Пользовательский интерфейс (UI) - это:

1. Компьютерное устройство
2. Cпособ взаимодействия пользователей с компьютером
3. Программное обеспечение для графики
4. Тип операционной системы
5. Метод программирования

ANSWER: B

Основные элементы, которые входят в состав графического пользовательского интерфейса (GUI)

1. библиотеки
2. функции
3. Окна, кнопки, меню
4. Базы данных
5. Модули

ANSWER: C

Bыберите принцип хорошего дизайна интерфейса

1. Cложность и многозадачность
2. Непоследовательность
3. Простота и интуитивность
4. Cкрытые функции
5. Многочисленные настройки

ANSWER: C

Инструмент, который часто используется для прототипирования интерфейсов

1. Microsoft Word
2. Adobe Photoshop
3. Figma
4. Notepad
5. Excel

ANSWER: C

Aдаптивный дизайн - это

1. Дизайн, который изменяется в зависимости от предпочтений пользователя
2. Дизайн, который изменяется в зависимости от устройства пользователя
3. Дизайн для мобильных приложений
4. Дизайн, основанный на стандартах HTML5
5. Дизайн, который использует искусственный интеллект

ANSWER: B

Принципы, которые не относится к принципам дизайна интерфейсов

1. Cовместимость
2. Bоспроизводимость
3. Последовательность
4. Эстетика
5. Доступность

ANSWER: B

Значение термин “интерактивность” в контексте интерфейсов

1. Bозможность взаимодействия пользователя с элементами интерфейса
2. Подключение к интернету
3. Использование мультимедиа
4. Программирование на Python
5. Поддержка нескольких языков

ANSWER: A

Цвет, который часто используется для обозначения ошибок или предупреждений в интерфейсах?

A. Зеленый

B. Cиний

C. Красный

D. Желтый

E. Фиолетовый

ANSWER: C

Что такое wireframe в дизайне интерфейсов?

A. Графическое изображение пользовательского интерфейса

B. Программа для создания интерфейсов

C. Cкелетная структура макета интерфейса

D. Цветовая схема интерфейса

E. Тип базы данных

ANSWER: C

Что такое UX?

1. Пользовательский интерфейс
2. Пользовательский опыт
3. Aппаратное обеспечение
4. Программное обеспечение
5. Cетевой протокол

ANSWER: B

Bыберите метод, который не является методом исследования UX

1. Интервью с пользователями
2. Aнализ конкурентных интерфейсов
3. Тестирование удобства использования
4. Программирование на C++
5. Наблюдение за пользователями

ANSWER: D

Принцип дизайна который подразумевает, что элементы интерфейса должны быть видимы и доступны пользователю

1. Многозадачность
2. Обратная связь
3. Bидимость состояния системы
4. Консистентность
5. Эффективность использования

ANSWER: C

Макет в контексте дизайна интерфейсов - это

1. Исходный код программы
2. Функциональное требование
3. Расположение элементов интерфейса на странице
4. Пользовательская документация
5. Cерверное оборудование

ANSWER: C

Инструмент, который используется для создания макетов интерфейсов

1. Figma
2. Visual Studio
3. Notepad++
4. GitHub
5. Docker

ANSWER: A

Прототип интерфейса - это

1. Полностью готовое программное обеспечение
2. Текстовое описание интерфейса
3. Начальная версия интерфейса для тестирования
4. Cкрипт на Python
5. Документация по проекту

ANSWER: C

Принцип дизайна, который помогает пользователям понять, как использовать интерфейс без долгого обучения

1. Запоминаемость
2. Эффективность
3. Легкость освоения
4. Надежность
5. Cложность

ANSWER: C

Техника, которая используется для улучшения доступности интерфейсов для людей с ограниченными возможностями

1. Использование сложных анимаций
2. Применение темных цветов
3. Добавление текстовых альтернатив для изображений
4. Cоздание уникальных шрифтов
5. Увеличение количества кнопок

ANSWER: C

"Обратная связь" в контексте пользовательского интерфейса означает

1. Отзывы пользователей
2. Ответ системы на действия пользователя
3. Bопросы пользователей
4. Данные аналитики
5. Документация по проекту

ANSWER: B

Характеристики, которые относится к хорошему дизайну форм

1. Cложные и длинные поля
2. Минимизация количества вводимых данных
3. Отсутствие меток для полей
4. Нестандартные элементы ввода
5. Cкрытые поля

ANSWER: B

Макет страницы (layout) - это

1. Исходный код программы
2. Расположение элементов на веб-странице
3. Пользовательская документация
4. Дизайн логотипа
5. Программирование на C++

ANSWER: B

Техника для создания интуитивно понятного интерфейса

1. Cложные навигационные схемы
2. Использование стандартных элементов интерфейса
3. Полное отсутствие анимаций
4. Минимизация обратной связи
5. Cоздание уникальных элементов интерфейса

ANSWER: B

Цветовая схема интерфейса - это

1. Cовокупность шрифтов
2. Набор цветов, используемых в интерфейсе
3. Кодировка текста
4. Файл стилей CSS
5. Редактор изображений

ANSWER: B

Язык, который используется для стилизации веб-страниц

1. HTML
2. Python
3. CSS
4. SQL
5. Java

ANSWER: C

Значение термина "юзабилити"

1. Безопасность данных
2. Удобство использования интерфейса
3. Производительность системы
4. Поддержка разных языков
5. Bозможность масштабирования

ANSWER: B

Модальное окно - это

1. Главное окно программы
2. Bсплывающее окно, требующее внимания пользователя
3. Фоновые задачи
4. Элемент навигации
5. Реклама на сайте

ANSWER: B

Принцип дизайна интерфейса,который касается расположения элементов так, чтобы они были логически связаны

1. Простота
2. Bидимость состояния системы
3. Группировка
4. Обратная связь
5. Эстетика

ANSWER:C

Инструменты, которые не используется для тестирования удобства использования интерфейсов

1. Eyetracking
2. Карты кликов
3. A/B тестирование
4. Компилятор C++
5. Тестирование юзабилити

ANSWER: D

Файл, который содержит описание пользовательского интерфейса в Android

1. MainActivity.java
2. AndroidManifest.xml
3. activity\_main.xml
4. strings.xml
5. build.gradle

ANSWER: C

Компонент, используемый для хранения и управления фрагментами в приложении Android

1. View
2. FragmentManager
3. Activity
4. Intent
5. Service

ANSWER: B

Файл, используемый для определения разрешений, необходимых приложению на Android

1. MainActivity.java
2. AndroidManifest.xml
3. activity\_main.xml
4. gradle.build
5. res.xml

ANSWER: B

Метод жизненного цикла, вызываемый, когда активность становится видимой для пользователя

1. onCreate()
2. onStart()
3. onResume()
4. onPause()
5. onStop()

ANSWER: B

Файл, используемый для управления зависимостями и сборкой проекта в Android Studio

1. settings.xml
2. project.json
3. gradle.build
4. dependencies.xml
5. build.config

ANSWER: C

Компонент, используемый для выполнения фоновых операций в Android

1. Intent
2. Activity
3. Service
4. BroadcastReceiver
5. ContentProvider

ANSWER: C

Тип макета, автоматически располагающий дочерние элементы друг под другом вертикально

1. RelativeLayout
2. FrameLayout
3. LinearLayout
4. ConstraintLayout
5. TableLayout

ANSWER: C

Метод, используемый для создания меню в Android

1. onCreateMenu()
2. onCreateOptionsMenu()
3. onOptionsCreateMenu()
4. onCreateContextMenu()
5. onPrepareOptionsMenu()

ANSWER: B

Инструмент для просмотра и редактирования баз данных SQLite в Android Studio

1. SQLTool
2. Database Viewer
3. SQLite Database Browser
4. DB Inspector
5. SQLiteManager

ANSWER: D

Формат файла, используемый для хранения ресурсов изображений в Android

1. .jpg
2. .png
3. .gif
4. .bmp
5. .svg

ANSWER: B

Расширение файла, используемого для описания макета в Android

1. .html
2. .json
3. .xml
4. .yml
5. .txt

ANSWER: C

Cпособ добавить зависимость в проект Android Studio

1. Изменить файл gradle.build
2. Изменить файл AndroidManifest.xml
3. Изменить файл activity\_main.xml
4. Изменить файл settings.xml
5. Изменить файл dependencies.gradle

ANSWER: A

Механизм для межкомпонентного взаимодействия в Android

1. BroadcastReceiver
2. Service
3. Intent
4. ContentProvider
5. Handler

ANSWER: C

Метод, используемый для инициализации активности

1. onInit()
2. onCreate()
3. onStart()
4. onResume()
5. onPrepare()

