KOTLIN НЕ ДЛЯ ANDROID

АЛЕКСЕЙ ГЛАДКОВ

Mobile Developer

Mobiledeveloper

Mobiledeveloper

Mobiledevelopercourse@Gmail.Com

ОБО МНЕ

- ☑ Автор канала Mobile Developer
- ☑ Mobile Broadcast Expert
- ☑ Android и iOS разработчик > 12 лет

Aurora SDK Setup

If you planning to use Narrow to build apps for Aurora OS, please setup Aurora SDK now, or do it later

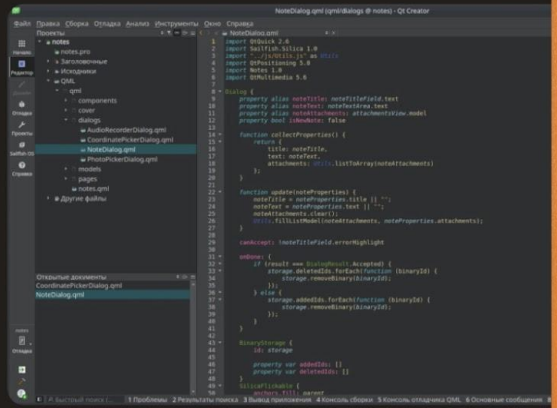
Path to Aurora IDE executable

Enter the path

Path to Aurora project folder

Enter the path

Finish



КАКОЙ СТЕК?

- [x] Kotlin Multiplatform
- [x] Compose Multiplatform
- [x] React

КАКИЕ ПЛАТФОРМЫ?

- [x] Android (Tablets, Phone)
- [x] iOS (Pad, Phone)
- [x] Desktop (Windows, MacOS, Linux)
- [x] Web

ЧТО ТАКОЕ ПЛАТФОРМА?

ЧТО ВХОДИТ В ПЛАТФОРМУ

- [x] Работа с файлами
- [ ] Работа с окнами
- [ ] Работа с безопасностью
- [ ] Работа с другими приложениями
- [ ] Работа с переменными окружения
- [ ] Многое другое

ЧТО ВХОДИТ В ПЛАТФОРМУ

- [x] Работа с файлами
- [x] Работа с окнами
- [ ] Работа с безопасностью
- [ ] Работа с другими приложениями
- [ ] Работа с переменными окружения
- [ ] Многое другое

ЧТО ВХОДИТ В ПЛАТФОРМУ

- [x] Работа с файлами
- [x] Работа с окнами
- [x] Работа с безопасностью
- [ ] Работа с другими приложениями
- [ ] Работа с переменными окружения
- [ ] Многое другое

ЧТО ВХОДИТ В ПЛАТФОРМУ

- [x] Работа с файлами
- [x] Работа с окнами
- [x] Работа с безопасностью
- [x] Работа с другими приложениями
- [ ] Работа с переменными окружения
- [ ] Многое другое

ЧТО ВХОДИТ В ПЛАТФОРМУ

- [x] Работа с файлами
- [x] Работа с окнами
- [x] Работа с безопасностью
- [x] Работа с другими приложениями
- [x] Работа с переменными окружения
- [ ] Многое другое

