

Форма проведения зачета

Всего приведено вопросов 82. Отвечаем **устно**. Один студент тянет **один** вопрос из приведенных в данном списке. Вопросы не повторяются. Ответ предполагает **краткое пояснение** по указанному вопросу, объемом чаще всего в одно (max 2-3) предложение. На обдумывание дается одна минута.

Если ответа нет или он неправильный, то допускается **вторая попытка** – но вытягивать надо будет 2 вопроса.

Вопросы по курсу МПДИС ДН 2020:

1. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ДЛЯ СОЗДАНИЯ ANDROID-ПРИЛОЖЕНИЙ

1) Что такое SDK? (расшифровка, определение)

SDK (Software Development Kit)

SDK – это набор средств разработки, позволяющий программистам создавать приложения под определенную архитектуру. Дает разработчику широкий набор средств, включая средства отладки и справочные материалы. SDK распространяется (как в случае с Android) часто бесплатно. Это помогает платформе обзаводиться разнообразными приложениями за счет сторонних разработчиков.

2) Что такое IDE? (расшифровка, определение)

IDE (Integrated development environment)

IDE - интегрированная среда разработки

Включает набор программных средств, таких как текстовый редактор (чаще всего с подсветкой синтаксиса и подсказками), компилятор или интерпретатор, отладчик и средства для автоматизации сборки. Часто поддерживается интеграция с внешним программным обеспечением. В IDE есть инструменты для быстрого построения и визуального редактирования графического интерфейса программы. Таким образом, интегрированная среда разработки призвана объединить различные инструменты в один программный комплекс.

3) Что необходимо для разработки под Android?

1. Java SDK (JDK)

Java SDK включает JRE (Java Runtime Environment) и JDK (Java Development Kit). JRE представляет среду выполнения, в которой запускается и выполняется программный код на языке Java. JRE содержит различные стандартные библиотеки классов Java, компилятор `javac`, служебные утилиты.

2. Среда разработки + Android SDK

Существуют различные IDE для Java-разработки.

4) Какие существуют средства для разработки под Android? (Android development tools)

1. Android Studio

2. AVD Manager - Android Virtual Device

3. Gradle - a build system for Android apps

4. IntelliJ IDEA

5. Eclipse

2. ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ ПОД ANDROID

5) Назовите основные составляющие Android приложения.

- Activities
- Services
- Broadcast Receivers
- Content Providers.

6) Что такое Activity?

Activities – представляет интерфейс пользователя и обработку событий экранных компонентов;

7) Что такое Service?

Services – представляет множество фоновых процессов;

8) Что такое Broadcast Receiver?

Broadcast Receivers – этот слой обеспечивает взаимодействие пользовательских программ и операционной системы;

9) Что такое Content Provider?

Content Providers – обеспечивает работу с данными (базами данных).

10) Назовите другие (неосновные) составляющие Android приложения?

- Fragments – представляют часть Activity
- Views – визуальные компоненты (кнопки, списки, поля ввода и др.)
- Layouts – определяют размещение Views на экране
- Intents – компоненты, срабатывающие при поступлении сообщений
- Resources – внешние ресурсы (например, картинки).

11) Что входит в дерево проекта Android приложения? (содержание, назначение)

Дерево проекта содержит следующие папки

- src - исходные файлы проекта
- gen - .R файл со ссылками на ресурсы проекта
- bin - результат компиляции проекта
- layout - описание визуального интерфейса
- values - xml-файлы с описанием ресурсов, используемых в проекте

12) Первоначальное содержимое класса Activity. (при создании пустого проекта)

Ответ:

```
public class MainActivity extends Activity
{
    @Override
```

```

public void onCreate(Bundle savedInstanceState)
{
    super(savedInstanceState);
    setContentView(R.layout.activity_main);
}
}

```

13) Что делает onCreate()?

Метод **onCreate()** вызывается при создании или перезапуска активности. Система может запускать и останавливать текущие окна в зависимости от происходящих событий. Внутри данного метода настраивают статический интерфейс активности. Метод onCreate() принимает объект Bundle, содержащий состояние пользовательского интерфейса, сохранённое в последнем вызове обработчика onSaveInstanceState.

14) Что делает setContentView(R.layout.activity_main)?

В метод setContentView() передается ресурс разметки графического интерфейса. Именно здесь и решается, какой именно визуальный интерфейс будет иметь MainActivity. Но что в данном случае представляет ресурс R.layout.activity_main? Это файл activity_main.xml из папки res/layout (в принципе можно заметить, что название ресурса соответствует названию файла), который также по умолчанию открыт в Android Studio.

3. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

15) С помощью каких объектов создаются все элементы интерфейса пользователя в приложении Android?

Графический интерфейс пользователя представляет собой иерархию объектов android.view.View и android.view.ViewGroup. (xml файлы).

16) Для чего нужен объект View?

Простые объекты View представляют собой элементы управления и прочие виджеты, например, кнопки, текстовые поля и т.д., через которые пользователь взаимодействует с программой.

17) Для чего нужен объект ViewGroup?

Каждый объект ViewGroup представляет контейнер, который содержит и упорядочивает дочерние объекты View. В частности, к контейнерам относят такие элементы, как RelativeLayout, LinearLayout, GridLayout, ConstraintLayout и ряд других.

18) Как загрузить файл XML макета?

Поместить его в res/layout.

19) Назовите 10 Android View.

- TextView
- EditText
- Image
- View (бэкграунд)
- SearchView

- Button
- RadioButton
- Spinner (вылетает менюшка)
- CheckBox
- Ratingbox
- SeekBar
- ProgressBar

20) У любого объекта View может быть связанный с ним целочисленный идентификатор. Для чего он нужен? Как он задается?

Id() -- идентификатор, которым Android управляет в фоновом режиме, - это уникальный номер, чтобы узнать этот элемент. Ваш идентификатор в файле XML-это имя, которое вы даете этому элементу, чтобы ссылаться на него в своем коде, его дружественное имя к этому элементу.

android:id

21) Что такое wrap_content и match_parent (fill_parent)? Для каких параметров задаются эти значения?

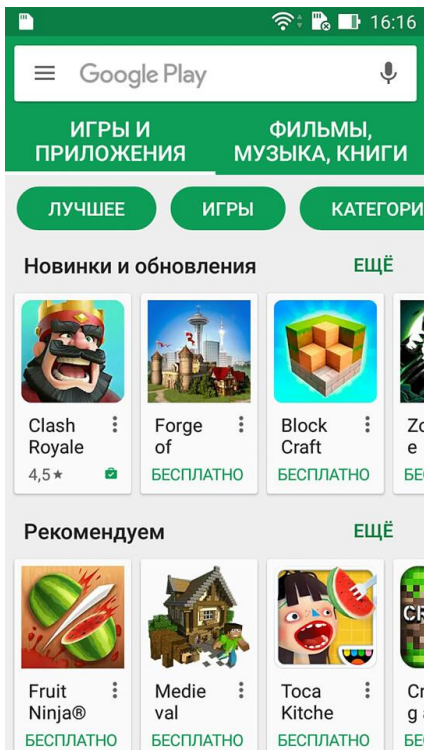
Параметр **Layout weight/ height**.

- match_parent (fill_parent) – означает, что элемент займет всю доступную ему в родительском элементе ширину/высоту.
- wrap_content – ширина/высота элемента будет определяться его содержимым

22) Назовите как минимум три вида вариантов размещения элементов на экране? (layouts)

- **LinearLayout** – отображает View-элементы в виде одной строки (если он Horizontal) или одного столбца (если он Vertical).
- **TableLayout** – отображает элементы в виде таблицы, по строкам и столбцам.
- **RelativeLayout** – для каждого элемента настраивается его положение относительно других элементов.
- **AbsoluteLayout** – для каждого элемента указывается явная позиция на экране в системе координат (x,y)

23) Посмотрите на картинку и укажите где и какие элементы интерфейса вы видите?



4. ЖИЗНЕННЫЙ ЦИКЛ ANDROID ПРИЛОЖЕНИЯ

24) Что представляет собой жизненный цикл Android приложения?

Все приложения Android имеют строго определенный системой жизненный цикл. При запуске пользователем приложения система дает этому приложению высокий приоритет. Каждое приложение запускается в виде отдельного процесса, что позволяет системе давать одним процессам более высокий приоритет, в отличие от других. Благодаря этому, например, при работе с одними приложениями Android позволяет не блокировать входящие звонки. После прекращения работы с приложением, система освобождает все связанные ресурсы и переводит приложение в разряд низкоприоритетного и закрывает его.

25) Назовите все методы жизненного цикла Android приложения.

- protected void onCreate()
- protected void onStart()
- protected void onRestart()
- protected void onResume()
- protected void onPause()
- protected void onStop()
- protected void onDestroy()

26) Обязательный для реализации метод жизненного цикла?