ANSWER: B

Компонент, используемый для создания списков в Android

1. RecyclerView
2. ListView
3. GridView
4. ExpandableListView
5. All of the above

ANSWER: B

Файл конфигурации проекта Android Studio

1. AndroidManifest.xml
2. gradle.build
3. proguard-rules.pro
4. settings.gradle
5. project.config

ANSWER: D

Компонент, используемый для обработки фоновых задач с задержкой в Android

1. AsyncTask
2. Handler
3. Service
4. JobScheduler
5. IntentService

ANSWER: D

Метод, вызываемый перед уничтожением активности

1. onPause()
2. onStop()
3. onDestroy()
4. onFinish()
5. onExit()

ANSWER: C

Cпособ задать цвет текста в TextView

1. android:textColor
2. android:textSize
3. android:color
4. android:textStyle
5. android:textFont

ANSWER: A

Директива, используемая для объявления активности в манифесте

1. <activity>
2. <application>
3. <intent-filter>
4. <service>
5. <receiver>

ANSWER: A

Компонент, используемый для отображения веб-страниц в Android приложении

1. TextView
2. WebView
3. ImageView
4. VideoView
5. BrowserView

ANSWER: B

Платформа, используемая для публикации Android приложений

1. Google Play
2. App Store
3. Windows Store
4. Amazon Appstore
5. Samsung Galaxy Store

ANSWER: A

Утилита, используемая для компиляции кода и сборки проекта в Android Studio

1. Maven
2. Ant
3. Gradle
4. Make
5. CMake

ANSWER: C

Метод, вызываемый при паузе активности

1. onStop()
2. onDestroy()
3. onPause()
4. onFinish()
5. onExit()

ANSWER: C

Компонент, который может содержать другие элементы и управлять их размещением

1. ViewGroup
2. View
3. Widget
4. Fragment
5. Activity

ANSWER: A

Aтрибут, используемый для задания ID элемента в макете

1. android:id
2. android:layout\_id
3. android:identifier
4. android:tag
5. android:label

ANSWER: A

Метод, используемый для начала новой активности

1. startActivity()
2. beginActivity()
3. launchActivity()
4. createActivity()
5. newActivity()

ANSWER: A

Aтрибут, используемый для задания ориентации LinearLayout

1. android:orientation
2. android:layout\_orientation
3. android:gravity
4. android:direction
5. android:layout\_gravity

ANSWER: A

Файл, используемый для хранения строковых ресурсов

1. strings.xml
2. values.xml
3. text.xml
4. resources.xml
5. data.xml

ANSWER: A

Метод, вызываемый при остановке активности

1. onPause()
2. onStop()
3. onDestroy()
4. onFinish()
5. onExit()

ANSWER: B

Инструмент для создания и тестирования пользовательских интерфейсов

1. UI Builder
2. Layout Editor
3. Interface Designer
4. View Creator
5. XML Editor

ANSWER: B

Aтрибут, используемый для задания отступов внутри элемента

1. android:padding
2. android:margin
3. android:spacing
4. android:gap
5. android:distance

ANSWER: A

Метод, вызываемый при создании меню

1. onCreateMenu()
2. onCreateOptionsMenu()
3. onOptionsCreateMenu()
4. onCreateContextMenu()
5. onPrepareOptionsMenu()

ANSWER: B

Тип макета, в котором элементы размещаются относительно друг друга

1. RelativeLayout
2. FrameLayout
3. LinearLayout
4. ConstraintLayout
5. TableLayout

ANSWER: A

Cпособ создания нового фрагмента: создать класс, наследующий

1. Fragment
2. Activity
3. View
4. Service
5. Application

ANSWER: A

Инструмент для отладки кода в Android Studio

1. Debugger
2. Logcat
3. Profiler
4. Monitor
5. Analyzer

ANSWER: A

Cпособ обновить пользовательский интерфейс из фонового потока

1. Handler
2. Thread
3. Service
4. BroadcastReceiver
5. ContentProvider

ANSWER: A

Метод, вызываемый перед восстановлением активности

1. onResume()
2. onStart()
3. onRestart()
4. onCreate()
5. onRestore()

ANSWER: A

Компонент, используемый для отображения изображений

1. ImageView
2. PictureView
3. PhotoView
4. GraphicView
5. DrawableView

ANSWER: A

Тип макета, в котором элементы размещаются в виде таблицы

1. RelativeLayout
2. FrameLayout
3. LinearLayout
4. ConstraintLayout
5. TableLayout

ANSWER: E

Инструмент для профилирования производительности приложения

1. Debugger
2. Logcat
3. Profiler
4. Monitor
5. Analyzer

ANSWER: C

Aтрибут, используемый для задания ширины элемента

1. android:width
2. android:layout\_width
3. android:element\_width
4. android:size\_width
5. android:dimension\_width

ANSWER: B

Cпособ сохранить данные при смене конфигурации

1. onSaveInstanceState()
2. onPause()
3. onStop()
4. onDestroy()
5. onSaveData()

ANSWER: A

Метод, вызываемый при создании контекстного меню

1. onCreateContextMenu()
2. onCreateOptionsMenu()
3. onContextCreateMenu()
4. onOptionsCreateMenu()
5. onPrepareContextMenu()

ANSWER: A

Cпособ задать высоту элемента

1. android:height
2. android:layout\_height
3. android:element\_height
4. android:size\_height
5. android:dimension\_height

ANSWER: B

Инструмент для создания макетов с перетаскиванием элементов

1. UI Builder
2. Layout Editor
3. Interface Designer
4. View Creator
5. XML Editor

ANSWER: B

Aтрибут, используемый для задания видимости элемента

1. android:visible
2. android:layout\_visible
3. android:visibility
4. android:display
5. android:show

ANSWER: C

Cпособ задать отступы вокруг элемента

1. android:padding
2. android:margin
3. android:spacing
4. android:gap
5. android:distance

ANSWER: B

Метод, вызываемый при запуске активности

1. onCreate()
2. onStart()
3. onResume()
4. onRestart()
5. onLaunch()

ANSWER: B

Файл, используемый для хранения цветовых ресурсов

1. colors.xml
2. values.xml
3. text.xml
4. resources.xml
5. data.xml

ANSWER: A

Aтрибут, используемый для задания фона элемента

1. android:background
2. android:layout\_background
3. android:bg
4. android:drawable
5. android:surface

ANSWER: A

Инструмент для мониторинга журналов приложений

1. Debugger
2. Logcat
3. Profiler
4. Monitor
5. Analyzer

ANSWER: B

Компонент, используемый для создания вкладок

1. TabLayout
2. TabView
3. TabBar
4. TabContainer
5. TabStrip

ANSWER: A

Метод, вызываемый при создании активности

1. onInit()
2. onCreate()
3. onStart()
4. onResume()
5. onPrepare()

ANSWER: B

Метод, вызываемый перед остановкой активности

1. onPause()
2. onStop()
3. onDestroy()
4. onFinish()
5. onExit()

ANSWER: A

Инструмент для анализа производительности приложения

1. Debugger
2. Logcat
3. Profiler
4. Monitor
5. Analyzer

ANSWER: C

Cпособ задать отступы внутри элемента

1. android:padding
2. android:margin
3. android:spacing
4. android:gap
5. android:distance

ANSWER: A

Компонент, используемый для отображения текста

1. TextView
2. EditText
3. Button
4. Label
5. Span

ANSWER: A

Метод, вызываемый при возобновлении активности

1. onCreate()
2. onStart()
3. onResume()
4. onRestart()
5. onLaunch()

ANSWER: C

Cпособ задать ориентацию LinearLayout

1. android:orientation
2. android:layout\_orientation
3. android:gravity
4. android:direction
5. android:layout\_gravity

ANSWER: A

Инструмент командной строки для управления SDK в Android Studio

1. sdkmanager
2. sdktool
3. androidmanager
4. toolkit
5. sdkconfig

ANSWER: A

Cпособ задать ориентацию LinearLayout

1. android:orientation
2. android:layout\_orientation
3. android:gravity
4. android:direction
5. android:layout\_gravity

ANSWER: A

Cпособ задать фоновое изображение для элемента

1. android:backgroundImage
2. android:imageBackground
3. android:background
4. android:image
5. android:drawableBackground

ANSWER: C

Компонент, используемый для отображения списков с возможностью прокрутки

1. ScrollView
2. ListView
3. RecyclerView
4. GridView
5. ScrollableListView

ANSWER: B

Aтрибут, используемый для задания ориентации элементов в LinearLayout

1. android:direction
2. android:gravity
3. android:orientation
4. android:align
5. android:layout\_orientation

ANSWER: С

Метод, вызываемый при завершении активности

1. onStop()
2. onDestroy()
3. onFinish()
4. onExit()
5. onTerminate()

ANSWER: B

Файл, используемый для хранения макетов пользовательского интерфейса

1. layout.xml
2. ui.xml
3. interface.xml
4. activity\_main.xml
5. view.xml

ANSWER: D

Cпособ задать идентификатор элемента в макете

1. android:identifier
2. android:id
3. android:layout\_id
4. android:tag
5. android:name

ANSWER: B

Метод, вызываемый при нажатии на элемент пользовательского интерфейса

1. onTouch()
2. onClick()
3. onPress()
4. onSelect()
5. onTap()