ЧТО ВХОДИТ В ПЛАТФОРМУ

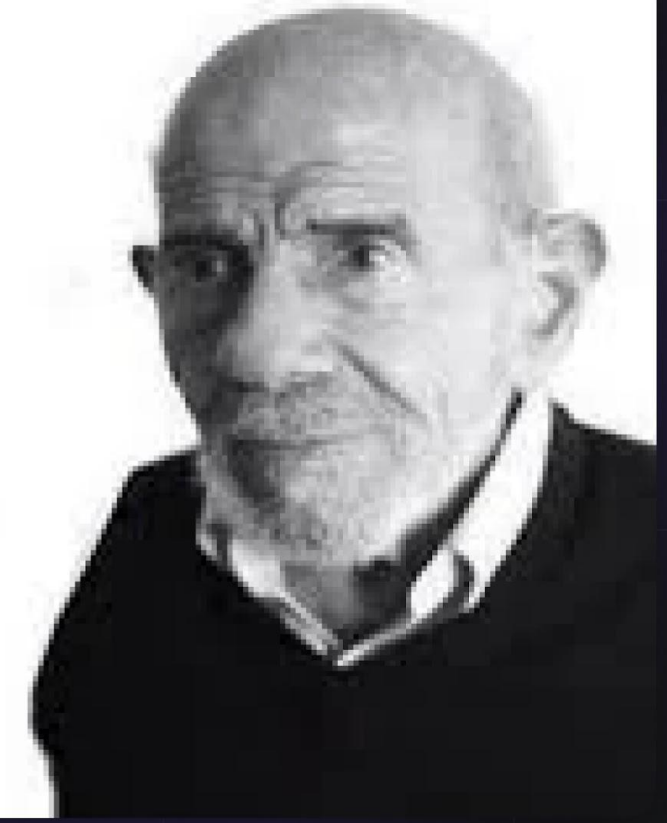
- [x] Работа с файлами
- [x] Работа с окнами
- [x] Работа с безопасностью
- [x] Работа с другими приложениями
- [x] Работа с переменными окружения
- [x] Многое другое


КАКОЙ КЛАСС В ANDROID ОТВЕЧАЕТ
ЗА РАБОТУ С ПЛАТФОРМОЙ?

CONTEXT

А ЧТО В ДРУГИХ ПЛАТФОРМАХ?

ладно



— Жак Фреско —

JAVA SWING

Integration of Compose Multiplatform and Swing

What is covered

In this tutorial, we'll show you how to make the `Swing/Compose` interop work in your application, what its limitations are, what you can achieve with it, in which cases you may use it and when you shouldn't do that.

The main goals of the interoperability between Compose Multiplatform and Swing are

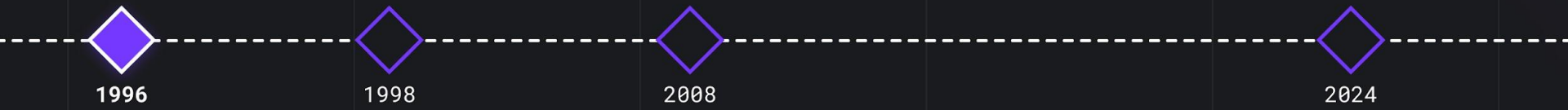
- make it easier and smoother to migrate Swing applications to to Compose
- allow to enhance Compose application with Swing components that don't have 'Compose' analogues

In many cases it is more efficient to implement a missing Component in Compose (and contribute it to community) rather than using a Swing component in a Compose Application.

КРАТКАЯ ИСТОРИЯ JAVA

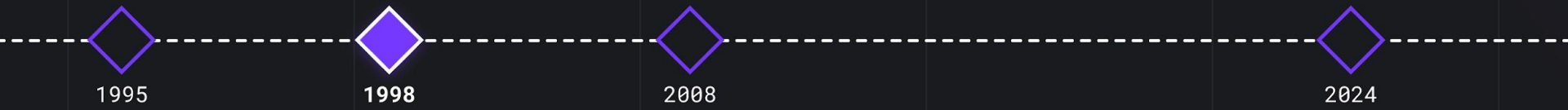
JAVA ABSTRACT WINDOW TOOLKIT

- [x] Появилась в Java 1.0
- [x] Тяжелые объекты и малое количество элементов
- [x] Неконсистентность системных компонентов



