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

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практической работе №7**

по дисциплине «Разработка мобильных приложений»

|  |  |
| --- | --- |
| **Выполнили:**  Студенты группыИКБО-12-22 | Солобай А.П. |
| **Проверил:**  Преподаватель | Степанов П.В. |

Москва 2024 г.

# Выполнение практической работы

Ссылка на GitHub со всеми работами: ***https://github.com/Eckorezze/Mobil***

1. Создать приложения, которое использует в файле Манифест атрибуты

android:allowBackup, android:icon, android:roundIcon, android:label,

android:theme и реализует активити, которая должна быть входной

точкой в приложение и не должна получать какие-либо данные извне и представлять стартовый экран.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 1 – использование требуемых атрибутов

2. С помощью атрибутов элемента manifest определить версию

созданного приложения и его кода.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2 – определение версий приложения и кода

3. Используя атрибуты произвести установку версии SDK.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 3 – установка версий SDK

4. Реализовать установку разрешений к определенным ресурсам.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 4 – установка разрешений к ресурсам

5. Реализовать поддержку разных разрешений, через ограничения

использования приложения для определенных разрешений экранов.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 5 – поддержка разных разрешений

6. Реализовать запрет на изменение ориентации экрана.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 6 – запрет на изменение ориентации

7. Добавить вторую активити, реализовать переход от одной активити к

другой.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 7 – переход от одной активити к другой + объявление в AndroidManifest

8. Реализовать передачу данных между Activity. В зависимости от типа

отправляемых данных при их получении можно использовать ряд

методов объекта Bundle: get(), getString(), getInt(), getByte(),… и т.д.

Изображение выглядит как текст, снимок экрана, мультимедиа, программное обеспечение

Автоматически созданное описание

Рисунок 8 – передача данных между активити.

9. Реализовать передачу сложных объектов.

*Листинг 1 – код в файле PracticeActivity*

|  |
| --- |
| Bundle arguments = getIntent().getExtras();  SUS sus;  if (arguments != null) {  sus = (SUS) arguments.getSerializable(SUS.class.getSimpleName());  assert sus != null;  Snackbar.make(findViewById(R.id.root), sus.Name + " " + sus.Company + " " + sus.Age, Snackbar.LENGTH\_SHORT).show(); } |

*Листинг 2 – код в файле TestActivity*

|  |
| --- |
| SUS sussy = new SUS("Скуф", "Яндекс", 35 ); intent.putExtra(SUS.class.getSimpleName(), sussy); |

Изображение выглядит как электроника, гаджет, текст, снимок экрана

Автоматически созданное описание

Рисунок 9 – передача сложных объектов между активити.

10. Реализовать интерфейс Parcelable.

*Листинг 3 – реализация через Parcelable*

|  |
| --- |
| SUS sussy; if(arguments!=null){  sussy = arguments.getParcelable(SUS.class.getSimpleName());  textView.setText("Name: " + sussy.getName() + "\nCompany: " + sussy.getCompany() + "\nAge: " + String.valueOf(sussy.getAge())); } |

Изображение выглядит как текст, снимок экрана, смартфон

Автоматически созданное описание

Рисунок 10 – передача сложных объектов через Parcelable.

11.Реализовать получение результата из Activity. Регистрация функции

для получения результата. Запуск activity для получения результата.

Практическое применение Activity Result API.

*Листинг 4 – код в PracticeActivity*

|  |
| --- |
| protected void onCreate(Bundle savedInstanceState) {   super.onCreate(savedInstanceState);  binding = PracticeActivityBinding.inflate(getLayoutInflater());  View view = binding.getRoot();  setContentView(view);   Bundle extras = getIntent().getExtras();  if (extras != null) {  TextView ageView = findViewById(R.id.ageView);  String age = extras.getString(TestActivity.AGE\_KEY);  ageView.setText("Возраст: " + age);  } }  public void onCancelClick(View v) {  setResult(RESULT\_CANCELED);  finish(); } public void onButton1Click(View v) {  sendMessage("Доступ разрешен"); } public void onButton2Click(View v) {  sendMessage("Доступ запрещен"); } public void onButton3Click(View v) {  sendMessage("Недопустимый возраст"); } private void sendMessage(String message){  Intent data = new Intent();  data.putExtra(TestActivity.ACCESS\_MESSAGE, message);  setResult(RESULT\_OK, data);  finish(); } |

*Листинг 5 – код в TestActivity*

|  |
| --- |
| static final String AGE\_KEY = "AGE"; static final String ACCESS\_MESSAGE="ACCESS\_MESSAGE"; ActivityResultLauncher<Intent> mStartForResult =  registerForActivityResult(new ActivityResultContracts.StartActivityForResult(),  new ActivityResultCallback<ActivityResult>() {  @Override  public void onActivityResult(ActivityResult result) {  TextView textView = findViewById(R.id.textView);  if(result.getResultCode() == Activity.RESULT\_OK){  Intent intent = result.getData();  assert intent != null;  String accessMessage =  intent.getStringExtra(ACCESS\_MESSAGE);  textView.setText(accessMessage);  }  else{  textView.setText("Ошибка доступа");  }  }  }); public void onClick(View view) {  *// получаем введенный возраст* EditText ageBox = findViewById(R.id.age);  String age = ageBox.getText().toString();  Intent intent = new Intent(this, PracticeActivity.class);  intent.putExtra(AGE\_KEY, age);  mStartForResult.launch(intent); } |

Изображение выглядит как электроника, текст, снимок экрана, мультимедиа

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, улыбка

Автоматически созданное описание

Рисунок 11,12 – тестирование кода для Activity Result API.

12.Реализовать взаимодействие между четырьмя Activity. Реализовать

управление стеком activity.

*Листинг 6 – код в PracticeActivity*

|  |
| --- |
| binding.toggleButton.setOnClickListener(v -> {   Intent intent = new Intent(PracticeActivity.this, TestActivity.class);  intent.addFlags(Intent.FLAG\_ACTIVITY\_REORDER\_TO\_FRONT);  intent.addFlags(Intent.FLAG\_ACTIVITY\_NO\_HISTORY);  startActivity(intent); });  binding.toggleButton2.setOnClickListener(v -> {   Intent intent = new Intent(PracticeActivity.this, MainActivity.class);  intent.addFlags(Intent.FLAG\_ACTIVITY\_REORDER\_TO\_FRONT);  startActivity(intent); }); |

*Листинг 7 – код в TestActivity*

|  |
| --- |
| binding.toggleButton.setOnClickListener(v -> {   *// Запускаем MainActivity* Intent intent = new Intent(TestActivity.this, MainActivity.class);  intent.addFlags(Intent.FLAG\_ACTIVITY\_REORDER\_TO\_FRONT);  startActivity(intent); });  binding.toggleButton2.setOnClickListener(v -> {  Intent intent = new Intent(TestActivity.this, PracticeActivity.class);  intent.addFlags(Intent.FLAG\_ACTIVITY\_REORDER\_TO\_FRONT);  startActivity(intent); }); |

# Вывод

В ходе работы мне удалось научиться работать с файлом манифеста и реализовывать взаимодействие между активностями разными способами с передачей простых и сложных данных.