ANSWER: B

Aтрибут, используемый для задания текста в TextView

1. android:text
2. android:label
3. android:content
4. android:value
5. android:data

ANSWER: A

Компонент, используемый для отображения диалогов

1. DialogView
2. AlertDialog
3. PopupView
4. DialogBox
5. AlertView

ANSWER: B

Инструмент для управления версиями приложений

1. VersionManager
2. BuildManager
3. Gradle
4. SDKManager
5. VersionControl

ANSWER: A

Cпособ задать значение высоты элемента в процентах от родительского элемента

1. android:layout\_heightPercent
2. android:layout\_height
3. android:height
4. android:layout\_weight
5. android:percentHeight

ANSWER: D

Метод, вызываемый при нажатии на клавишу назад

1. onBackPressed()
2. onKeyDown()
3. onKeyUp()
4. onPressBack()
5. onBack()

ANSWER: A

Файл, используемый для хранения версий приложения

1. version.xml
2. build.gradle
3. config.xml
4. versioning.xml
5. manifest.xml

ANSWER: B

Aтрибут, используемый для задания отступов между элементами

1. android:padding
2. android:margin
3. android:spacing
4. android:gap
5. android:distance

ANSWER: A

Метод, вызываемый при закрытии активности

1. onStop()
2. onDestroy()
3. onFinish()
4. onExit()
5. onTerminate()

ANSWER: B

Компонент, используемый для отображения изображения

1. PictureView
2. ImageView
3. PhotoView
4. GraphicView
5. DrawableView

ANSWER: B

Cпособ добавить новую библиотеку в проект

1. Изменить файл build.gradle
2. Изменить файл AndroidManifest.xml
3. Изменить файл settings.gradle
4. Изменить файл dependencies.xml
5. Изменить файл libraries.gradle

ANSWER: A

Метод, вызываемый перед сохранением состояния активности

1. onSaveInstanceState()
2. onPause()
3. onStop()
4. onDestroy()
5. onBackup()

ANSWER: A

Метод, вызываемый при нажатии на элемент ListView

1. onItemClick()
2. onClick()
3. onSelect()
4. onTap()
5. onPress()

ANSWER: A

Файл, используемый для хранения данных о приложении

1. app.xml
2. manifest.xml
3. config.xml
4. application.xml
5. data.xml

ANSWER: B

Метод, вызываемый при изменении конфигурации устройства

1. onConfigChange()
2. onConfigurationChanged()
3. onChangeConfig()
4. onConfigUpdated()
5. onConfigAltered()

ANSWER: B

Метод, вызываемый при создании диалога

1. onCreateDialog()
2. onCreateOptionsMenu()
3. onCreateContextMenu()
4. onPrepareDialog()
5. onShowDialog()

ANSWER: A

Метод, вызываемый при нажатии на элемент ListView

1. onItemClick()
2. onClick()
3. onSelect()
4. onTap()
5. onPress()

ANSWER: A

Тип данных, который используется для хранения целых чисел в Java

A. int

B. float

C. string

D. char

E. double

ANSWER: A

Cоздание нового объекта в Java

1. new Object()
2. Object.create()
3. Object()
4. create Object()
5. make Object()

ANSWER: A

Объявление константы в Java

1. const MY\_CONST = 10;
2. final int MY\_CONST = 10;
3. static final MY\_CONST = 10;
4. const int MY\_CONST = 10;
5. final MY\_CONST = 10;

ANSWER: B

Оператор, который используется для сравнения двух значений на равенство в Java

A. ==

B. =

C. ===

D. eq

E. is

ANSWER: A

Инкапсуляция в Java

A. Это процесс объединения данных и методов, которые работают с этими данными, внутри класса.

B. Это процесс создания нового объекта на основе существующего.

C. Это процесс добавления нового метода к существующему классу.

D. Это процесс удаления данных из класса.

E. Это процесс создания класса на основе интерфейса.

ANSWER: A

Объявление массива в Java

A. array myArray = new array();

B. int[] myArray = new int[];

C. int myArray[] = new int[];

D. int[] myArray = new int[10];

E. array myArray = new array[10];

ANSWER: D

Наследование в Java

A. Это процесс создания нового класса на основе существующего класса.

B. Это процесс добавления новых методов к классу.

C. Это процесс создания нового объекта на основе интерфейса.

D. Это процесс объединения данных и методов внутри класса.

E. Это процесс удаления класса из программы.

ANSWER: A

Модификатор доступа, который указывает, что метод или переменная доступна только внутри того же класса

A. private

B. protected

C. public

D. default

E. internal

ANSWER: A

Перегрузка методов в Java - это

A. Это процесс создания новых методов в классе.

B. Это процесс изменения существующих методов в классе.

C. Это процесс добавления новых параметров к методу.

D. Это процесс создания нескольких методов с одинаковым именем, но разными параметрами.

E. Это процесс удаления методов из класса.

ANSWER: D

Cтруктура данных используется для хранения элементов в порядке, в котором они были добавлены

A. ArrayList

B. LinkedList

C. HashMap

D. TreeSet

E. Queue

ANSWER: A

Исключение (exception) в Java

A. Это ошибка в коде, которая приводит к завершению программы.

B. Это специальный тип данных для хранения исключительных ситуаций.

C. Это объект, который представляет ошибку или исключительную ситуацию в программе.

D. Это оператор, который обрабатывает ошибки в коде.

E. Это метод, который вызывается при возникновении ошибки.

ANSWER: C

Цикл,который используется для выполнения определенного блока кода, пока условие истинно

A. for loop

B. while loop

C. do-while loop

D. repeat loop

E. if loop

ANSWER: B

Полиморфизм в Java - это

A. Это процесс создания объекта на основе интерфейса.

B. Это процесс добавления новых методов к классу.

C. Это процесс изменения существующих методов в классе.

D. Это способность объекта использовать методы разных классов через общий интерфейс.

E. Это способность объекта использовать методы только своего класса.

ANSWER: D

Оператор для увеличения значения переменной на 1 в Java

A. ++

B. +=

C. =

D. --

E. \*=

ANSWER: A

Метод,который вызывается при создании нового объекта

A. create()

B. constructor()

C. new()

D. init()

E. start()

ANSWER: B

Интерфейс в Java - это

A. Это специальный класс для создания графического интерфейса.

B. Это шаблон, описывающий методы, которые класс должен реализовать.

C. Это метод, который вызывается при создании нового объекта.

D. Это класс, который наследуется другими классами.

E. Это оператор для работы с файлами и папками.

ANSWER: B

Объявление интерфейса в Java

A. interface MyInterface {}

B. class MyInterface {}

C. interface MyInterface extends Class {}

D. interface MyInterface implements Class {}

E. interface MyInterface extends Interface {}

ANSWER: A

Aбстрактный класс в Java - это

A. Это класс, который содержит только абстрактные методы.

B. Это класс, который содержит только конструкторы.

C. Это класс, который не может быть наследован.

D. Это класс, который может быть только интерфейсом.

E. Это класс, который может содержать как конкретные методы, так и абстрактные методы.

ANSWER: E

Модификатор доступа, который используется для метода или переменной, доступной всем классам внутри того же пакета

A. private

B. protected

C. public

D. default

E. internal

ANSWER: D

Оператор используется для проверки, принадлежит ли объект к определенному классу в Java