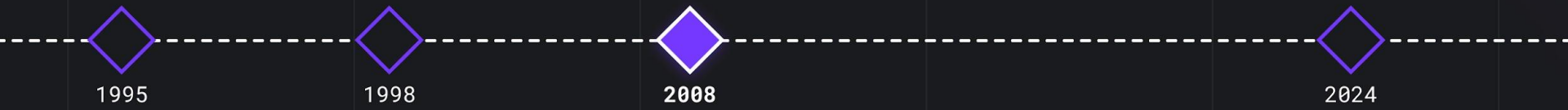
JAVA SWING

- [x] Java 1.2
- [x] Большой набор готовых и легковесных компонентов
- [x] Легкий доступ к платформе
- [x] Низкая производительность в сложных приложениях



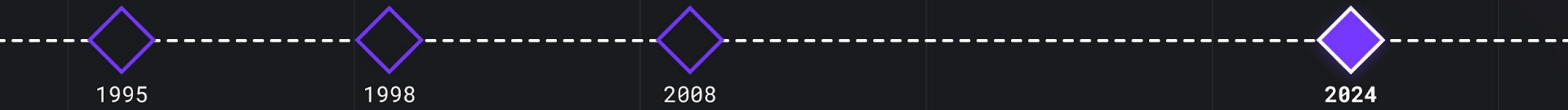
JAVA FX

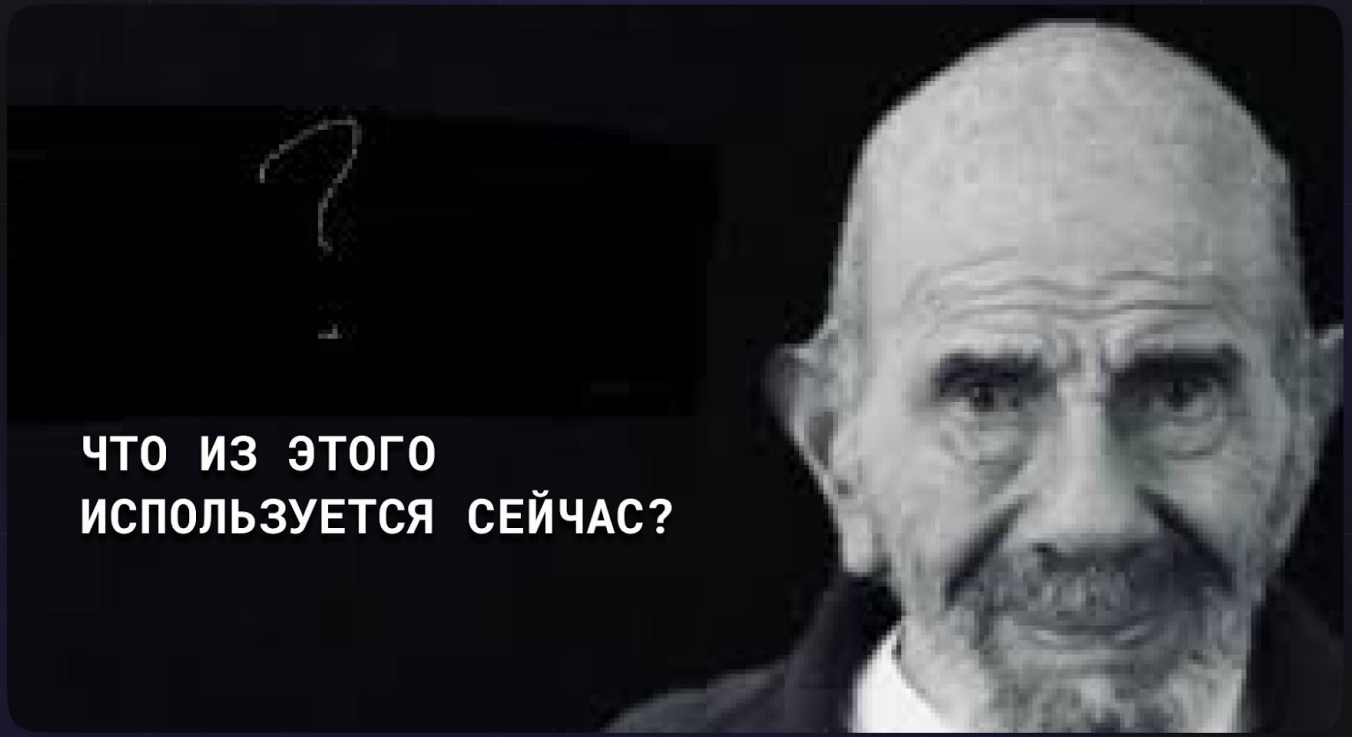
- [x] Был интегрирован в JDK 7u6 в 2012 году
- [x] Поддержка мультимедийных технологий
- [x] Свой мета-язык для разметки
- [x] Поддержка различных новых системных фич



