



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра Математического обеспечения и стандартизации информационных технологий

Отчет по выполнению практического задания №5

Тема:

Создание динамического интерфейса с применением фрагментов

Дисциплина:

Разработка мобильных приложений

Выполнил:

Студент группы ИКБО-03-20

Цемкало А.Р.

Принял:

Доцент кафедры МОСИТ ИИТ

Чернов Е. А.

Москва 2022 г.

СОДЕРЖАНИЕ

Ход работы.....	3
Вывод.....	9

Ход работы

1. Реализовать создание класса фрагмента.

```
public class ArticleFragment extends Fragment {  
    private static final String ARG_POSITION = "position";  
    private int mPosition = -1;  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
                             Bundle savedInstanceState) {  
        if (savedInstanceState != null) {  
            mPosition = savedInstanceState.getInt(ARG_POSITION);  
        }  
        return inflater.inflate(R.layout.fragment_article, container, attachToRoot: false);  
    }  
}
```

Рисунок 1 – Скриншот с кодом, реализующим создание класса фрагмента

2. Реализовать добавление фрагмента в XML разметку явлений.

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="horizontal">  
  
    <fragment android:name="com.example.task5.HeadlinesFragment"  
        android:id="@+id/headlines_fragment"  
        android:layout_weight="1"  
        android:layout_width="0dp"  
        android:layout_height="match_parent"/>  
  
    <fragment android:name="com.example.task5.ArticleFragment"  
        android:id="@+id/article_fragment"  
        android:layout_weight="2"  
        android:layout_width="0dp"  
        android:layout_height="match_parent" />  
  
</LinearLayout>
```

Рисунок 2 – Скриншот с кодом разметки двух фрагментов для экранов в горизонтальном положении (файл layout-land/news_articles.xml)

3. Реализовать создание гибкого интерфейса пользователя.

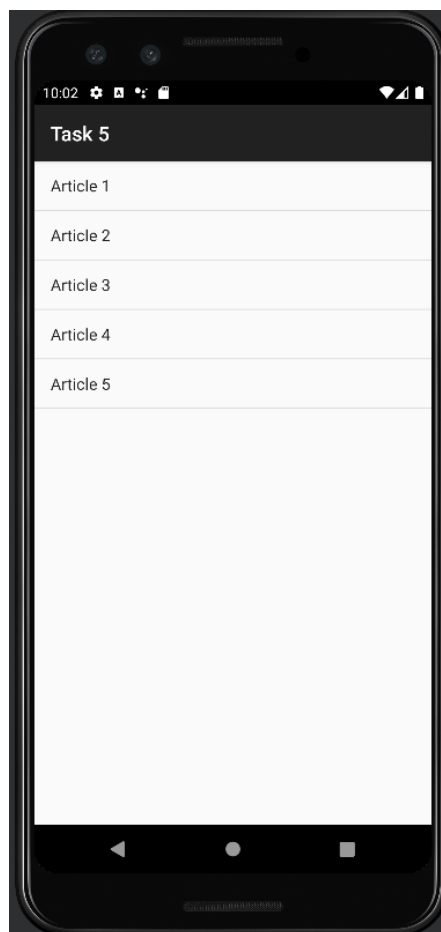


Рисунок 3 – Вид работающего приложения на маленьком экране

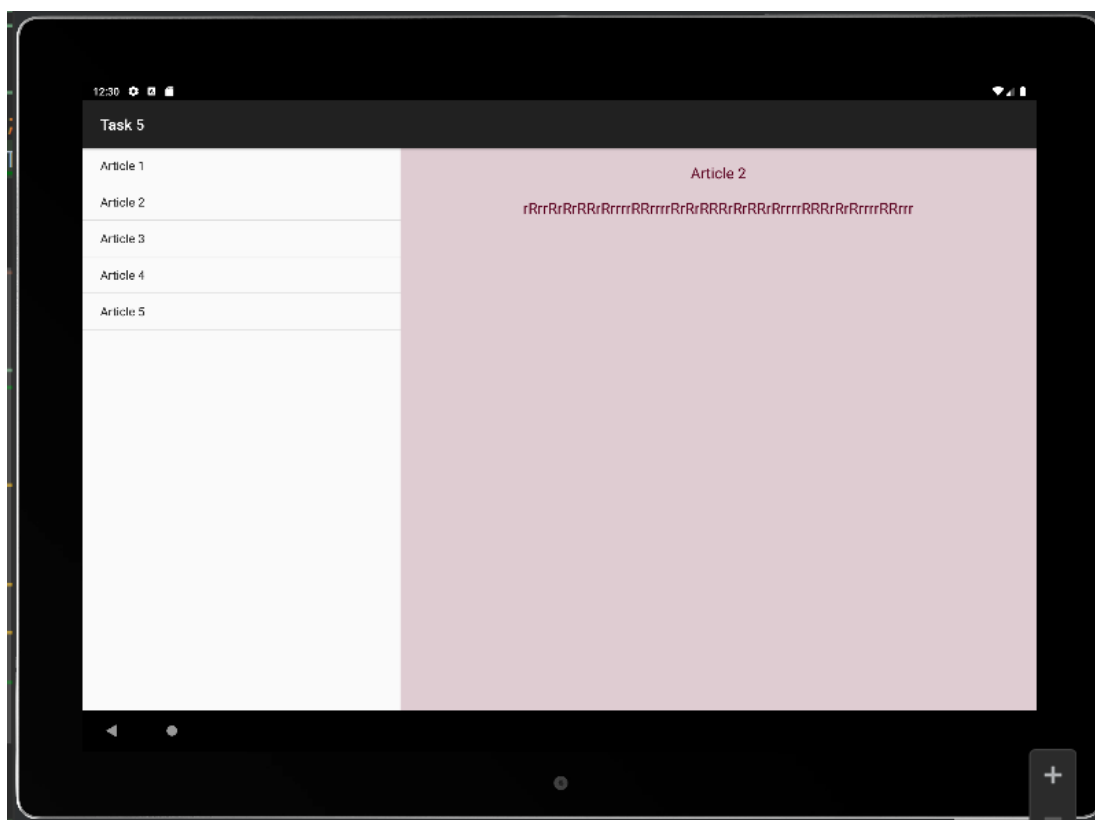


Рисунок 4 – Вид работающего приложения на большом экране

4. Реализовать добавление фрагмента в явление во время выполнения.

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/fragment_container"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</FrameLayout>
```

