Частное учреждение образования
Колледж бизнеса и права

УТВЕРЖДАЮ
Заведующий
методическим кабинетом
_____Е.В. Паскал
«___»_____2021

Специальность: 2-40 01 01 «Программное обеспечение информационных технологий»	Дисциплина: «Основы кроссплатформенного программирования»

ЛАБОРАТОРНАЯ РАБОТА № 17
Инструкционно-технологическая карта

Тема: «Мобильная разработка»

Цель: Научиться использовать средства языка Java для мобильной разработки.

Время выполнения: 4 часа

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить теоретические сведения;
2. Ответить на контрольные вопросы;
3. Откомпилировать примеры программ из раздела «Теоретические сведения»;
4. Выполнить ИДЗ.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ ДЛЯ ВЫПОЛНЕНИЯ РАБОТЫ

Язык программирования Java является универсальным и может выполняться на множестве платформ. Одна из новых платформ, появившихся в последние годы, стала чрезвычайно популярной среди программистов на Java. Речь идет об Android, которая начинала свое триумфальное шествие как операционная система для смартфонов, а затем была перенесена на другие устройства. Данная система предназначена для выполнения программ, написанных на Java. Эти программы, называемые мобильными приложениями, пишутся на мобильной платформе с открытым исходным кодом, которая полностью бесплатна для разработчиков. Любой желающий может создавать, развертывать и продавать приложения Android.

Приложения Android -это обычные программы на Java, которые используют каркас приложения, представляющий собой базовый набор классов и файлов, присущих всем мобильным приложениям. Каркас строится

по определенному набору правил структурирования, обеспечивающих корректное выполнение приложений на устройствах Android. Для создания приложений нужно установить среду Android Studio. Перейдите на сайт https://developer.android.com, загрузите установочный файл, запустите его и следуйте инструкциям, чтобы установить программу на компьютере. Чтобы начать работу в Android Studio, выполните следующие действия.

1. Запустите Android Studio.

2. В появившемся диалоговом окне выберите параметр Start a New Android Project (Создать новый проект Android). На экране появится окно мастера Create New Project (Создание нового проекта).

3. В поле Application name (Имя приложения) введите HelloWorld.

4. Можно принять заданное по умолчанию значение поля Project location (Местоположение проекта) (подпапка папки AndroidStudioProj есть в вашей пользовательской папке) либо выбрать другую папку с помощью кнопки ..., находящейся справа от поля.

5.Щелкните на кнопке Next (Далее).

6.Каждому проекту Android назначается целевая платформа. В качестве такой платформы выбирается старейшая версия Android, на которой может выполняться приложение. Поскольку каждая новая версия Android включает расширенные средства, результат выбора целевой Платформы определяет доступные средства. В качестве цели выберите Phone and Tablet (Телефон и Планшет), а в качестве минимальной версии SDK - API 16 (либо более раннюю версию, если эта недоступна).

7.Щелкните на кнопке Next. На Следующем этапе нужно задать операцию, выполняемую приложением. Выберите Fullscreen Activity (Полноэкранная операция) и щелкните на кнопке Next.

8.В поле Activity Name (Имя операции) введите SalutonActivity. Это приведет к изменению значений полей Layout Name (Имя компоновки) и Title (Название). Сохраните эти изменения, щелкнув на кнопке Next.

9.Щелкните на кнопке Finish (Готово). В результате будет создано новое приложение, а на панели Package Explorer (Браузер проектов) появится элемент HelloWorld.

Создание приложений для Android

Структура нового проекта Android Новый проект Android состоит примерно из двадцати файлов и папок, которые имеют одинаковую структуру для любого приложения Android. В зависимости от возможностей приложения количество файлов может быть большим, но структура файлов и папок остается неизменной. На рис. 24.2 показана панель Projects (Проекты) Android Studio сразу же после создания новою проекта. Если эта панель не отображается, щелкните на вкладке Projects, находящейся вдоль левого края окна IDE.

Исследуем структуру файлов и папок проекта. Новое приложение HelloWorld включает Следующие компоненты.

• /javar. Корневая папка, в которой находится исходный код приложения Java.

• /src/com.java24hours.HelloWorld/SalutonActivity.java. Класс, предназначенный для операции, которая выполняется по умолчанию при запуске приложения.

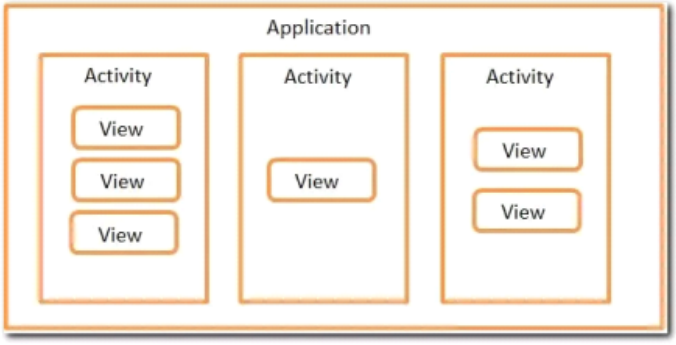
• /res. В этой папке находятся ресурсы приложения, такие как файлы данных, файлы компоновки, графика и анимация. Здесь же находятся подпапки для определенных типов ресурсов: layout, values, drawable и mipmap.