COMPOSE MULTIPLATFORM

- [x] Использует Swing как канвас
- [x] Работа с системой через Java
- [x] Полноценная мультиплатформенность




?
ЧТО ИЗ ЭТОГО
ИСПОЛЬЗУЕТСЯ СЕЙЧАС?


ВСЁ!

SYSTEM

> System - стандартный класс из библиотеки Java, которые позволяет работать с входными и выходными потоками и позволяет работать с переменными окружения и файлами

РАБОТА С ФАЙЛАМИ

```
expect class BaseFileWorker {
    suspend fun readBytes(absolutePath: String): ByteArray
    suspend fun readFile(absolutePath: String): List<String>
    suspend fun createFile(absolutePath: String, fileName: String): Boolean
    suspend fun writeFile(absolutePath: String, content: String)
    suspend fun writeFile(absolutePath: String, content: List<String>)
    suspend fun createCatalog(absolutePath: String): Boolean
    suspend fun checkFileExists(absolutePath: String): Boolean
    suspend fun copyFiles(source: String, to: String)
}
```

```
expect class BaseFileWorker {
    suspend fun readBytes(absolutePath: String): ByteArray
    suspend fun readFile(absolutePath: String): List<String>
    suspend fun createFile(absolutePath: String, fileName: String): Boolean
    suspend fun writeFile(absolutePath: String, content: String)
    suspend fun writeFile(absolutePath: String, content: List<String>)
    suspend fun createCatalog(absolutePath: String): Boolean
    suspend fun checkFileExists(absolutePath: String): Boolean
    suspend fun copyFiles(source: String, to: String)
}
```

JVM

```
val userName = System.getProperty("user.name")
        val userHome = System.getProperty("user.home")
        val osName = System.getProperty("os.name")

        systemType = when {
            osName.contains("windows", true) -> DesktopSystemType.Windows(userHome, userName, appName)
            osName.contains("Linux", true) -> DesktopSystemType.Linux(userHome, userName, appName)
            osName.contains("Mac", true) ->  DesktopSystemType.MacOS(userHome, userName, appName)
            else -> { throw IllegalStateException("This type OS not implemented") }
        }
```

JVM

```
class MacOS(userHome: String, userName: String, appName: String) {
    override fun platformPath(userHome: String, userName: String, appName: String) =
            "/Users/$userName/Library/Application Support/$appName"
}
```

JVM

```
class Linux(userHome: String, userName: String, appName: String) {
    override fun platformPath(userHome: String, userName: String, appName: String) =
                "$userName/$appName"
}
```

JVM

```
class Windows(userHome: String, userName: String, appName: String) {
    override fun platformPath(userHome: String, userName: String, appName: String) =
                "$userHome\\AppData\\Local\\$appName"
}
```

JVM

```
actual fun platformHomeDocsPath(): String {
        val strUserHomePath = System.getProperty("user.home")
        return Paths.get(strUserHomePath, "Documents")
                    .toAbsolutePath()
                    .toString()
    }
```

JVM

```
actual suspend fun createCatalog(absolutePath: String): Boolean {
    val folder = File(absolutePath)
    return withContext(Dispatchers.IO) {
        if (!folder.exists()) {
            folder.mkdirs()
        }

        return@withContext true
    }
}
```