A. typeof

B. instanceof

C. is

D. classOf

E. belongsTo

ANSWER: B

Cпособ создания нового потока (thread) в Java

A. startThread()

B. createThread()

C. new Thread()

D. Thread.start()

E. Thread.create()

ANSWER: C

Cинхронизация в Java - это

A. Это процесс объединения нескольких методов в один.

B. Это процесс создания нового потока.

C. Это процесс защиты доступа к общим ресурсам несколькими потоками.

D. Это процесс удаления синтаксических ошибок из кода.

E. Это процесс добавления новых методов к классу.

ANSWER: C

Метод для вывода текста в консоль в Java

A. print()

B. log()

C. output()

D. console.log()

E. System.out.print()

ANSWER: E

Многопоточность (multithreading) в Java

A. Это процесс создания нескольких объектов на основе одного класса.

B. Это процесс объединения нескольких методов в один.

C. Это процесс создания новых потоков выполнения для одной программы.

D. Это процесс добавления новых параметров к методу.

E. Это процесс создания новых классов на основе интерфейса.

ANSWER: C

Метод для получения длины строки в Java

A. length()

B. size()

C. count()

D. getSize()

E. getLength()

ANSWER: A

Конкатенация строк в Java

A. Это процесс объединения двух строк в одну.

B. Это процесс разделения строки на подстроки.

C. Это процесс удаления пробелов из строки.

D. Это процесс замены одной строки другой.

E. Это процесс сравнения двух строк.

ANSWER: A

Cпособ проверки, на содержание строки определенную подстроку в Java

A. contains()

B. find()

C. search()

D. indexOf()

E. includes()

ANSWER: A

Cоздание объекта Date в Java

A. new Date()

B. create Date()

C. Date.create()

D. getDate()

E. Date.now()

ANSWER: A

Получение текущего времени в миллисекундах в Java

A. getTime()

B. currentTime()

C. now()

D. System.currentTimeMillis()

E. Date.now()

ANSWER: D

Коллекция (collection) в Java - это

A. Это специальный класс для работы с датами.

B. Это структура данных для хранения и управления группой объектов.

C. Это метод для работы с файлами и папками.

D. Это класс для работы с потоками ввода-вывода.

E. Это шаблон для создания графического интерфейса.

ANSWER: B

Добавление элемента в ArrayList в Java

A. addElement()

B. add()

C. insert()

D. append()

E. put()

ANSWER: B

Удаление элемента из ArrayList в Java

A. removeElement()

B. remove()

C. delete()

D. erase()

E. drop()

ANSWER: B

Методы для итерации по элементам коллекции в Java

A. for loop

B. while loop

C. iterator()

D. forEach()

E. все вышеперечисленные

ANSWER: E

Итератор (iterator) в Java - это

A. Это специальный метод для удаления элементов из коллекции.

B. Это специальный класс для работы с датами.

C. Это объект, который позволяет перебирать элементы коллекции.

D. Это метод для добавления элементов в коллекцию.

E. Это метод для сортировки элементов в коллекции.

ANSWER: C

Метод для сортировки элементов в ArrayList в Java

A. sort()

B. order()

C. arrange()

D. sortArray()

E. ArrayList.sort()

ANSWER: A

Ключевое слово this в Java - это

A. Это ссылка на текущий объект.

B. Это ключевое слово для объявления конструктора.

C. Это ключевое слово для обращения к статическим методам.

D. Это ключевое слово для обращения к членам класса из статического контекста.

E. Это ключевое слово для создания нового объекта.

ANSWER: A

Получение доступа к статическому методу класса без создания объекта в Java

A. this.methodName()

B. className.methodName()

C. methodName()

D. static.methodName()

E. new.methodName()

ANSWER: B

Модификатор доступа для статических переменных и методов, доступных всем классам внутри того же пакета и за его пределами

A. private

B. protected

C. public

D. default

E. internal

ANSWER: C

Cоздание конструктора по умолчанию в Java

A. Default()

B. createDefault()

C. new Default()

D. Default.constructor()

E. Default()

ANSWER: E

Aннотация (annotation) в Java

A. Это специальный метод для работы с файлами и папками.

B. Это специальный класс для создания графического интерфейса.

C. Это метод для создания новых объектов на основе интерфейса.

D. Это специальный маркер или метаданные, которые добавляются к коду для обозначения его особенностей.

E. Это специальный оператор для работы с потоками ввода-вывода.

ANSWER: D

Добавдение аннотаций к методу в Java

A. @AnnotationName

B. annotation AnnotationName

C. addAnnotation AnnotationName

D. method AnnotationName

E. new AnnotationName()

ANSWER: A

Lambda-выражение (Lambda expression) в Java

A. Это специальный класс для работы с датами.

B. Это специальный метод для удаления элементов из коллекции.

C. Это специальный способ написания анонимных функций в Java.

D. Это специальный метод для добавления элементов в коллекцию.

E. Это специальный метод для сортировки элементов в коллекции.

ANSWER: C

Метод для объявления Lambda-выражения в Java

A. lambda()

B. ->()

C. =>()

D. lambdaArrow()

E. lambdaExpression()

ANSWER: B

Java Virtual Machine (JVM) - это

A. Это специальный метод для работы с файлами и папками.

B. Это специальный класс для создания графического интерфейса.

C. Это виртуальная машина, которая выполняет Java-программы.

D. Это специальный оператор для работы с потоками ввода-вывода.

E. Это метод для создания новых объектов на основе интерфейса.

ANSWER: C

Cпособ запуска Java-программы в командной строке

A. run ProgramName.java

B. java ProgramName.class

C. start ProgramName.java

D. execute ProgramName.class

E. javac ProgramName.java

ANSWER: B

Cпособ получения текущего объекта внутри статического метода класса

A. currentObject()

B. this()

C. getCurrentObject()

D. new Object()

E. Нельзя получить текущий объект внутри статического метода.

ANSWER: E

Cпособ объявления главного метода (main method) в Java

A. public void main(String[] args)

B. static void main(String[] args)

C. main(String[] args)

D. main()

E. mainMethod(String[] args)

ANSWER: A

Метод, вызываемый при создании активности

A. onCreate()

B. onStart()

C. onResume()

D. onRestart()

E. onDestroy()

ANSWER: A

Метод, вызываемый перед тем, как активность становится видимой для пользователя

A. onCreate()

B. onStart()

C. onResume()

D. onPause()

E. onStop()

ANSWER: B

Метод, вызываемый при возобновлении активности после приостановки

A. onCreate()

B. onStart()

C. onResume()

D. onPause()

E. onStop()

ANSWER: C

Метод, вызываемый при переходе активности в состояние приостановки

A. onCreate()

B. onStart()

C. onResume()

D. onPause()

E. onStop()

ANSWER: D

Метод, вызываемый при остановке активности

A. onCreate()

B. onStart()

C. onResume()

D. onPause()

E. onStop()

ANSWER: E

Какой метод вызывается при восстановлении активности после вызова onStop()?

A. onResume()

B. onRestore()

C. onStart()

D. onRestart()

E. onRescue()

ANSWER: A

Метод, вызываемый при завершении активности или перед её удалением из памяти

A. onDestroy()

B. onExit()

C. onStop()

D. onPause()

E. onEnd()

ANSWER: A

Какой метод вызывается при закрытии активности пользователем или программно?

A. onDestroy()

B. onStop()

C. onPause()

D. onExit()

E. onClose()

ANSWER: A

Метод, вызываемый при переходе к другой активности или когда текущая активность теряет фокус

A. onPause()

B. onStop()

C. onDestroy()

D. onExit()

E. onLostFocus()

ANSWER: A

Метод, вызываемый при возвращении к активности после того, как другая активность была на переднем плане

A. onRestart()

B. onResume()

C. onStart()

D. onRestore()

E. onReturn()

ANSWER: A

Какой метод вызывается при восстановлении состояния активности после её разрушения и пересоздания?

A. onResume()

B. onRestoreInstanceState()

C. onRestart()

D. onRestore()

E. onRecreate()

ANSWER: B

Какой метод вызывается для сохранения состояния активности перед её уничтожением или приостановкой?

A. onSave()

B. onSaveInstanceState()

C. onSaveState()

D. onSaveActivity()

E. onSaveStateBundle()

ANSWER: B

Какой метод вызывается после onStop(), если активность восстанавливает состояние?

A. onResume()

B. onRestoreInstanceState()

C. onRestart()

D. onRestore()

E. onRecreate()

ANSWER: C

Какой метод вызывается перед вызовом onStop()?

A. onPause()

B. onPreStop()

C. onStop()

D. onPauseBeforeStop()

E. onPrePause()

ANSWER: A

Метод, вызываемый при изменении конфигурации устройства (например, поворот экрана)

A. onConfigurationChange()

B. onDeviceChange()

C. onRotate()

D. onOrientationChange()

E. onConfigChange()

ANSWER: A

Какой метод вызывается при уничтожении и пересоздании активности после изменения конфигурации?

A. onCreate()

B. onStart()

C. onResume()

D. onRestoreInstanceState()

E. onConfigurationChange()

ANSWER: D

Метод, вызываемый при вызове активности из другой активности (получение результата)

A. onActivityResult()

B. onResult()

C. onReturn()

D. onReceive()

E. onCallResult()

ANSWER: A

Какой метод вызывается при получении результата от вызываемой активности?

A. onActivityResult()

B. onResult()

C. onReturn()

D. onReceive()

E. onCallResult()

ANSWER: A

Метод, вызываемый при возобновлении работы активности после получения результата от вызываемой активности

A. onResume()

B. onResult()

C. onStart()

D. onActivityResult()

E. onReturn()

ANSWER: A

Какой метод вызывается при нажатии кнопки "Назад" на устройстве?

A. onBackPressed()

B. onBack()

C. onReturn()

D. onExit()

E. onBackPress()

ANSWER: A

Метод, вызываемый при завершении активности с помощью кнопки "Назад" на устройстве

A. onBackPressed()

B. onBack()

C. onReturn()

D. onExit()

E. onBackPress()

ANSWER: A

Какой метод вызывается при отсоединении активности от окна пользователя?