• AndroidManifest. xml. Основной файл конфигурации приложения.

Эти файлы и папки образуют каркас приложения. Первое, что должен освоить разработчик приложений Android, - научиться изменять каркас. Благодаря этому можно лучше узнать о назначении каждою компонента. Для изменения каркаса нужно включить в проект дополнительные файлы. Создание приложения на этом этапе вы можете успешно запустить на выполнение новый проект Android, поскольку каркас сам по себе является работоспособным приложением. Однако вряд ли запуск каркаса представляет какой-либо интерес. Поэтому мы сначала настроим приложение HelloWorld, чтобы отобразить наше традиционное приветствие "Hello World! ".

Если проводить аналогию с Windows, то приложение состоит из окон, называемых Activity. В конкретный момент времени обычно отображается одно Activity и занимает весь экран, а приложение переключается между ними. В качестве примера можно рассмотреть почтовое приложение. В нем одно Activity – список писем, другое – просмотр письма, третье – настройки ящика. При работе вы перемещаетесь по ним. Макет активности по умолчанию определяет два виджета (widgets): RelativeLayout и TextView. Виджеты представляют собой структурные элементы, из которых составляется пользовательский интерфейс. Виджет может выводить текст или графику, взаимодействовать с пользователем или размещать другие виджеты на экране. Кнопки, текстовые поля, флажки — все это разновидности виджетов. Android SDK включает множество виджетов, которые можно настраивать для получения нужного оформления и поведения. Каждый виджет является экземпляром класса View или одного из его субклассов (например, TextView или Button).

Содержимое Activity формируется из различных компонентов, называемых View.



Необходимо заметить, что View обычно размещаются в ViewGroup. Самый распространенный пример ViewGroup – это Layout. Layout бывает различных типов и отвечает за то, как будут расположены его дочерние View на экране (таблицей, строкой, столбцом …).

Обычно при работе приложения на экране дисплея находятся элементы управления, которые являются доступными для манипулирования экранными объектами.

Виджет – это элемент управления, являющийся объектом класса View, который служит интерфейсом для взаимодействия с пользователем. Условно элементы управления можно разделить на четыре основные категории:

- командные элементы управления, применяемые для выполнения функций;
- элементы выбора, позволяющие выбирать данные или настройки;
- элементы ввода, применяемые для ввода данных;
- элементы отображения, используемые для вывода.

Командные элементы управления выполняют некоторые действия. Главным командным элементом является кнопка Button, которая обладает множеством вариантов отображения. Действие выполняется сразу после нажатия на кнопку.

Элементы выбора позволяют пользователю выбрать из группы допустимых объектов тот, с которым будет совершено действие. Элементы выбора применяются также для действий по настройке. Распространенными элементами выбора являются флажки и списки. Щелкнув по флажку, пользователь немедленно увидит появившуюся галочку. Элементы управления типа «список» позволяют осуществлять выбор из конечного множества текстовых строк, каждая из которых представляет команду, объект или признак. Пользователь может выбрать единственную строку текста, нажав на нее.

Элементы ввода дают пользователю возможность не только выбирать существующие сведения, но и вводить новую нформацию. Самые простые элементы – виджет TextView, который отображает текст без возможности его редактирования, и виджет EditText, позволяющий редактирование текста (поле ввода). Для отображения графики используется виджет ImageView. В эту категорию попадают также такие элементы управления, как счетчики и ползунки.

Элементы управления отображением используются для управления визуальным представлением информации на экране. Типичными примерами элементов отображения являются разделители и полосы прокрутки. Сюда же входят разделители страниц, линейки, направляющие, сетки и рамки.

Каждому объекту нужно задать размеры, координаты, цвет, текст и т.д. Android поддерживает способ, основанный на XML-разметке, который напоминает разметку веб-страницы. Можно использовать и визуальный способ перетаскивания объектов с помощью мыши.

Файлы XML-разметки находятся в папке res/layout проекта. Папка содержит ресурсы, не связанные с кодом. Кроме разметки, там же содержатся изображения, звуки, строки для локализации и т.д.

