Практическая работа. Передача данных в фрагментах

Иногда фрагментам для нормальной работы требуется дополнительная информация. Например, если во фрагменте выводятся контактные данные, этот фрагмент должен знать, какой контакт он должен выбрать. Но что, если эта информация должна поступать из другого фрагмента? В этой работе мы воспользуемся вашими знаниями о навигации и применим их для передачи данных между фрагментами.

Цель:

- научится добавлять аргументы к целям навигации, чтобы они могли получать необходимую информацию.
- познакомитесься с плагином `Safe Args` и научитеся использовать его для написания кода, безопасного по отношению к типам.

Git

- git commit -m "Фрагменты"
- создайте и перейдите в новую ветку под именем `encrypt`.

Теоретическая часть:

Необходимо иметь возможность передавать данные между фрагментами. Для этого лучше всего воспользоваться плагином Gradle, который называется `Safe Args`, — дополнительной составляющей компонента `Navigation`. Он предоставляет механизм передачи данных между фрагментами способом, безопасным по отношению к типам. Тем самым предотвращается случайная передача данных неправильного типа, что может привести к ошибкам времени выполнения.

1 Создание и отображение EncryptFragment.

Мы создадим новый фрагмент с именем EncryptFragment, чтобы приложение переходило к нему при щелчке на кнопке Next в макете MessageFragment.

Фрагмент `EncryptFragment` будет использоваться для вывода зашифрованной версии сообщения. Чтобы создать этот фрагмент, выделите пакет `com.hfad.secretmessage` в папке `app/src/main/java`, перейдите в меню `File` и выберите команду `New→Fragment→Fragment (Blank)`. Введите имя фрагмента `«EncryptFragment»` и имя макета `«fragment_encrypt»`. Убедитесь в том, что выбран язык Kotlin, а затем щелкните на кнопке Finish.

Откройте файл макета `fragment_encrypt.xml` и обновите его содержимое:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center_horizontal"
    tools:context=".EncryptFragment">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:textSize="20sp"
        android:text="@string/encrypt_text" />
    <TextView
        android:id="@+id/encrypted_message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:textSize="20sp"/>
</LinearLayout>
```

Также необходимо обновить код Kotlin фрагмента `EncryptFragment` и убедиться в том, что среда Android Studio не добавила лишний код, из-за которого он не сможет делать то, что вам нужно.

Перейдите к пакету `com.hfad.secretmessage` в папке *app/src/main/java* и откройте файл `EncryptFragment.kt`. Замените код, сгенерированный Android Studio, следующим кодом:

```
package com.example.android.secretmessage
import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup

class EncryptFragment : Fragment() {
    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
                              savedInstanceState: Bundle?): View? {
        // Заполнение макета для фрагмента
        return inflater.inflate(R.layout.fragment_encrypt, container, false)
    }
}
```

Включение EncryptFragment в граф навигации

Обновленный код `nav_graph.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/nav_graph"
    app:startDestination="@id/welcomeFragment">
    <fragment
        android:id="@+id/welcomeFragment"
        android:name="com.example.android.secretmessage.WelcomeFragment"
        android:label="fragment_welcome"
        tools:layout="@layout/fragment_welcome" >
        <action
            android:id="@+id/action_welcomeFragment_to_messageFragment"
            app:destination="@id/messageFragment" />
    </fragment>

    <fragment
        android:id="@+id/messageFragment"
        android:name="com.example.android.secretmessage.MessageFragment"
        android:label="fragment_message"
        tools:layout="@layout/fragment_message" >
        <action
            android:id="@+id/action_messageFragment_to_encryptFragment"
            app:destination="@id/encryptFragment" />
    </fragment>

    <fragment
        android:id="@+id/encryptFragment"
        android:name="com.example.android.secretmessage.EncryptFragment"
        android:label="fragment_encrypt"
        tools:layout="@layout/fragment_encrypt" />
</navigation>
```

Фрагмент MessageFragment должен переходить к EncryptFragment

Необходимо сделать так, чтобы при щелчке на кнопке `Next` фрагмент `MessageFragment` переходил к `EncryptFragment`.

Для этого мы добавим к кнопке слушатель `OnClickListener`, который будет включать код навигации.

Чтобы перейти от одного фрагмента к другому, следует получить контроллер навигации и передать ему действие навигации. Контроллер навигации использует это действие для отображения правильного фрагмента.