- onCreate - первый метод, с которого начинается выполнение activity. В этом методе activity переходит в состояние Created. Этот метод обязательно должен быть определен в классе activity. В нем производится первоначальная настройка activity. В частности, создаются объекты визуального интерфейса.

27) Обязательным при реализации любого из методов жизненного цикла является ...
Наличие android.app.Activity.

28) Какие выделяют циклы, которые можно отслеживать в классе активности?

- entire lifetime
- visible lifetime
- active(foreground) lifetime

29) Расскажите про полное время жизни (entire lifetime).

Время с момента первого вызова метода onCreate() до вызова onDestroy().

30) Расскажите про видимое время жизни (visible lifetime).

Время между вызовом метода onStart() и вызовом onStop().

31) Расскажите про активное время жизни (foreground lifetime).

Время между вызовами onResume() и onPause().

32) Расскажите про onCreate().

onCreate - первый метод, с которого начинается выполнение activity. В этом методе activity переходит в состояние Created. Этот метод обязательно должен быть определен в классе activity. В нем производится первоначальная настройка activity. В частности, создаются объекты визуального интерфейса. Этот метод получает объект Bundle, который содержит прежнее состояние activity, если оно было сохранено. Если activity заново создается, то данный объект имеет значение null. Если же activity уже ранее была создана, но находилась в приостановленном состоянии, то bundle содержит связанную с activity информацию.

33) Расскажите про onStart().

В методе onStart() осуществляется подготовка к выводу activity на экран устройства. Как правило, этот метод не требует переопределения, а всю работу производит встроенный код. После завершения работы метода activity отображается на экране, вызывается метод onResume, а activity переходит в состояние Resumed.

34) Расскажите про onRestart().

onRestart -> onStart -> onResume

Если окно возвращается в приоритетный режим после вызова onStop(), то в этом случае вызывается метод onRestart(). Т.е. вызывается после того, как активность была остановлена и снова была запущена пользователем. Всегда сопровождается вызовом метода onStart().

onRestart предшествует вызовам метода onStart() (кроме самого первого). Используйте его для специальных действий, которые должны выполняться только при повторном запуске активности в рамках «полноценного» состояния.

35) Расскажите про onResume().

При вызове метода onResume activity переходит в состояние Resumed и отображается на экране устройства, и пользователь может с ней взаимодействовать. И собственно activity остается в этом

состоянии, пока она не потеряет фокус, например, вследствие переключения на другую activity или просто из-за выключения экрана устройства.

36) Расскажите про onPause().

Если пользователь решит перейти к другой activity, то система вызывает метод onPause.

- В этом методе можно освобождать используемые ресурсы, приостанавливать процессы, например, воспроизведение аудио, анимаций, останавливать работу камеры (если она используется) и т.д., чтобы они меньше сказывались на производительность системы.
- После выполнения этого метода activity уходит из фокуса, но она все еще активна. И если пользователь решит вернуться к этой activity, то система вызовет снова метод onResume, и activity снова появится на экране.

37) Расскажите про onStop().

В этом методе activity переходит в состояние Stopped. В методе onStop следует освобождать используемые ресурсы, которые не нужны пользователю, когда он не взаимодействует с activity.

- При этом во время состояния Stopped activity остается в памяти устройства, сохраняется состояние всех элементов интерфейса.
- Если после вызова метода onStop пользователь решит вернуться к прежней activity, тогда система вызовет метод onStart. Если же activity вовсе завершила свою работу, например, из-за закрытия приложения, то вызывается метод onDestroy().

38) Расскажите про onDestroy().

Завершается работа активности вызовом метода onDestroy, который возникает либо, если система решит убить activity, либо при вызове метода finish().

5. ОБЪЕКТЫ INTENT И ФИЛЬТРЫ ОБЪЕКТОВ INTENT

39) Что такое Intent?

intent(намерение) - механизм описания одной операции (выбрать фотографию, отправить письмо, сделать звонок, запустить браузер и перейти по указанному адресу). В Android-приложениях многие операции работают через намерения. Объект обмена сообщениями, с помощью которого можно запросить выполнение действия у компонента другого приложения.

40) Назовите три основные ситуации, когда используется Intent.

- для запуска операции.
- для запуска службы.
- для рассылки широковещательных сообщений.

41) Расскажите как используется Intent для запуска операции.