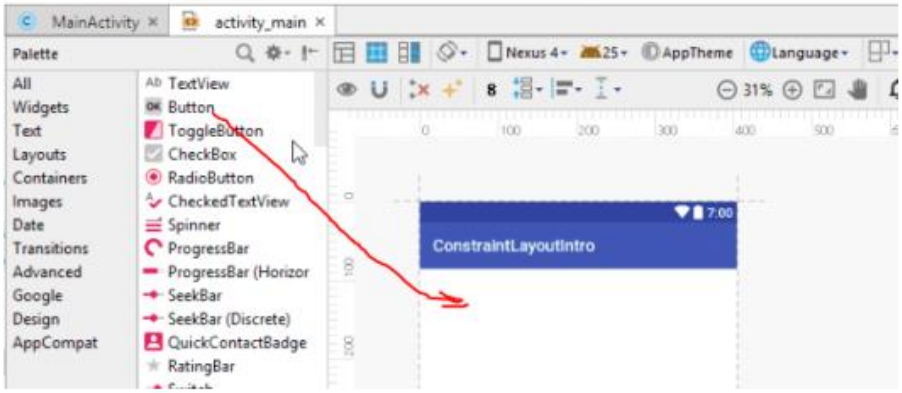
Чтобы размещать на экране различные компоненты (кнопки, поля ввода, чекбоксы и т.п.), необходимо использовать специальный контейнер, в котором будут помещаться компоненты. В Android компоненты называются View, а контейнер - ViewGroup. Существуют несколько типов ViewGroup: LinearLayout, RelativeLayout, FrameLayout, TableLayout, ConstraintLayout и т.д. Они различаются тем, как они будут упорядочивать компоненты внутри себя. LinearLayout, например, выстроит их по горизонтальной или вертикальной линии, TableLayout - в виде таблицы.

Рассмотрим контейнер ConstraintLayout. Слово Constraint переводится как ограничение, принуждение, привязка. Класс ConstraintLayout является родительским для классов ViewGroup и TextView.

Код на языке xml, находящийся в файле activity_main.xml для «пустого» приложения, которое выводит в середине экрана текст ―Hello Word!‖:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:app="http://schemas.android.com/apk/res-auto"
 android:layout_width="match_parent"
 android:layout_height="match_parent">
 <TextView
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="Hello World!"
 app:layout_constraintBottom_toBottomOf="parent"
 app:layout_constraintLeft_toLeftOf="parent"
 app:layout_constraintRight_toRightOf="parent"
 app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

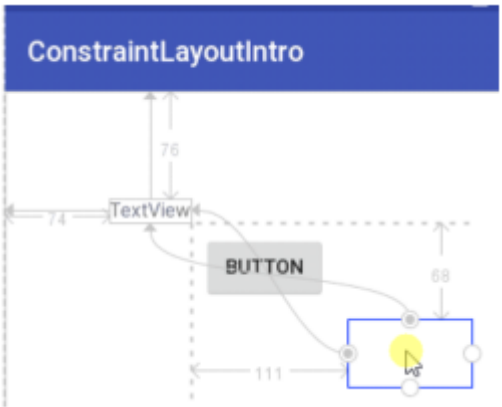
Для размещения виджета в нужном месте экрана воспользуемся привязкой. Разместите не экране какой-нибудь виджет, например TextView. Сделать это можно перетаскиванием мышью соответствующего компонента из палитры Palette



В запущенном приложении виджет окажется не в том месте, где его разместили, а в верхнем левом углу (координаты x=0, y=0). Чтобы его закрепить в желаемом месте надо добавить привязки (constraints). Они будут задавать положение View на экране относительно каких-либо других элементов или относительно родительского View. Если выделить на экране виджет TextView, то можете видеть 4 круга по его бокам.

Эти круги используются, чтобы создавать привязки. Существует два типа привязок: одни задают положение View по горизонтали, а другие - по вертикали.

Создадим горизонтальную привязку. Привяжем положение TextView к левому краю его родителя. Родителем TextView этом примере является ConstraintLayout, который занимает весь экран. Поэтому края ConstraintLayout совпадают с краями экрана. Чтобы создать привязку, нажмите мышкой на TextView, чтобы выделить его. Затем зажмите левой кнопкой мыши левый кружок и тащите его к левой границе. TextView также уехал влево. Он привязался к левой границе своего родителя. Не обязательно, чтобы они должны быть вплотную.Мы можем задать отступ. Для этого просто зажмите левой кнопкой мыши TextView, перетащите вправо и отпустите. Обратите внимание на число, которое меняется. Это величина отступа TextView от объекта, к которому он привязан (в нашем случае - от левой границы родителя). Аналогично можно делать вертикальную привязку, после чего наш виджет окажется в нужном месте.