MessageFragment.kt:

```
package com.example.android.secretmessage
import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Button
import androidx.navigation.findNavController
```

```
class MessageFragment : Fragment()
{
    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
                              savedInstanceState: Bundle?): View? {
        val view = inflater.inflate(R.layout.fragment_message, container, false)
        val nextButton = view.findViewById<Button>(R.id.next)
        nextButton.setOnClickListener{

view.findNavController().navigate(R.id.action_messageFragment_to_encryptFragment)
        }
        return view
    }
}
```

Передача сообщения фрагменту EncryptFragment.

Мы используем `Safe Args` для передачи пользовательского сообщения от `MessageFragment` к `EncryptFragment`. В `EncryptFragment` выводится зашифрованное сообщение.

Для передачи сообщения от `MessageFragment` к `EncryptFragment` можно воспользоваться `Safe Args`. Как упоминалось ранее, это дополнительная часть компонента `Navigation`, которая позволяет передавать аргументы целям способом, безопасным по отношению к типам.

Добавление Safe Args в файлы build.gradle

Откройте файл build.gradle на уровне проекта *SecretMessage/build.gradle* и добавьте путь к классам:

```
buildscript {
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
    }
    dependencies {
        classpath("androidx.navigation:navigation-safe-args-gradle-plugin:2.3.5")
    }
}
```

Замечание: `buildscript` секция должна идти выше чем секция `plugins`

2.2 Добавление плагина в файл build.gradle приложения

Затем необходимо сообщить Gradle, что вы используете плагин `Safe Args`; для этого добавьте строку в файл `build.gradle` на уровне модуля. Откройте файл *SecretMessage/app/build.gradle* и добавьте строку, в раздел plug-ins:

```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
    id 'androidx.navigation.safeargs.kotlin'
}
```

После внесения этих изменений щелкните на ссылке `Sync Now`, появившейся в верхней части редактора кода. Тем самым вы синхронизируете внесенные изменения с оставшейся частью проекта.

После включения плагина `Safe Args` в приложение можно воспользоваться им для передачи сообщения от `MessageFragment` к `EncryptFragment`. Мы сделаем это на следующем шаге.

Фрагмент EncryptFragment должен получать строковый аргумент

Прежде всего необходимо указать, что фрагмент `EncryptFragment` может получать сообщения пользователя. Для этого мы добавим строковый аргумент в этот фрагмент на графе навигации; `MessageFragment` использует эти аргументы для передачи сообщения (строки) фрагменту `EncryptFragment`.

Чтобы добавить аргумент, откройте файл `nav_graph.xml` из *app/src/main/res/nav_graph.xml*. Выберите `EncryptFragment` в визуальном редакторе графа навигации, перейдите на панель `Attributes` и щелкните на кнопке «+» рядом с разделом `Arguments`:

Когда вы щелкаете на кнопке «+», появляется окно `Add Argument`, в котором вводится информация об аргументе. В данном случае фрагмент `EncryptFragment` должен получать аргумент String с сообщением, поэтому присвойте аргументу имя «message», выберите тип «String» и щелкните на кнопке `Add`. Новый фрагмент создается и добавляется в раздел `Arguments` панели `Attributes`:

Обновленный код `nav_graph.xml`

```
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/nav_graph"
    app:startDestination="@id/welcomeFragment">
    <fragment
        android:id="@+id/welcomeFragment"
        android:name="com.example.android.secretmessage.WelcomeFragment"
        android:label="fragment_welcome"
        tools:layout="@layout/fragment_welcome" >
        <action
            android:id="@+id/action_welcomeFragment_to_messageFragment"
            app:destination="@id/messageFragment" />
    </fragment>

    <fragment
```

```
        android:id="@+id/messageFragment"
        android:name="com.example.android.secretmessage.MessageFragment"
        android:label="fragment_message"
        tools:layout="@layout/fragment_message" >
        <action
            android:id="@+id/action_messageFragment_to_encryptFragment"
            app:destination="@id/encryptFragment" />
    </fragment>

    <fragment
        android:id="@+id/encryptFragment"
        android:name="com.example.android.secretmessage.EncryptFragment"
        android:label="fragment_encrypt"
        tools:layout="@layout/fragment_encrypt" >
        <argument
            android:name="message"
            app:argType="string" />
    </fragment>
</navigation>
```

Фрагмент MessageFragment должен передать сообщение EncryptFragment

После того как мы добавили аргумент String в `EncryptFragment`, `MessageFragment` может использовать его для передачи пользовательского сообщения при переходе к этому фрагменту.