JVM

```
actual suspend fun writeFile(absolutePath: String, content: List<String>) {
    withContext(Dispatchers.IO) {
        val file = File(absolutePath)
        val fileWriter = FileWriter(file, true)
        content.forEach {
            fileWriter.write("$it\n")
        }

        fileWriter.close()
    }
}
```

JVM

```
actual suspend fun readFile(absolutePath: String): List<String> {
    val fileDir = File(absolutePath)
    return fileDir.readLines()
}
```

JVM

```
actual suspend fun createFile(absolutePath: String, fileName: String): Boolean {
    return withContext(Dispatchers.IO) {
        val file = File("$absolutePath/$fileName")
        return@withContext file.createNewFile()
    }
}
```

ЗАПУСК ПРИЛОЖЕНИЙ

JVM

```
actual fun runApp(pathToExecutable: String, args: String): Result<Unit> {
    try {
        Runtime.getRuntime().exec("$pathToExecutable $args")
        return Result.success(Unit)
    } catch (e: Exception) {
        return Result.failure(e)
    }
}
```

JVM

```
actual fun listComposeResUriContent(uri: String, resKlaz: KClass<*>): List<String> {
    val parts = uri.split("!")

    val jarPath = parts
        .firstOrNull()
        ?.replaceFirst("jar:file:/", "")
        ?: return listOf("No jar path")

    val dirPath = parts
        .lastOrNull()
        ?.replaceFirst("/", "")
        ?: return listOf("No dir path")

    val resPackageAndClassName = resKlaz.qualifiedName ?: return listOf("No
resPackageAndClassName")
    val resClassName = resKlaz.simpleName ?: return listOf("No resClassName")
    val prefixToRemove =
            "composeResources/${resPackageAndClassName.replaceFirst(".$resClassName", "")}/"

    val jarUri = URI.create(jarPath)
    val path = jarUri.path
    val file = File(path)
    val jarFile = JarFile(file)

    return jarFile.entries()
                .asSequence()
                .filter { it.name.startsWith(dirPath) }
                .map { it.toString().replaceFirst(prefixToRemove, "") }
                .toList()
}
```

РАБОТА С ОКНАМИ

JVM

```
AppTitleBar(systemType = platformConfiguration.systemType,
            onMinimize = {
                windowState.isMinimized = true
            }, onMaximize = {
                if (windowState.placement == WindowPlacement.Floating) {
                    windowState.placement = WindowPlacement.Maximized
                } else {
                    windowState.placement = WindowPlacement.Floating
                }
            }, onClose = {
                exitApplication()
            }
)
```

```
/*
        TODO: Finish this
        Compose Desktop still not fully support custom app bar,
        especially such things as snap to edge on windows or linux

        Some tutorials on how to do it now
        https://sasikanth.dev/dragging-undecorated-windows-in-compose-desktop/
        https://github.com/JetBrains/compose-multiplatform/tree/master/tutorials/Window_API_new

        Issues with snapping
        https://github.com/JetBrains/compose-multiplatform/issues/3771
        https://github.com/JetBrains/compose-multiplatform/issues/178
        https://stackoverflow.com/questions/77311317/how-to-support-os-window-features-snapping-
animations-with-a-custom-title-ba

        Some current solutions
        https://github.com/amir1376/compose-custom-window-frame/
        https://github.com/ButterCam/compose-jetbrains-theme
        https://github.com/JetBrains/jewel
        https://github.com/JetBrains/jewel/tree/main/decorated-window

        https://stackoverflow.com/questions/61373284/is-there-a-way-to-hide-the-title-bar-but-
keep-the-buttons-in-jframe
        https://medium.com/swlh/customizing-the-title-bar-of-an-application-window-50a4ac3ed27e
        https://stackoverflow.com/questions/12822037/how-to-create-customize-title-bar-with-
close-button-on-jframe
    */
```

```kotlin
/**
 * WindowDraggableArea is a component that allows you to drag the window using the mouse.
 *
 * @param modifier The modifier to be applied to the layout.
 * @param content The content lambda.
 */
@Composable
fun WindowScope.WindowDraggableArea(
    modifier: Modifier = Modifier,
    content: @Composable () -> Unit = {}
) {
    // ...
}
```

```
when (systemType) {
    is PlatformConfiguration.DesktopSystemType.MacOS -> {

    }

    is PlatformConfiguration.DesktopSystemType.Windows -> {

    }

    is PlatformConfiguration.DesktopSystemType.Linux -> {

    }
}
```

```
Row() {
    Spacer(modifier = Modifier.weight(1f).background(Color.Red))
    Icon()
    Icon()
    Icon()
}
```