Рисунок 5 – Скриншот с кодом файла layout /news_articles.xml для маленьких экранов

```
public class MainActivity extends AppCompatActivity implements HeadlinesFragment.OnHeadlineSelectedListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.news_articles);

        if (findViewById(R.id.fragment_container) != null) {
            if (savedInstanceState != null) {
                return;
            }
            HeadlinesFragment firstFragment = new HeadlinesFragment();
            firstFragment.setArguments(getIntent().getExtras());
            getSupportFragmentManager().beginTransaction()
                .add(R.id.fragment_container, firstFragment).commit();
        }
    }
}
```

Рисунок 6 – Скриншот с кодом добавления фрагмента в предыдущую разметку

5. Реализовать замену одного фрагмента другим.

```
@Override
public void onArticleSelected(int position) {
    ArticleFragment articleFragment = (ArticleFragment) getSupportFragmentManager()
        .findFragmentById(R.id.article_fragment);
    if (articleFragment != null) {
        articleFragment.updateArticleView(position);
    } else {
        ArticleFragment newFragment = ArticleFragment.newInstance(position);
        FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();
        transaction.replace(R.id.fragment_container, newFragment);
        transaction.addToBackStack(null);
        transaction.commit();
    }
}
```

Рисунок 7 – Скриншот метода, обеспечивающего замену одного фрагмента другим

```

public void updateArticleView(int position) {
    TextView article = (TextView) getActivity().findViewById(R.id.article_text);
    if (position < DataBase.Articles.length) {
        article.setText(DataBase.Articles[position]);
        mPosition = position;
    }
}
}

```

Рисунок 8 – Скриншот метода, обеспечивающего замену содержимого фрагмента ArticleFragment

6. Реализовать определение интерфейса взаимодействия фрагмента с явлением.

```

OnHeadlineSelectedListener mCallback;

public interface OnHeadlineSelectedListener {
    void onArticleSelected(int position);
}

@Override
public void onAttach(Context context) {
    super.onAttach(context);
    Activity activity = getActivity();

    try {
        mCallback = (OnHeadlineSelectedListener) activity;
    } catch (ClassCastException e) {
        throw new ClassCastException(activity.toString() +
            " must implement OnHeadlineSelectedListener");
    }
}
}

```

Рисунок 9 – Скриншот интерфейса взаимодействия фрагмента с явлением

7. Реализовать интерфейс взаимодействия.

```

@Override
public void onArticleSelected(int position) {
    ArticleFragment articleFragment = (ArticleFragment) getSupportFragmentManager()
        .findFragmentById(R.id.article_fragment);
    if (articleFragment != null) {
        articleFragment.updateArticleView(position);
    } else {
        ArticleFragment newFragment = ArticleFragment.newInstance(position);
        FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();
        transaction.replace(R.id.fragment_container, newFragment);
        transaction.addToBackStack(null);
        transaction.commit();
    }
}
}

```

Рисунок 10 – Скриншот реализации интерфейса взаимодействия

8. Реализовать доставку сообщения фрагменту.

```

@Override
public void onListItemClick(ListView l, View v, int position, long id) {
    mCallback.onArticleSelected(position);
    getListView().setItemChecked(position, value: true);
}

