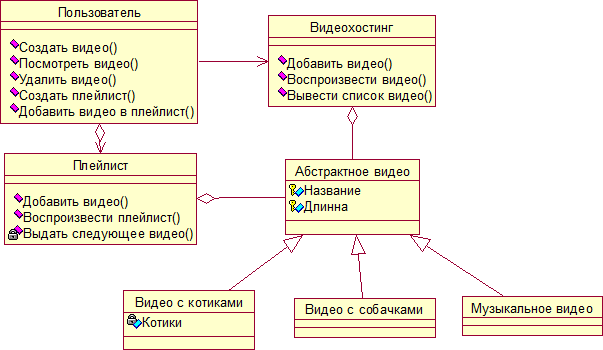


Рисунок 60 – Диаграмма классов

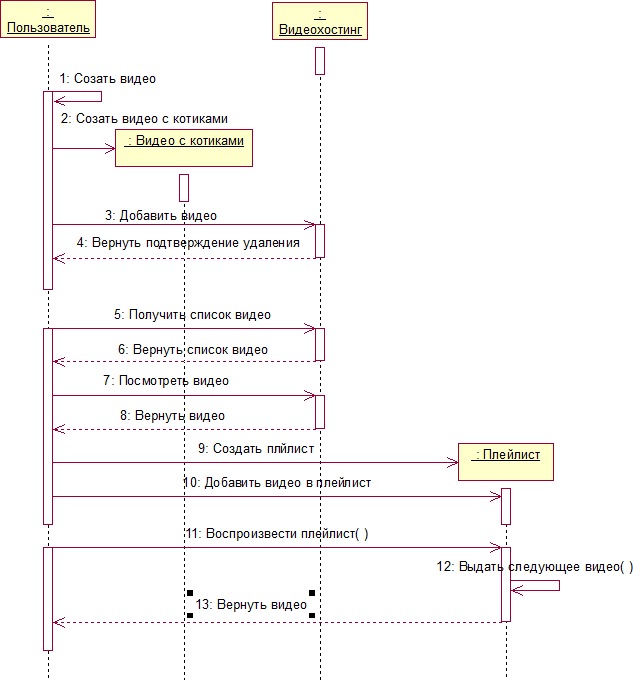


Рисунок 61 – Диаграмма последовательности

fun main() {

val user = User()

val videoHosting = VideoHosting()

val playlist = user.createPlaylist()

while (true) {

println("1. Создать видео")

println("2. Посмотреть видео")

println("3. Создать плейлист")

println("4. Добавить видео в плейлист")

println("5. Воспроизвести плейлист")

println("6. Вывести список видео")

println("7. Удалить видео")

println("8. Выйти")

print("Выберите опцию: ")

when (readLine()?.toIntOrNull()) {

1 -> {

println("1. Видео с котиками")

println("2. Видео с собаками")

println("3. Музыкальное видео")

print("Выберите тип видео: ")

when (readLine()?.toIntOrNull()) {

1 -> {

println("Введите название видео: ")

val name = readLine() ?: ""

println("Введите длину видео: ")

val length = readLine()?.toIntOrNull() ?: 0

println("Введите описание котиков: ")

val cats = readLine() ?: ""

val catVideo = CatVideo(name, length, cats)

user.createVideo(videoHosting, catVideo)

}

2 -> {

println("Введите название видео: ")

val name = readLine() ?: ""

println("Введите длину видео: ")

val length = readLine()?.toIntOrNull() ?: 0

val dogVideo = DogVideo(name, length)

user.createVideo(videoHosting, dogVideo)

}

3 -> {

println("Введите название видео: ")

val name = readLine() ?: ""

println("Введите длину видео: ")

val length = readLine()?.toIntOrNull() ?: 0

val musicVideo = MusicVideo(name, length)

user.createVideo(videoHosting, musicVideo)

}

}

}

2 -> {

println("Введите название видео для просмотра: ")

val name = readLine() ?: ""

user.watchVideo(videoHosting, name)

}

3 -> {

println("Плейлист создан.")

}

4 -> {

println("Введите название видео для добавления в плейлист: ")

val name = readLine() ?: ""

val video = videoHosting.videos.find { it.videoName == name }

if (video != null) {

user.addToPlaylist(playlist, video)

} else {

println("Видео не найдено!")