- Для запуска нового экземпляра Activity необходимо передать объект Intent методу **startActivity()**. Объект Intent описывает действие, которое требуется выполнить, а также содержит все необходимые данные.
- Если после окончания действия требуется получить результат, то вызывается метод **startActivityForResult()**. Операция получит результат в виде отдельного объекта Intent в обратном вызове метода **onActivityResult()**

42) Расскажите как используется Intent для запуска службы.

Сервис можно запустить для выполнения однократного действия, передав объект Intent методу **startService()**. Объект Intent описывает сервис, который требуется запустить, а также содержит все необходимые данные.

43) Расскажите как используется Intent для рассылки широковещательных сообщений.

Для выдачи широковещательных сообщений другим приложениям необходимо передать объект Intent методу **sendBroadcast()**, **sendOrderedBroadcast()** или **sendStickyBroadcast()**.

44) Что такое явный объект Intent?

Явные объекты Intent указывают компонент, который требуется запустить, по имени (полное имя класса).

45) Что такое неявный объект Intent?

Не содержат имени конкретного компонента. Вместо этого они в целом объявляют **действие**, которое требуется выполнить, что дает возможность компоненту из другого приложения обработать этот запрос.

46) Какие основные сведения содержатся в объекте Intent?

- Имя компонента
- Действие
- Данные
- Категория
- Дополнительные штуки вида ключ-значения

6. СЕРВИСЫ (СЛУЖБЫ) В ANDROID (SERVICES)

47) Для чего нужен сервис (служба)?

Компонентом приложения, который может выполнять длительные операции в фоновом режиме и не имеет пользовательского интерфейса.

48) Назовите примеры операций, которые может выполнять сервис (служба)?

- обрабатывать сетевые транзакции,
- воспроизводить музыку,
- выполнять ввод-вывод файла
- взаимодействовать с поставщиком контента

49) Какие формы может принимать сервис (служба)? Каким методом вызывается каждая из форм?

- Запущенная -- **startService**
- Привязанная -- **bindService**

50) Что такое запущенный сервис (служба) ?

- Обычно запущенная служба выполняет одну операцию и не возвращает результатов вызывающему компоненту.
- Когда операция выполнена, служба должна остановиться самостоятельно.

51) Что такое привязанный сервис (служба)?

- Привязанная служба предлагает интерфейс клиент-сервер, который позволяет компонентам взаимодействовать со службой, отправлять запросы, получать результаты и даже делать это между разными процессами посредством межпроцессного взаимодействия (IPC).

52) Назовите методы жизненного цикла сервиса (службы)?

- onStartCommand() возвращает значения:
 - START_NOT_STICKY (сервис остается в остановленном положении)
 - START_STICKY (сервис снова возвращается в запущенное состояние как если бы снова был бы вызван метод onStartCommand() без передачи в этот метод объекта Intent)
 - START_REDELIVER_INTENT (StartCommand() с передачей в этот метод объекта Intent)
- onCreate()
- onDestroy()
- onBind().

53) Расскажите про onStartCommand()

- Система вызывает этот метод, когда другой компонент, например, операция, запрашивает запуск этой службы, вызывая startService().
- После выполнения этого метода служба запускается и может в течение неограниченного времени работать в фоновом режиме.
- Если вы реализуете такой метод, вы обязаны остановить службу посредством вызова stopSelf() или stopService(). (Если требуется только обеспечить привязку, реализовывать этот метод не обязательно).

54) Расскажите про onBind().

Система вызывает этот метод, когда другой компонент должен выполнить привязку к службе (например, для выполнения удаленного вызова процедуры) путем вызова bindService().

55) Расскажите про onCreate().

- Система вызывает этот метод при первом создании службы для выполнения однократных процедур настройки (перед вызовом onStartCommand() или onBind()).
- Если служба уже запущена, этот метод не вызывается.

56) Расскажите про onDestroy().

Система вызывает этот метод, когда служба более не используется и выполняется ее уничтожение. Служба должна реализовать методы для очистки ресурсов, таких как потоки, зарегистрированные приемники и т. д.

57) Как сервис (служба) объявляются в манифесте?

Все службы, как и операции (и другие компоненты), должны быть объявлены в файле манифеста приложения. Чтобы объявить службу, необходимо добавить элемент <service>, в качестве дочернего элемента <application> . Например:

<manifest ... > ...

<application ... >

<service android:name=".ExampleService" /> ...

</application> </manifest>

7. ПРИЕМНИКИ ШИРОКОВЕЩАТЕЛЬНЫХ СООБЩЕНИЙ (BROADCAST RECEIVERS)

58) Что такое приёмник широковещательных сообщений?