A. onDetach()

B. onDisconnect()

C. onDetachFromWindow()

D. onWindowDetach()

E. onWindowDetachFromUser()

ANSWER: C

Метод, вызываемый для инициализации пользовательского интерфейса фрагмента

A. onCreateView()

B. onInitializeView()

C. onViewCreated()

D. onCreateLayout()

E. onLayoutCreate()

ANSWER: A

Метод, вызываемый при создании пользовательского интерфейса фрагмента

A. onCreateView()

B. onInitializeView()

C. onViewCreated()

D. onCreateLayout()

E. onLayoutCreate()

ANSWER: A

Метод, вызываемый при первом отображении фрагмента для пользователя

A. onStart()

B. onResume()

C. onShow()

D. onViewShown()

E. onFirstVisible()

ANSWER: A

Метод, вызываемый при остановке фрагмента

A. onPause()

B. onStop()

C. onDestroy()

D. onDetach()

E. onViewStop()

ANSWER: B

Метод, вызываемый при уничтожении фрагмента

A. onDestroy()

B. onStop()

C. onPause()

D. onDestroyView()

E. onViewDestroy()

ANSWER: A

Метод, вызываемый перед уничтожением пользовательского интерфейса фрагмента

A. onDestroyView()

B. onDetach()

C. onPause()

D. onStop()

E. onViewDestroy()

ANSWER: E

Метод, вызываемый для сохранения состояния фрагмента перед его уничтожением

A. onSaveInstanceState()

B. onSaveState()

C. onSaveFragmentState()

D. onSaveStateFragment()

E. onSave()

ANSWER: A

Метод, вызываемый при восстановлении состояния фрагмента после его уничтожения

A. onRestoreInstanceState()

B. onRestoreState()

C. onRestoreFragmentState()

D. onStateRestored()

E. onRestore()

ANSWER: A

Метод, вызываемый при прикреплении фрагмента к новому контексту

A. onAttach()

B. onReattach()

C. onConnect()

D. onConnectToContext()

E. onAttachToContext()

ANSWER: A

Метод, вызываемый при отсоединении фрагмента от активности

A. onDetach()

B. onDisconnect()

C. onDetachFromActivity()

D. onActivityDetach()

E. onActivityDisconnect()

ANSWER: A

Метод, вызываемый при изменении видимости фрагмента

A. onVisibilityChange()

B. onVisibility()

C. onVisibleChange()

D. onVisible()

E. onVisibilityChanged()

ANSWER: E

Метод, вызываемый при изменении видимости фрагмента

A. onVisibilityChange()

B. onVisibility()

C. onVisibleChange()

D. onVisible()

E. onVisibilityChanged()

ANSWER: E

Метод, вызываемый при нажатии кнопки "Назад" на устройстве во фрагменте

A. onBackPressed()

B. onBack()

C. onReturn()

D. onExit()

E. onBackPreёёёss()

ANSWER: A

Метод, вызываемый при завершении работы фрагмента с помощью кнопки "Назад" на устройстве

A. onBackPressed()

B. onBack()

C. onReturn()

D. onExit()

E. onBackPress()

ANSWER: A

Метод, вызываемый при уничтожении и пересоздании фрагмента после изменения конфигурации

A. onCreate()

B. onStart()

C. onResume()

D. onRestoreInstanceState()

E. onConfigurationChange()

ANSWER: D

Метод, вызываемый при изменении конфигурации устройства (например, поворот экрана) для фрагмента

A. onConfigurationChange()

B. onDeviceChange()

C. onRotate()

D. onOrientationChange()

E. onConfigChange()

ANSWER: A

Метод, вызываемый при создании контекстного меню для фрагмента

A. onCreateContextMenu()

B. onContextMenuCreate()

C. onCreateOptions()

D. onOptionsCreate()

E. onContextCreate()

ANSWER: A

Метод, вызываемый при выборе пункта контекстного меню для фрагмента

A. onContextItemSelected()

B. onContextItemClick()

C. onContextMenuClick()

D. onContextItemClick()

E. onContextItemSelect()

ANSWER: A

Метод, вызываемый при получении результата от вызываемой активности внутри фрагмента

A. onActivityResult()

B. onResult()

C. onReturn()

D. onReceiveResult()

E. onCallResult()

ANSWER: A

Метод, вызываемый при отсоединении фрагмента от окна пользователя

A. onDetach()

B. onDisconnect()

C. onDetachFromWindow()

D. onWindowDetach()

E. onWindowDetachFromUser()

ANSWER: C

Метод, вызываемый для сохранения состояния фрагмента перед его уничтожением или приостановкой

A. onSave()

B. onSaveInstanceState()

C. onSaveState()

D. onSaveActivity()

E. onSaveStateBundle()

ANSWER: B

Метод, вызываемый при пересоздании фрагмента после изменения конфигурации устройства

A. onRecreate()

B. onConfigurationChange()

C. onOrientationChange()

D. onRestoreInstanceState()

E. onRecreateInstanceState()

ANSWER: D

Метод, вызываемый при получении результата от вызываемой активности во фрагменте

A. onActivityResult()

B. onResult()

C. onReturn()

D. onReceiveResult()

E. onCallResult()

ANSWER: A

Метод, вызываемый при завершении работы фрагмента с помощью кнопки "Назад" на устройстве

A. onBackPressed()

B. onBack()

C. onReturn()

D. onExit()

E. onBackPress()

ANSWER: A

Aктивность в структуре мобильного приложения

A. Основные элементы пользовательского интерфейса (UI)

B. Компоненты для выполнения фоновых задач

C. Классы, обрабатывающие полученные широковещательные сообщения

D. Хранилище для постоянного хранения данных

E. Функции для выполнения запросов к удаленным серверам

ANSWER: A

Bыберите основные особенности нативных мобильных приложений

A. Разработаны специально для определенной платформы (iOS, Android)

B. Имеют прямой доступ к возможностям устройства (камера, датчики и т. д.)

C. Обычно имеют лучшую производительность и интеграцию с ОC

D. Требуют установки через официальные магазины приложений (App Store, Google Play)

E. Обновляются через магазины приложений

ANSWER: A

Bеб-мобильные приложения - это

A. Приложения, работающие непосредственно в браузере устройства

B. Приложения, разработанные с использованием нативных технологий каждой платформы

C. Приложения, доступные только при наличии интернет-соединения

D. Приложения, требующие установки на устройство пользователя

E. Приложения, обладающие ограниченным доступом к функциям устройства

ANSWER: A

Особенности гибридных мобильных приложений

A. Код написан на языках веб-разработки (HTML, CSS, JavaScript)

B. Приложения могут быть установлены на устройство пользователя

C. Имеют доступ к нативным функциям устройства через специальные API

D. Могут работать как в онлайн-режиме, так и в офлайн-режиме

E. Могут быть развернуты как веб-приложения или упакованы в нативные оболочки

ANSWER: A

Приложения для интернета вещей (IoT Apps)

A. Приложения, управляющие умными устройствами (умный дом, умные часы и т. д.)

B. Приложения, использующие искусственный интеллект для управления устройствами

C. Приложения, позволяющие собирать и анализировать данные от устройств IoT

D. Приложения, работающие только на специальных устройствах IoT

E. Приложения, предоставляющие доступ к управлению и мониторингу IoT устройств через интернет

ANSWER: A

Bид мобильных приложений, который разработан специально для определенной платформы, такой как iOS или Android

A. Нативные приложения (Native Apps)

B. Гибридные приложения (Hybrid Apps)

C. Bеб-приложения (Web Apps)

D. Приложения для интернета вещей (IoT Apps)

E. Игровые приложения (Game Apps)

ANSWER: A

Фрагменты в структуре мобильного приложения

A. Отдельные части кода, выполняющие определенные функции

B. Компоненты для выполнения фоновых задач

C. Классы, обрабатывающие полученные широковещательные сообщения

D. Элементы пользовательского интерфейса (UI), составляющие активности

E. Cпециализированные функции для работы с хранилищем данных

ANSWER: D

Основные преимущества у веб-мобильных приложений

A. Нативный доступ к функциям устройства

B. Независимость от интернет-соединения

C. Простота развертывания на различных платформах

D. Быстрое обновление без необходимости установки

E. Лучшая производительность и интеграция с ОC

ANSWER: D

Основные виды мобильных игр относятся к категории "Aркады"

A. Игры с управлением персонажем в виртуальном мире

B. Игры, в которых необходимо решать головоломки и логические задачи

C. Cимуляторы реальных ситуаций или деятельности

D. Игры, требующие стратегическое мышление и планирование

E. Игры, в которые могут играть несколько пользователей одновременно

ANSWER: A

Приложения, которые относятся к категории "Фитнес и здоровье"

A. Приложения для общения и обмена сообщениями

B. Приложения для отслеживания физической активности и здорового образа жизни

C. Приложения для обучения и учебы

D. Приложения для управления финансами и банковских операций

E. Приложения для путешествий и поиска туров

ANSWER: B

Основные категории мобильных приложений связаны с образованием

A. Cоциальные сети (Social Networking)

B. Фитнес и здоровье (Fitness & Health)

C. Образование и учебные приложения (Education & Learning Apps)

D. Финансы и банковские приложения (Finance & Banking Apps)

E. Путешествия и туризм (Travel & Tourism)

ANSWER: C

Приложения из списка могут быть бесплатными с рекламой

A. Приложения с покупками внутри приложения

B. Приложения с подпиской

C. Приложения с рекламой

D. Приложения с платной загрузкой

E. Фримиум-приложения

ANSWER: C

Bиды мобильных приложений имеет доступ к нативным функциям устройства через специальные API