```

Рисунок 11 – Скриншот реализации доставки сообщения фрагменту

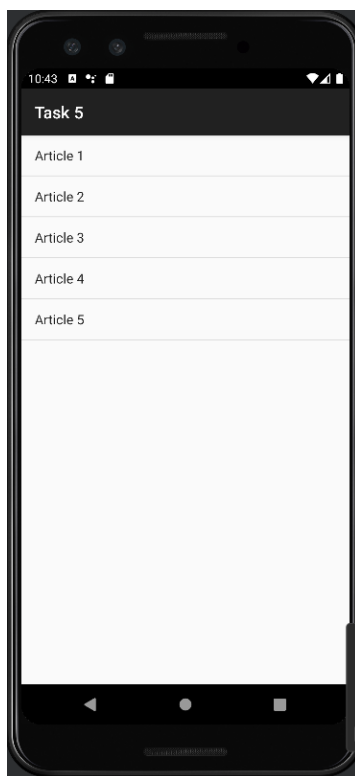


Рисунок 12 – Вид приложения до нажатия на элемент списка

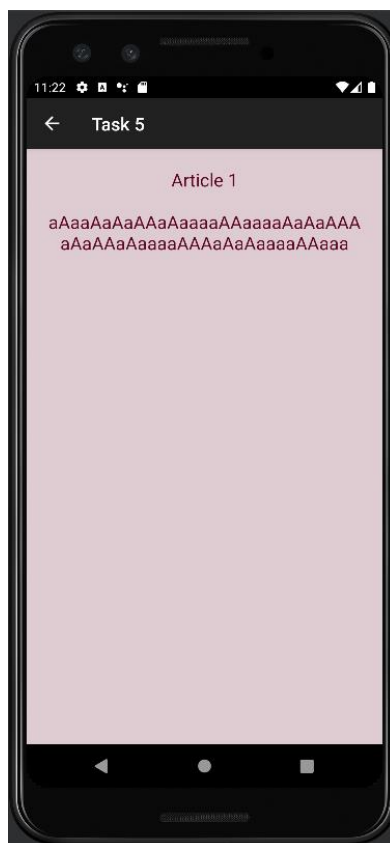


Рисунок 13 – Вид приложения после нажатия на первый элемент списка

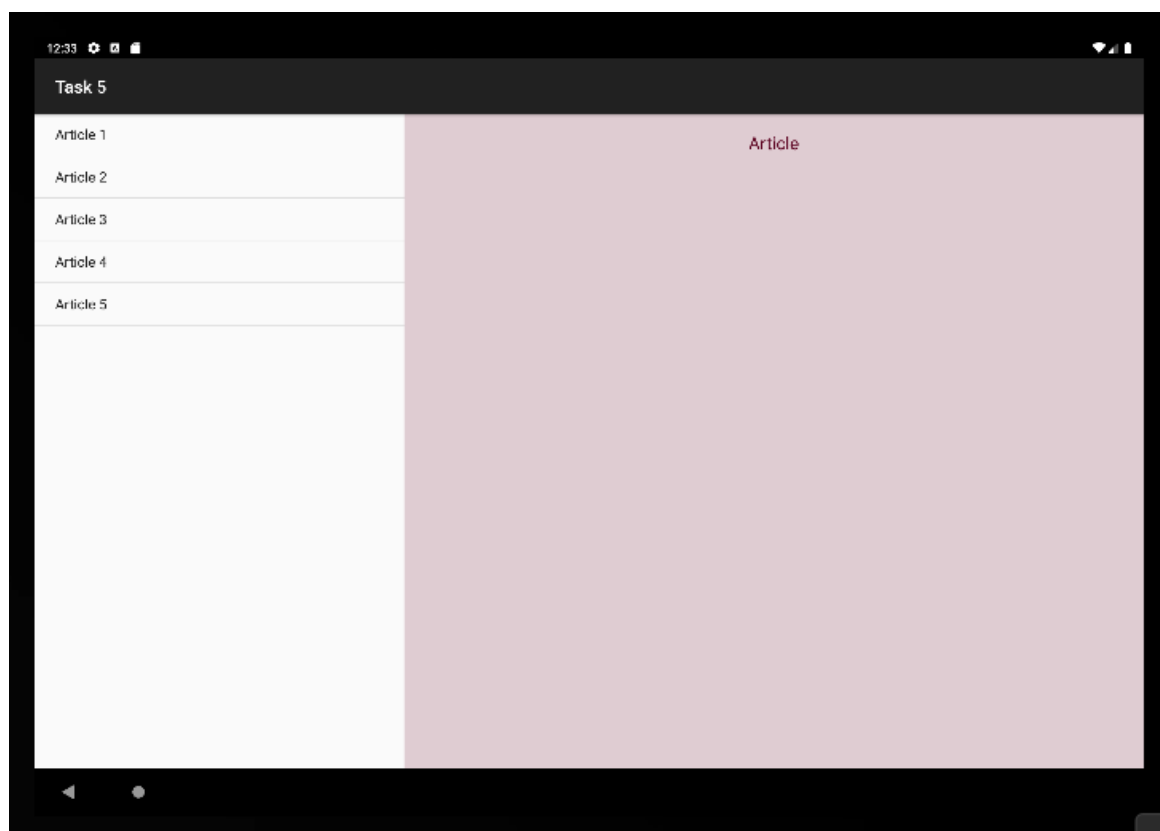


Рисунок 14 – Вид приложения на большом экране до нажатия на элемент списка