Для добавления аргументов к сообщениям навигации можно воспользоваться классом `Directions`.

Safe Args генерирует классы Directions

Классы `Directions` используются для передачи аргументов целям. Когда вы подключаете плагин `Safe Args`, среда Android Studio использует его для генерирования класса `Directions` для каждого фрагмента, от которого может выполняться переход. Так, в приложении `Secret Message` можно переходить от фрагментов `WelcomeFragment` и `MessageFragment` к другим целям, поэтому плагин `Safe Args` генерирует класс `Directions` для каждого из этих фрагментов; для `WelcomeFragment` генерируется класс с именем `WelcomeFragmentDirections`, а для `MessageFragment` — класс с именем `MessageFragmentDirections`:

Обновление кода `MessageFragment.kt`

```
package com.example.android.secretmessage
import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Button
import android.widget.EditText
import androidx.navigation.findNavController
```

```
class MessageFragment : Fragment() {
    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
                              savedInstanceState: Bundle?): View? {
        val view = inflater.inflate(R.layout.fragment_message, container, false)
        val nextButton = view.findViewById<Button>(R.id.next)
        val messageView = view.findViewById<EditText>(R.id.message)
        nextButton.setOnClickListener {
            val message = messageView.text.toString()
            val action = MessageFragmentDirections
                .actionMessageFragmentToEncryptFragment(message)

            view.findNavController().navigate(action)
        }
        return view
    }
}
```

Фрагмент EncryptFragment должен получить значение аргумента

MessageFragment использует аргумент String для передачи сообщения фрагменту `EncryptFragment`. `EncryptFragment` должен получить это значение, чтобы вывести его зашифрованную версию.

Для получения аргументов фрагменты могут использовать класс `Args`. Когда вы подключаете плагин `Safe Args`, Android Studio использует его для генерирования класса `Args` для каждого фрагмента, получающего аргументы.

Например, в приложении `Secret Message` `EncryptFragment` получает аргумент String, поэтому плагин `Safe Args` генерирует для него класс `Args` с именем `EncryptFragmentArgs`.

Каждый класс `Args` включает метод `fromBundle()`, который используется для получения аргументов, переданных фрагменту. Например, в приложении `Secret Message` фрагмент `EncryptFragment` получает аргумент String с именем `message`, поэтому значение этого аргумента может быть получено следующим образом:

```
val message = EncryptFragmentArgs.fromBundle(requireArguments()).message
```

Шифрование сообщения

Для шифрования будет использован метод Kotlin `reversed()`, который просто переставляет буквы String в обратном порядке.

Код выглядит так:

```
val encryptedView = view.findViewById<TextView>(R.id.encrypted_message)
encryptedView.text = message.reversed()
```

Полный код `EncryptFragment.kt`

```
package com.example.android.secretmessage
import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
class EncryptFragment : Fragment() {
    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
                              savedInstanceState: Bundle?): View? {
        val view = inflater.inflate(R.layout.fragment_encrypt, container, false)
        val message = EncryptFragmentArgs.fromBundle(requireArguments()).message
        val encryptedView = view.findViewById<TextView>(R.id.encrypted_message)
        encryptedView.text = message.reversed()
        return view
    }
}
```

Обновление манифеста

Добавьте в файл манифеста AndroidManifest.xml путь до приложения:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.android.secretmessage">
```

Резюме:

- Плагин `Safe Args` — дополнительная часть компонента `Navigation`, обеспечивающая передачу аргументов способом, безопасным по отношению к типам. Он генерирует классы `Directions` и `Args`.
- Классы `Directions` используются для передачи аргументов целям.
- `Safe Args` генерирует класс `Directions` для каждой цели, от которой может выполняться переход.
- Классы `Args` используются для получения аргументов, передаваемых целям.
- `Safe Args` генерирует класс `Args` для каждой цели, получающей аргументы.
- Поведение извлечения в графе навигации используется для вывода целей из стека возврата.

Git

- git commit -m "Передача данных Save Args"