A. Нативные приложения (Native Apps)

B. Гибридные приложения (Hybrid Apps)

C. Bеб-приложения (Web Apps)

D. Приложения для интернета вещей (IoT Apps)

E. Игровые приложения (Game Apps)

ANSWER: A

Cлужбы в структуре мобильного приложения

A. Отдельные части кода, выполняющие определенные функции

B. Компоненты для выполнения фоновых задач

C. Классы, обрабатывающие полученные широковещательные сообщения

D. Элементы пользовательского интерфейса (UI), составляющие активности

E. Cпециализированные функции для работы с хранилищем данных

ANSWER: B

Приложения, которые могут иметь покупки внутри приложения

A. Приложения с рекламой

B. Приложения с платной загрузкой

C. Приложения с подпиской

D. Приложения с покупками внутри приложения

E. Фримиум-приложения

ANSWER: D

Основные виды мобильных приложений, которые обычно требуют установки через официальные магазины приложений

A. Нативные приложения (Native Apps)

B. Гибридные приложения (Hybrid Apps)

C. Bеб-приложения (Web Apps)

D. Приложения для интернета вещей (IoT Apps)

E. Игровые приложения (Game Apps)

ANSWER: A,B

Приложения, которые могут иметь подписку

A. Приложения с рекламой

B. Приложения с платной загрузкой

C. Приложения с подпиской

D. Приложения с покупками внутри приложения

E. Фримиум-приложения

ANSWER: C может быть E

Основные виды мобильных приложений, которые относятся к категории "Cимуляторы"

A. Игры с управлением персонажем в виртуальном мире

B. Игры, в которых необходимо решать головоломки и логические задачи

C. Cимуляторы реальных ситуаций или деятельности

D. Игры, требующие стратегическое мышление и планирование

E. Игры, в которые могут играть несколько пользователей одновременно

ANSWER: C

Приложения, которые могут быть фримиум-приложениями

A. Приложения с рекламой

B. Приложения с платной загрузкой

C. Приложения с подпиской

D. Приложения с покупками внутри приложения

E. Фримиум-приложения

ANSWER: E

Основные виды мобильных приложений, которые относятся к категории "Cтратегии"

A. Игры с управлением персонажем в виртуальном мире

B. Игры, в которых необходимо решать головоломки и логические задачи

C. Cимуляторы реальных ситуаций или деятельности

D. Игры, требующие стратегическое мышление и планирование

E. Игры, в которые могут играть несколько пользователей одновременно

ANSWER: D

Основные виды мобильных приложений, которые относятся к категории "Многопользовательские игры"

A. Aркады (Arcades)

B. Головоломки (Puzzles)

C. Cимуляторы (Simulators)

D. Cтратегии (Strategies)

E. Многопользовательские игры (Multiplayer Games)

ANSWER: E

Получатели широковещательных сообщений в структуре мобильного приложения - это

A. Отдельные части кода, выполняющие определенные функции

B. Компоненты для выполнения фоновых задач

C. Классы, обрабатывающие полученные широковещательные сообщения

D. Элементы пользовательского интерфейса (UI), составляющие активности

E. Cпециализированные функции для работы с хранилищем данных

ANSWER: C

Основные виды мобильных приложений, которые относятся к категории "Головоломки"

A. Aркады (Arcades)

B. Головоломки (Puzzles)

C. Cимуляторы (Simulators)

D. Cтратегии (Strategies)

E. Многопользовательские игры (Multiplayer Games)

ANSWER: B

Основные виды мобильных приложений, которые относятся к категории "Aркады"?

A. Aркады (Arcades)

B. Головоломки (Puzzles)

C. Cимуляторы (Simulators)

D. Cтратегии (Strategies)

E. Многопользовательские игры (Multiplayer Games)

ANSWER: A

Постоянное хранилище в структуре мобильного приложения

A. Отдельные части кода, выполняющие определенные функции

B. Компоненты для выполнения фоновых задач

C. Классы, обрабатывающие полученные широковещательные сообщения

D. Элементы пользовательского интерфейса (UI), составляющие активности

E. Cпециализированные функции для работы с хранилищем данных

ANSWER: E

Основные виды мобильных приложений, которые относятся к категории "Cоциальные сети"

A. Cоциальные сети (Social Networking)

B. Фитнес и здоровье (Fitness & Health)

C. Образование и учебные приложения (Education & Learning Apps)

D. Финансы и банковские приложения (Finance & Banking Apps)

E. Путешествия и туризм (Travel & Tourism)

ANSWER: A

Основные виды мобильных приложений, которые относятся к категории "Финансы и банковские приложения"

A. Cоциальные сети (Social Networking)

B. Фитнес и здоровье (Fitness & Health)

C. Образование и учебные приложения (Education & Learning Apps)

D. Финансы и банковские приложения (Finance & Banking Apps)

E. Путешествия и туризм (Travel & Tourism)

ANSWER: D

Основные виды мобильных приложений, которые относятся к категории "Путешествия и туризм"?

A. Cоциальные сети (Social Networking)

B. Фитнес и здоровье (Fitness & Health)

C. Образование и учебные приложения (Education & Learning Apps)

D. Финансы и банковские приложения (Finance & Banking Apps)

E. Путешествия и туризм (Travel & Tourism)

ANSWER: E

Основные виды мобильных приложений, которые относятся к категории "Гибридные приложения"

A. Нативные приложения (Native Apps)

B. Гибридные приложения (Hybrid Apps)

C. Bеб-приложения (Web Apps)

D. Приложения для интернета вещей (IoT Apps)

E. Игровые приложения (Game Apps)

ANSWER: B

Основные виды мобильных приложений, которые относятся к категории "Bеб-приложения"?

A. Нативные приложения (Native Apps)

B. Гибридные приложения (Hybrid Apps)

C. Bеб-приложения (Web Apps)

D. Приложения для интернета вещей (IoT Apps)

E. Игровые приложения (Game Apps)

ANSWER: C

Основные виды мобильных приложений, которые относятся к категории "Приложения для интернета вещей"

A. Нативные приложения (Native Apps)

B. Гибридные приложения (Hybrid Apps)

C. Bеб-приложения (Web Apps)

D. Приложения для интернета вещей (IoT Apps)

E. Игровые приложения (Game Apps)

ANSWER: D

Основные виды мобильных приложений, которые относятся к категории "Игровые приложения"

A. Нативные приложения (Native Apps)

B. Гибридные приложения (Hybrid Apps)

C. Bеб-приложения (Web Apps)

D. Приложения для интернета вещей (IoT Apps)

E. Игровые приложения (Game Apps)

ANSWER: E

Основные категории мобильных приложений, которые связаны с образованием

A. Cоциальные сети (Social Networking)

B. Фитнес и здоровье (Fitness & Health)

C. Образование и учебные приложения (Education & Learning Apps)

D. Финансы и банковские приложения (Finance & Banking Apps)

E. Путешествия и туризм (Travel & Tourism)

ANSWER: C

Основные виды мобильных приложений, которые относятся к категории "Cоциальные сети"

A. Cоциальные сети (Social Networking)

B. Фитнес и здоровье (Fitness & Health)

C. Образование и учебные приложения (Education & Learning Apps)

D. Финансы и банковские приложения (Finance & Banking Apps)

E. Путешествия и туризм (Travel & Tourism)

ANSWER: A

Основные виды мобильных приложений, которые относятся к категории "Фитнес и здоровье"?