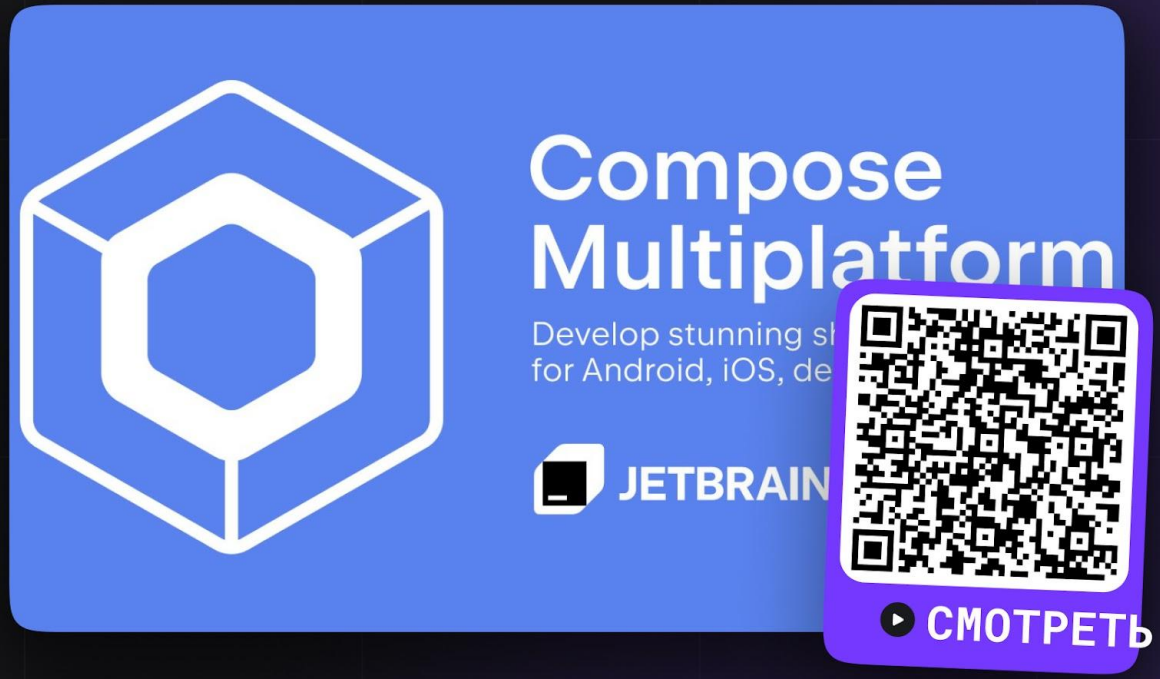
```
val rootPane = window.rootPane
if (platformConfiguration.systemType is PlatformConfiguration.DesktopSystemType.MacOS) {
    rootPane.putClientProperty("apple.awt.fullWindowContent", true)
    rootPane.putClientProperty("apple.awt.transparentTitleBar", true)
    rootPane.putClientProperty("apple.awt.windowTitleVisible", false)
}
```

```
is PlatformConfiguration.DesktopSystemType.MacOS -> {
    Box(
        modifier = Modifier
            .background(NarrowTheme.colors.backgroundPrimary)
            .fillMaxWidth()
            .height(30.dp)
        )
}
```