Это компонент для получения внешних событий и реакции на них. Инициализировать передачи могут другие приложения или службы. BroadcastReceiver

59) Назовите виды сообщений, которые могут быть получены приёмником широковещательных сообщений. Расскажите о каждом виде.

- нормальные сообщения о намерениях (Normal broadcasts) Посылаются вызовом метода context.sendBroadcast() и являются полностью асинхронными.
- порядковые сообщения о намерениях (Ordered broadcasts) Посылаются методом Context.sendOrderedBroadcast(). Эти сообщения посылаются одному получателю за один раз.

60) Перечислите методы жизненного цикла приёмников широковещательных сообщений.

Расскажите про них.

- Когда широковещательное сообщение прибывает для получателя сообщения, Android вызывает его методом **onReceive()** и передаёт в него объект Intent, содержащий сообщение. Приёмник широковещательных сообщений является активным только во время выполнения этого метода.

61) Для чего используются приёмники широковещательных сообщений.

- системных событий, таких как уровень зарядки батареи,
- сетевые подключения,
- входящие звонки,
- изменения часового пояса,
- состояние подключения данных,
- входящие сообщения SMS,
- обращения по телефону

62) Как зарегистрировать приёмник широковещательных сообщений.

- либо в коде активности,
- либо в манифесте приложения при помощи фильтра намерений, чтобы определить, какие сообщения приёмник должен прослушивать.

8. МАНИФЕСТ ПРИЛОЖЕНИЯ

63) Что такое файл манифест

Файл манифеста AndroidManifest.xml предоставляет основную информацию о программе системе.

64) Для чего нужен файл манифест?

Файл манифеста инкапсулирует всю архитектуру Android-приложения, его функциональные возможности и конфигурацию. В процессе разработки приложения вам придется постоянно редактировать данный файл, изменяя его структуру и дополняя новыми элементами и атрибутами.

65) Назовите элементы структуры файла манифеста. (все называть не обязательно)

<permission>

<uses-permission>

<instrumentation>

<uses-sdk>

<application>

<activity>

<intent-filter>

66) Расскажите про <uses-permission>.

Элемент <uses-permission> запрашивает разрешение, которые приложению должны быть предоставлены системой для его нормального функционирования. Разрешения предоставляются во время установки приложения, а не во время его работы.

67) Расскажите про <uses-sdk>.

Элемент <uses-sdk> позволяет объявлять совместимость приложения с указанной версией (или более новыми версиями API) платформы Android. Уровень API, объявленный приложением, сравнивается с уровнем API системы мобильного устройства, на который устанавливается данное приложение.

68) Расскажите про <activity>.

Элемент <activity> объявляет активность. Если активность не объявлена в манифесте, она не будет видна системе и не будет запущена при выполнении приложения или будет выводиться сообщение об ошибке.

Вопросы помимо курса:

69) Назовите принципы ООП.

- инкапсуляция
- абстракция
- наследование
- полиморфизм

70) Что такое наследование? Как оно реализовано в Java?

Наследование (англ. inheritance) - это механизм, позволяющий создавать классы на основе другого класса.

```
class Class1 extends Class2{}
```

71) Что такое полиморфизм? Как он реализован в Java?

Полиморфизм — это способность программы идентично использовать объекты с одинаковым интерфейсом без информации о конкретном типе этого объекта.
через `@Override`.

72) Что такое инкапсуляция?

Инкапсуляция (encapsulation) — это сокрытие реализации класса и отделение его внутреннего представления от внешнего (интерфейса).

73) Как реализуется инкапсуляция?

`public, private, protected, default(package-private)`

74) Что такое абстракция?

Абстра́кция — это использование только тех характеристик **объекта**, которые с достаточной точностью представляют его в данной системе.

75) Как реализуется абстракция?

Через Abstract class OR Interface

76) Что делает `super()`?

Ключевое слово **super** можно использовать для вызова конструктора суперкласса и для обращения к члену суперкласса, скрытому членом подкласса.

77) Что делает `super.callMethod()`?

Выражение **super.callMethod()** обращается к методу **callMethod()** из базового класса.

Вопросы помимо всего прочего:

78) На какой кафедре вы обучаетесь?

ИТАС

79) Как зовут заведующего кафедрой?

Толя

80) Как расшифровывается предмет?

Мобильные приложения для информационных систем.

81) Как зовут преподавателя?

Кузнецова Ольга Владимировна

82) Любимый вопрос преподавателя?

Как дела?