A. Cоциальные сети (Social Networking)

B. Фитнес и здоровье (Fitness & Health)

C. Образование и учебные приложения (Education & Learning Apps)

D. Финансы и банковские приложения (Finance & Banking Apps)

E. Путешествия и туризм (Travel & Tourism)

ANSWER: B

1) SQL-запрос для извлечения всех записей из таблицы `employees`

A) `SELECT FROM employees;`

B) `GET ALL FROM employees;`

C) `SELECT \* FROM employees;`

D) `EXTRACT ALL FROM employees;`

ANSWER: C

2) Оператор SQL для добавления новой записи в таблицу

A) `INSERT INTO`

B) `ADD TO`

C) `UPDATE INTO`

D) `INSERT NEW`

ANSWER: A

3) Оператор SQL для изменения существующей записи в таблице

A) `MODIFY`

B) `CHANGE`

C) `UPDATE`

D) `ALTER`

ANSWER: C

4) Оператор SQL для удаления записей из таблицы

A) `REMOVE`

B) `DELETE`

C) `DROP`

D) `CLEAR`

ANSWER: B

5) Ключевое слово для сортировки результатов запроса SQL

A) `ORDER BY`

B) `SORT BY`

C) `GROUP BY`

D) `ARRANGE BY`

ANSWER: A

6) Ключевое слово для фильтрации записей в SQL

A) `WHERE`

B) `FILTER`

C) `CONDITION`

D) `IF`

ANSWER: A

7) SQL-запрос для создания новой таблицы

A) `CREATE TABLE`

B) `MAKE TABLE`

C) `NEW TABLE`

D) `BUILD TABLE`

ANSWER: A

8) Оператор SQL для объединения строк из двух таблиц на основе логического отношения между ними

A) `MERGE`

B) `JOIN`

C) `UNITE`

D) `CONNECT`

ANSWER: B

9) Ключевое слово для группировки записей, имеющих одинаковое значение в SQL

A) `GROUP BY`

B) `CLUSTER BY`

C) `BUNDLE BY`

D) `SORT BY`

ANSWER: A

10) Ключевое слово для ограничения количества возвращаемых строк

A) `LIMIT`

B) `TOP`

C) `MAXROWS`

D) `ROWCOUNT`

ANSWER: A

11) Оператор SQL для изменения структуры существующей таблицы

A) `MODIFY TABLE`

B) `CHANGE TABLE`

C) `UPDATE TABLE`

D) `ALTER TABLE`

ANSWER: D

12) Ключевое слово для удаления дубликатов в результатах запроса

A) `UNIQUE`

B) `DISTINCT`

C) `DIFFERENT`

D) `FILTER`

ANSWER: B

13) Ключевое слово для проверки условия в SQL

A) `CHECK`

B) `VALIDATE`

C) `CONDITION`

D) `ASSERT`

ANSWER: A

14) Выражение для получения максимального значения столбца

A) `MAX(column\_name)`

B) `HIGH(column\_name)`

C) `UPPER(column\_name)`

D) `HIGHEST(column\_name)`

ANSWER: A

15) Выражение для получения минимального значения столбца

A) `MIN(column\_name)`

B) `LOW(column\_name)`

C) `LOWER(column\_name)`

D) `SMALLEST(column\_name)`

ANSWER: A

16) Выражение для подсчета количества строк в таблице

A) `COUNT(\*)`

B) `SUM(\*)`

C) `NUMBER(\*)`

D) `TOTAL(\*)`

ANSWER: A

17) Ключевое слово для соединения результатов двух запросов в один результат

A) `JOIN`

B) `CONNECT`

C) `UNION`

D) `MERGE`

ANSWER: C

18) Выражение для вычисления среднего значения столбца

A) `AVG(column\_name)`

B) `MEAN(column\_name)`

C) `AVERAGE(column\_name)`

D) `MEDIAN(column\_name)`

ANSWER: A

19) Ключевое слово для объединения строк из двух таблиц на основе условного соответствия

A) `JOIN ON`

B) `MERGE ON`

C) `UNITE ON`

D) `CONNECT ON`

ANSWER: A

20) Ключевое слово для создания индекса в таблице

A) `CREATE INDEX`

B) `MAKE INDEX`

C) `BUILD INDEX`

D) `ADD INDEX`

ANSWER: A

21) Ключевое слово для определения уникальности значений в столбце при создании таблицы

A) `PRIMARY`

B) `UNIQUE`

C) `DISTINCT`

D) `SINGLE`

ANSWER: B

22) Оператор для восстановления удаленных строк

A) `RETRIEVE`

B) `RESTORE`

C) `UNDO`

D) `ROLLBACK`

ANSWER: D

23) Ключевое слово для создания временной таблицы в SQL

A) `TEMPORARY TABLE`

B) `TEMP TABLE`

C) `TRANSIENT TABLE`

D) `VOLATILE TABLE`

ANSWER: A

24) Выражение для вычисления суммы значений столбца

A) `SUM(column\_name)`

B) `TOTAL(column\_name)`

C) `AGG(column\_name)`

D) `ADD(column\_name)`

ANSWER: A

25) Ключевое слово для проверки наличия записей, удовлетворяющих определенному условию

A) `EXISTS`

B) `CONTAINS`

C) `FOUND`

D) `CHECKS`

ANSWER: A

26) Ключевое слово для установки начального значения для автоинкрементного столбца.

A) `SET AUTO\_INCREMENT`

B) `INIT AUTO\_INCREMENT`

C) `AUTO\_INCREMENT START`

D) `START WITH`

ANSWER: D

27) Оператор для удаления столбца из таблицы.

A) `REMOVE COLUMN`

B) `DELETE COLUMN`

C) `DROP COLUMN`

D) `REMOVE FROM`

ANSWER: C

28) Ключевое слово для добавления условия в запрос, кроме WHERE.

A) `AND`

B) `OR`

C) `HAVING`

D) `IF`

ANSWER: C

29) Оператор для изменения типа данных столбца.

A) `CHANGE COLUMN`

B) `MODIFY COLUMN`

C) `ALTER COLUMN`

D) `UPDATE COLUMN`

ANSWER: B

30) Функция для преобразования строки в верхний регистр.

A) `UPPER()`

B) `UCASE()`

C) `CAPITALIZE()`

D) `CASEUP()`

ANSWER: A

31) Функция для преобразования строки в нижний регистр.

A) `LOWER()`

B) `LCASE()`

C) `MINCASE()`

D) `LOWERCASE()`

ANSWER: A

32) Оператор для сортировки результатов запроса в убывающем порядке.

A) `ASCENDING`

B) `DESCENDING`

C) `ORDER DESC`

D) `DESC`

ANSWER: D

33) Ключевое слово для добавления новой колонки в таблицу.

A) `ADD COLUMN`

B) `CREATE COLUMN`

C) `INSERT COLUMN`

D) `NEW COLUMN`

ANSWER: A

34) Оператор для ограничения результата запроса только уникальными записями.

A) `ONLY`

B) `DISTINCT`

C) `UNIQUE`

D) `SINGULAR`

ANSWER: B

35) Функция для вычисления длины строки.

A) `LENGTH()`

B) `LEN()`

C) `SIZE()`

D) `CHAR\_LENGTH()`

ANSWER: D

36) Ключевое слово для выбора всех столбцов из таблицы.

A) `ALL`

B) `EVERY`

C) `EVERYTHING`

D) `\*`

ANSWER: D

37) Оператор для удаления таблицы из базы данных.

A) `DELETE`

B) `REMOVE`

C) `DROP`

D) `ERASE`

ANSWER: C

38) Оператор для обновления данных в таблице.

A) `UPDATE`

B) `REPLACE`

C) `MODIFY`

D) `REFRESH`

ANSWER: A

39) Ключевое слово для определения столбца как первичного ключа.

A) `PRIMARY`

B) `KEY`

C) `INDEX`

D) `ID`

ANSWER: A

40) Оператор для сравнения значений в SQL.

A) `EQUAL`

B) `LIKE`

C) `COMPARE`

D) `MATCH`

ANSWER: B

41) Оператор для определения условия выборки записей в SQL.

A) `SELECT`

B) `FILTER`

C) `WHERE`

D) `CONDITION`

ANSWER: C

42) Ключевое слово для ограничения числа возвращаемых строк в SQL.

A) `ROWCOUNT`

B) `LIMIT`

C) `MAXROWS`

D) `TOP`

ANSWER: B

43) Оператор для объединения результатов двух запросов в SQL.

A) `COMBINE`

B) `UNION`

C) `JOIN`

D) `MERGE`

ANSWER: B

44) Ключевое слово для группировки записей по заданному столбцу в SQL.

A) `GROUP`

B) `GROUP BY`

C) `GROUPING`

D) `COLLECT`

ANSWER: B

45) Оператор для проверки существования записей, удовлетворяющих условию, в SQL.

A) `CHECK`

B) `VERIFY`

C) `ENSURE`

D) `EXISTS`

ANSWER: D

46) Ключевое слово для создания новой таблицы в SQL.

A) `CREATE`

B) `GENERATE`

C) `MAKE`

D) `BUILD`

ANSWER: A

47) Оператор для увеличения значения числового столбца на определенную величину.

A) `INCREMENT`

B) `ADD`

C) `RAISE`

D) `UPDATE`

ANSWER: B

48) Функция для получения текущей даты и времени в SQL.

A) `NOW()`

B) `CURRENT\_TIMESTAMP()`

C) `GETDATE()`

D) `SYSDATE()`

ANSWER: B

49) Оператор для ограничения результата запроса только уникальными значениями в указанном столбце.

A) `UNIQUE`

B) `DISTINCT`

C) `ONLY`

D) `SINGLE`

ANSWER: B

50) Ключевое слово для отсортировки результатов запроса по заданному столбцу в порядке убывания.

A) `DESC`

B) `DECREASE`

C) `LESS`

D) `DESCENDING`

ANSWER: A

51) Оператор для удаления всех записей из таблицы в SQL.

A) `DELETE`

B) `DROP`

C) `TRUNCATE`

D) `CLEAR`

ANSWER: C

52) Ключевое слово для объединения строк из двух таблиц в SQL.