}

}

5 -> {

println("Воспроизведение плейлиста...")

playlist.playPlaylist()

}

6 -> {

videoHosting.showAllVideo()

}

7 -> {

println("Введите название видео для удаления: ")

val name = readLine() ?: ""

user.deleteVideo(videoHosting, name)

playlist.deleteFromPlaylist(name)

}

8 -> {

break

}

else -> println("Неверный выбор. Пожалуйста, выберите снова.")

}

}

} class User() {

fun createVideo(videoHosting: VideoHosting, video: AbstractVideo) {

videoHosting.addVideo(video)

println("Создал видео ${video.videoName}")

}

fun watchVideo(videoHosting: VideoHosting, videoName: String) {

videoHosting.playVideo(videoName)

}

fun deleteVideo(videoHosting: VideoHosting, videoName: String) {

videoHosting.deleteVideo(videoName)

}

fun createPlaylist(): Playlist {

return Playlist()

}

fun addToPlaylist(playlist: Playlist, video: AbstractVideo) {

playlist.addVideo(video)

println("Добавлено видео ${video.videoName} в плейлист")

}

} class VideoHosting {

val videos = mutableListOf<AbstractVideo>()

private val playlists = mutableListOf<Playlist>()

fun addVideo(video: AbstractVideo) {

videos.add(video)

}

fun playVideo(name: String) {

val video = videos.find { it.videoName == name }

if (video != null) {

println("Воспроизведение видео: ${video.videoName}, длина: ${video.videoLength}")

} else {

println("Видео не найдено!")

}

}

fun showAllVideo() {

println("Список видео:")

for (video in videos) {

println("- ${video.videoName}, длина: ${video.videoLength}")

}

}

fun deleteVideo(name: String) {

val video = videos.find { it.videoName == name }

if (video != null) {

videos.remove(video)

println("Видео '${name}' удалено.")

playlists.forEach { it.deleteFromPlaylist(name) }

} else {

println("Видео не найдено!")

}

}

} open class AbstractVideo(

protected val name: String,

protected val length: Int

) {

val videoName: String

get() = name

val videoLength: Int

get() = length

} class CatVideo(

name: String,

length: Int,

private val cats: String

) : AbstractVideo(name, length) class Playlist {

private val videos = mutableListOf<AbstractVideo>()

private var currentIndex = 0

fun addVideo(video: AbstractVideo) {

videos.add(video)

}

fun deleteFromPlaylist(name: String) {

val videoToRemove = videos.find { it.videoName == name }

if (videoToRemove != null) {

videos.remove(videoToRemove)

}

}

private fun showNextVideo(video: AbstractVideo){

println("Воспроизведение видео: ${video.videoName}, длина: ${video.videoLength}")

}

fun playPlaylist() {

while (currentIndex < videos.size) {

val video = videos[currentIndex]

showNextVideo(videos[currentIndex])

println("1. Следующее видео")

println("2. Предыдущее видео")

println("3. Выйти из плейлиста")

when (readLine()?.toIntOrNull()) {

1 -> {

currentIndex++

if (currentIndex >= videos.size) {

println("Вы достигли конца плейлиста.")

currentIndex = videos.size - 1

}

}

2 -> {

if (currentIndex > 0) {

currentIndex--

} else {

println("Вы находитесь в начале плейлиста.")

}

}

3 -> {

return

}

else -> println("Неверный выбор. Пожалуйста, выберите снова.")

}

}

currentIndex = 0 // Сброс индекса при достижении конца плейлиста

}

} class DogVideo(

name: String,

length: Int

) : AbstractVideo(name, length) class MusicVideo(

name: String,

length: Int

) : AbstractVideo(name, length)

Пример работы программы представлен на рисунке 1.

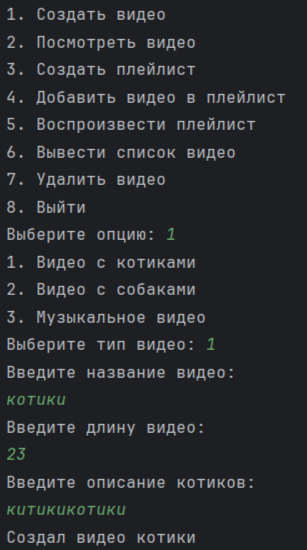


Рисунок 1 – Пример работы программы

Вывод: в ходе лабораторной работы я научилась работать с языком программирования Kotlin, а также применила полученные знания, выполнив разработку приложения с использованием этого языка.