RELEASE

```
./gradlew packageDeb
./gradlew packageExe
./gradlew runDistributable
./gradlew runReleaseDistributable
./gradlew runRelease
./gradlew notarizeDmg
./gradlew notarizeReleaseDmg
```

```
./gradlew packageDeb
./gradlew packageExe
./gradlew runDistributable
./gradlew runReleaseDistributable
./gradlew runRelease
./gradlew notarizeDmg
./gradlew notarizeReleaseDmg
```


Compose
Multiplatform
Develop stunning sh
for Android, iOS, de
JETBRAIN
СМОТРЕТЬ

DATABASE

```
expect class DbDriverFactory(platformConfiguration: PlatformConfiguration) {
    suspend fun provideDbDriver(
        schema: SqlSchema<QueryResult.AsyncValue<Unit>>,
    ): SqlDriver
}

internal val dbFileName = "narrow.db"
```

```
class DatabaseHolder(private val driverFactory: DbDriverFactory) {
    private var database: Database? = null
    private val mutex = Mutex()

    suspend fun createDatabase() {
        if (database != null) return
        mutex.lock()
        if (database != null) return
        database = Database(driverFactory.provideDbDriver(Database.Schema))
        mutex.unlock()
    }

    fun database(): Database {
        if (database == null) throw IllegalStateException("Database creation must precede getting operation")
        return database!!
    }
}
```

```
// ViewModel
private val databaseHolder = Inject.instance<DatabaseHolder>()

try {
    databaseHolder.createDatabase()
} catch (e: Exception) {
    viewState = viewState.copy(debugMessage = e.message.orEmpty())
}

// Main.kt
if (viewState.isDbReady) {
    Locale.setDefault(Locale.forLanguageTag(viewState.languageItem.tag))
    RootView()
}
```

```
memScoped {
    val errorPtr = alloc<ObjCObjectVar<NSError?>>()
    val fileManager = NSFileManager.defaultManager
    fileManager.createDirectoryAtPath(
        absolutePath,
        withIntermediateDirectories = true,
        attributes = null,
        error = errorPtr.ptr
    )
    errorPtr.value == null
}
```


COCOAPODS
DEPRECATED

RELEASE

ИТОГИ

- [x] Все десктопные ОС разные
- [ ] Нюансы достаточно сложно искать
- [ ] Используйте ChatGPT
- [ ] Настройте pipeline
- [ ] Настройте обфускацию

ИТОГИ

- [x] Все десктопные ОС разные
- [x] Нюансы достаточно сложно искать
- [ ] Используйте ChatGPT
- [ ] Настройте pipeline
- [ ] Настройте обфускацию

ИТОГИ

- [x] Все десктопные ОС разные
- [x] Нюансы достаточно сложно искать
- [x] Используйте ChatGPT
- [ ] Настройте pipeline
- [ ] Настройте обфускацию

ИТОГИ

- [x] Все десктопные ОС разные
- [x] Нюансы достаточно сложно искать
- [x] Используйте ChatGPT
- [x] Настройте pipeline
- [ ] Настройте обфускацию

ИТОГИ

- [x] Все десктопные ОС разные
- [x] Нюансы достаточно сложно искать
- [x] Используйте ChatGPT
- [x] Настройте pipeline
- [x] Настройте обфускацию

СМОТРИ
НА YOUTUBE

ЧИТАЙ
ТЕЛЕГРАМ

СПАСИБО ЗА ВНИМАНИЕ!

АЛЕКСЕЙ ГЛАДКОВ
Mobile Developer

Mobiledeveloper
Mobiledeveloper
Mobiledevelopercourse@Gmail.Com