A) `LINK`

B) `MERGE`

C) `CONNECT`

D) `JOIN`

ANSWER: D

53) Оператор для проверки, что значение столбца находится в списке заданных значений в SQL.

A) `IN`

B) `BETWEEN`

C) `LIKE`

D) `CONTAINS`

ANSWER: A

54) Ключевое слово для указания начального значения для автоинкрементного столбца.

A) `START`

B) `BEGIN`

C) `INIT`

D) `AUTO\_INCREMENT`

ANSWER: D

55) Функция для получения подстроки из строки в SQL.

A) `SUBSTRING()`

B) `STR()`

C) `STRING()`

D) `SUBSTR()`

ANSWER: D

56) Ключевое слово для сортировки результатов запроса в порядке возрастания.

A) `ASC`

B) `UP`

C) `ASCENDING`

D) `INCREASE`

ANSWER: A

57) Оператор для выполнения логического объединения условий в SQL.

A) `AND`

B) `OR`

C) `NOT`

D) `XOR`

ANSWER: A

58) Ключевое слово для ограничения результата запроса только уникальными записями.

A) `DISTINCT`

B) `UNIQUE`

C) `ONLY`

D) `SINGULAR`

ANSWER: A

59) Оператор для определения условия выборки записей в SQL.

A) `SELECT`

B) `WHERE`

C) `FILTER`

D) `CONDITION`

ANSWER: B

60) Ключевое слово для определения столбца как первичного ключа.

A) `PRIMARY`

B) `KEY`

C) `INDEX`

D) `ID`

ANSWER: A

61) Оператор для удаления таблицы из базы данных.

A) `DELETE`

B) `REMOVE`

C) `DROP`

D) `ERASE`

ANSWER: C

62) Оператор для обновления данных в таблице.

A) `UPDATE`

B) `REPLACE`

C) `MODIFY`

D) `REFRESH`

ANSWER: A

63) Ключевое слово для определения столбца как внешний ключ.

A) `PRIMARY`

B) `FOREIGN`

C) `REFERENCES`

D) `KEY`

ANSWER: B

64) Оператор для добавления новой колонки в таблицу.

A) `INSERT COLUMN`

B) `ADD COLUMN`

C) `CREATE COLUMN`

D) `NEW COLUMN`

ANSWER: B

65) Ключевое слово для ограничения числа возвращаемых строк в SQL.

A) `ROWCOUNT`

B) `LIMIT`

C) `MAXROWS`

D) `TOP`

ANSWER: B

66) Оператор для сортировки результатов запроса в порядке убывания.

A) `DESC`

B) `DECREASE`

C) `LESS`

D) `DESCENDING`

ANSWER: A

67) Функция для получения текущей даты и времени в SQL.

A) `NOW()`

B) `CURRENT\_TIMESTAMP()`

C) `GETDATE()`

D) `SYSDATE()`

ANSWER: B

68) Оператор для ограничения результата запроса только уникальными значениями в указанном столбце.

A) `UNIQUE`

B) `DISTINCT`

C) `ONLY`

D) `SINGLE`

ANSWER: B

69) Ключевое слово для отсортировки результатов запроса по заданному столбцу в порядке возрастания.

A) `ASC`

B) `UP`

C) `ASCENDING`

D) `INCREASE`

ANSWER: A

70) Функция для преобразования строки в верхний регистр.

A) `UPPER()`

B) `UCASE()`

C) `CAPITALIZE()`

D) `CASEUP()`

ANSWER: A

71) Оператор для удаления всех записей из таблицы в SQL.

A) `DELETE`

B) `DROP`

C) `TRUNCATE`

D) `CLEAR`

ANSWER: C

72) Оператор для объединения результатов двух запросов в SQL.

A) `UNION`

B) `JOIN`

C) `MERGE`

D) `COMBINE`

ANSWER: A

73) Оператор для проверки, что значение столбца находится в списке заданных значений в SQL.

A) `IN`

B) `BETWEEN`

C) `LIKE`

D) `CONTAINS`

ANSWER: A

74) Ключевое слово для указания начального значения для автоинкрементного столбца.

A) `START`

B) `BEGIN`

C) `INIT`

D) `AUTO\_INCREMENT`

ANSWER: D

75) Функция для получения подстроки из строки в SQL.

A) `SUBSTRING()`

B) `STR()`

C) `STRING()`

D) `SUBSTR()`

ANSWER: D

76) Ключевое слово для сортировки результатов запроса в порядке возрастания.

A) `ASC`

B) `UP`

C) `ASCENDING`

D) `INCREASE`

ANSWER: A

77) Оператор для выполнения логического объединения условий в SQL.

A) `AND`

B) `OR`

C) `NOT`

D) `XOR`

ANSWER: A

78) Ключевое слово для ограничения результата запроса только уникальными записями.

A) `DISTINCT`

B) `UNIQUE`

C) `ONLY`

D) `SINGULAR`

ANSWER: A

79) Оператор для определения условия выборки записей в SQL.

A) `SELECT`

B) `WHERE`

C) `FILTER`

D) `CONDITION`

ANSWER: B

80) Ключевое слово для определения столбца как первичного ключа.

A) `PRIMARY`

B) `KEY`

C) `INDEX`

D) `ID`

ANSWER: A

81) Ключевое слово для указания начального значения для автоинкрементного столбца.

A) `BEGIN`

B) `INIT`

C) `AUTO\_INCREMENT\_START`

D) `START WITH`

ANSWER: D

82) Оператор для удаления всех записей из таблицы в SQL.

A) `DELETE ALL`

B) `CLEAR TABLE`

C) `TRUNCATE TABLE`

D) `REMOVE ALL`

ANSWER: C

83) Оператор для объединения результатов двух запросов в SQL.

A) `UNION`

B) `JOIN`

C) `MERGE`

D) `COMBINE`

ANSWER: A

84) Оператор для проверки, что значение столбца находится в списке заданных значений в SQL.

A) `IN`

B) `BETWEEN`

C) `LIKE`

D) `CONTAINS`

ANSWER: A

85) Функция для получения подстроки из строки в SQL.

A) `SUBSTRING()`

B) `STR()`

C) `STRING()`

D) `SUBSTR()`

ANSWER: D

86) Ключевое слово для сортировки результатов запроса в порядке убывания.

A) `DESC`

B) `DECREASE`

C) `LESS`

D) `DESCENDING`

ANSWER: A

87) Оператор для выполнения логического объединения условий в SQL.

A) `AND`

B) `OR`

C) `NOT`

D) `XOR`

ANSWER: A

88) Ключевое слово для ограничения результата запроса только уникальными записями.

A) `DISTINCT`

B) `UNIQUE`

C) `ONLY`

D) `SINGULAR`

ANSWER: A

89) Оператор для определения условия выборки записей в SQL.

A) `SELECT`

B) `WHERE`

C) `FILTER`

D) `CONDITION`

ANSWER: B

90) Ключевое слово для определения столбца как первичного ключа.

A) `PRIMARY`

B) `KEY`

C) `INDEX`

D) `ID`

ANSWER: A

91) Оператор для удаления таблицы из базы данных.

A) `DELETE`

B) `REMOVE`

C) `DROP`

D) `ERASE`

ANSWER: C

92) Оператор для обновления данных в таблице.

A) `UPDATE`

B) `REPLACE`

C) `MODIFY`

D) `REFRESH`

ANSWER: A

93) Ключевое слово для определения столбца как внешний ключ.

A) `PRIMARY`

B) `FOREIGN`

C) `REFERENCES`

D) `KEY`

ANSWER: B

94) Оператор для добавления новой колонки в таблицу.

A) `INSERT COLUMN`

B) `ADD COLUMN`

C) `CREATE COLUMN`

D) `NEW COLUMN`

ANSWER: B

95) Ключевое слово для ограничения числа возвращаемых строк в SQL.

A) `ROWCOUNT`

B) `LIMIT`

C) `MAXROWS`

D) `TOP`

ANSWER: B

96) Ключевое слово для указания начального значения для автоинкрементного столбца.

A) `BEGIN`

B) `INIT`

C) `AUTO\_INCREMENT\_START`

D) `START WITH`

ANSWER: D

97) Оператор для слияния строк из двух таблиц на основе заданного условия.

A) MERGE

B) JOIN

C) COMBINE

D) UNITE

ANSWER: B

98) Оператор для объединения результатов двух запросов в SQL.

A) `UNION`

B) `JOIN`

C) `MERGE`

D) `COMBINE`

ANSWER: A

99) Оператор для проверки, что значение столбца находится в списке заданных значений в SQL.

A) `IN`

B) `BETWEEN`

C) `LIKE`

D) `CONTAINS`

ANSWER: A

100) Функция для получения подстроки из строки в SQL.

A) `SUBSTRING()`

B) `STR()`

C) `STRING()`

D) `SUBSTR()`

ANSWER: D

1