Можно привязывать не только к границам родителя, но и к другим View. Сделайте привязку кнопки Button к TextView. Для этого разместите виджет кнопки на экране, выделите ее мышью, коснитесь указателем мыши кружка в левой ее части (кружок станет зеленым). Не отпуская левую кнопку мыши, проведите указатель к правому кружку виджета TextView. Когда этот кружок станет зеленым, отпустите кнопку мыши. Теперь виджет Button оказался привязанным к виджету TextView по горизонтали. Аналогично

сделайте привязку по вертикали, проведя линию от верхнего кружка виджета Button до нижнего кружка виджета TextView. Перемещая Button, увидем линии со стрелками и в них относительные координаты виджета Button.

Рассмотрим процесс разработки простого приложения, содержащего типичный набор виджетов. Создадим обработчик событий на примере Button.

Откройте созданный автоматически проект, содержащий пустую Activity. Когда разметка открыта в графическом представлении, то слева от основной части редактора кода можно увидеть панель инструментов, в которой сгруппированы различные элементы по категориям: Widgets, Texts, Layouts и т.д. В группе Images найдите элемент ImageButton, перетащите его на форму и отпустите. Далее необходимо выбрать изображение для кнопки.

Во вкладке Component Tree выберем элемент Constraint Layout (экран). В панели свойств Properties отобразятся самые употребительные свойства выбранного компонента. К ним относятся идентификатор, ширина и высота. Выбираем View all properties (две стрелочки), чтобы открыть все свойства компонента. Найдите свойство background. Щелкните рядом с этим словом во второй колонке. Появится текстовое поле, в которое можно ввести значение вручную, и кнопка с тремя точками, которая запустит диалоговое окно для создания ресурса. Переходим на вкладку Color и выбираем нужный цвет.

Далее можно поменять картинку для графической кнопки. Находим подходящее изображение и копируем его в папку res/drawable проекта. Затем выделяем элемент ImageButton на форме и в панели свойств выбираем свойство srcCompat. Снова щелкаем на кнопке с тремя точками и выбираем ресурс в категории Drawable - там должен появиться ресурс с именем добавленного ранее файла. Там же в окне свойств находим свойство onClick; прописываем имя метода для обработки нажатия на кнопку onClick.

Установите курсор мыши внутри текста ―onClick‖ у кнопки (переключившись в режим Text) и нажмите комбинацию Alt+Enter. Во всплывающем окне выбрать вариант Create ―onClick(View)‖ in ―MainActivity‖. В коде класса MainActivity появится заготовка для обработки щелчка кнопки. Необходимо набрать следующий текст:

```
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity {
private TextView mHelloTextView;
@Override
protected void onCreate(Bundle savedlnstanceState) {
super.onCreate(savedlnstanceState);
setContentView(R.layout.activitymain);
mHelloTextView=(TextView) fmdViewById(R.id.textView);
}
public void onClick(View view) {
mHelloTextView.setText("Hello!”);
```

```
}
}
```

Далее необходимо запустить программу в эмуляторе Android, нажать на созданную кнопку и наблюдать за изменением текста на экране.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие возможности имеет среда разработки приложений Android Studio?
2. Этапы создания нового проекта?
3. В каких папках хранятся файлы проекта?
4. Что содержится в манифесте проекта?

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Разработать графический интерфейс приложения «Калькулятор валют»
2. Разработать графический интерфейс приложения «Калькулятор цены автомобиля с учетом возможных опций»
3. Разработать графический интерфейс приложения «Калькулятор цены компьютера» с учетом цен на комплектующие
4. Разработать графический интерфейс приложения «Калькулятор затрат на автомобильные запчасти»
5. Разработать графический интерфейс приложения «Калькулятор стоимости ремонта автомобиля»
6. Разработать графический интерфейс приложения «Калькулятор затрат на ремонт дома»
7. Разработать графический интерфейс приложения «Калькулятор стоимости оплат за оказанные услуги ЖКХ для квартиры многоэтажного дома»
8. Разработать графический интерфейс приложения «Калькулятор перевода расстояния пройденного пути мили-километры»
9. Разработать графический интерфейс приложения «Калькулятор перевода веса килограммы-фунты»
10. Разработать графический интерфейс приложения «Калькулятор перевода температуры по шкалам Цельсия-Фаренгейта Кельвина»
11. Разработать графический интерфейс приложения «Калькулятор расчета стоимости печати фотографий разного размера»
12. Разработать графический интерфейс приложения «Калькулятор стоимости заказанных блюд в кафе»
13. Разработать графический интерфейс приложения «Калькулятор»
14. Разработать графический интерфейс приложения «Калькулятор стоимости домашнего блюда»

ДОМАШНЕЕ ЗАДАНИЕ

Индивидуальное задание

ЛИТЕРАТУРА

У. Савитч, язык Java. Курс программирования, Вильямс, 2017.

Преподаватель А.С.Кибисова

Рассмотрено на заседании цикловой комиссии программного обеспечения информационных технологий
Протокол № ____ от «___» _____ 2021
Председатель ЦК _____В.Ю.Михалевич