

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ В.Ф. УТКИНА»

КАФЕДРА

«ВЫЧИСЛИТЕЛЬНАЯ И ПРИКЛАДНАЯ МАТЕМАТИКА»

**МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ
ФТД.В «ПРОГРАММИРОВАНИЕ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ»**

Направление подготовки – 09.03.04 «Программная инженерия»

Направленность (профиль) подготовки
«Программная инженерия»

ОПОП академического бакалавриата

Квалификация выпускника – бакалавр

Формы обучения – очная

Рязань 2022 г

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Рекомендуется следующим образом организовать время, необходимое для изучения дисциплины:

Для освоения лекционного материала следует: изучить конспект лекции в тот же день, после лекции: 10 – 15 минут, повторно прочитать конспект лекции за день перед следующей лекцией: 10 – 15 минут. Также следует изучить теоретический лекционный материал по рекомендуемому учебнику/ учебному пособию: 1 час в неделю.

Следует максимально использовать лекционное время для изучения дисциплины, понимания лекционного материала и написания конспекта лекций. В процессе лекционного занятия студент должен уметь выделять важные моменты и основные положения. При написании конспекта лекций следует придерживаться следующих правил и рекомендаций.

1. При ведении конспекта рекомендуется структурировать материал по разделам, главам, темам. Вести нумерацию формул. Выделять по каждой теме постановку задачи, основные положения, выводы. Кратко записывать те пояснения лектора, которые показались особенно важными. Это позволит при подготовке к сдаче зачёта и экзамена не запутаться в структуре лекционного материала.
2. Лекционный материал следует записывать в конспект лишь после того, как излагаемый лектором тезис будет вами дослушан до конца и понят.
3. При конспектировании следует отмечать непонятные, на данном этапе, положения, доказательства и пр.
4. Рекомендуется по каждой теме выразить свое мнение, комментарий, вывод.

Подготовка к практическим занятиям:

Практические занятия по дисциплине существенно дополняют лекции. В процессе анализа теоретических положений и решения практических задач студенты расширяют и углубляют свои знания, полученные из лекционного курса и учебников, приобретают умение применять общие закономерности к конкретным случаям. В процессе решения задач развивается логическое мышление и вырабатываются навыки вычислений, работы со справочной литературой. Практические занятия способствуют закреплению знаний и практических навыков, формированию конструктивного стиля мышления, расширению кругозора.

При подготовке к практическому занятию необходимо внимательно ознакомиться с соответствующим теоретическим материалом по конспекту

лекций и рекомендованному учебнику, затем изучить конспект или материалы предыдущего практического занятия и выполнить заданное расчетное задание: 1 – 2 часа в неделю.

Следует максимально использовать аудиторное время практических занятий. В процессе занятия студент должен активно участвовать в дискуссиях, обсуждениях и решениях практических задач и вести конспект практических занятий отдельно от конспекта лекций.

Дополнительно в часы самостоятельной работы студенты могут повторно решить задачи, с которыми они плохо освоились во время аудиторных занятий, и обязательно те задачи, которые не получились дома при предыдущей подготовке к практическим занятиям.

Подготовка к лабораторным работам.

Перед началом проведения лабораторной работы необходимо ознакомиться с методическими указаниями к данной лабораторной работе, внимательно ознакомиться с заданием и желательно заранее выполнить подготовку проекта в используемой инструментальной среде, чтобы время лабораторного занятия использовать для исправления ошибок, модификации проекта и защиты данной работы.

Выполнение каждой из запланированных работ заканчивается предоставлением отчета. Требования к форме и содержанию отчета приведены в методических указаниях к лабораторным работам или определяются преподавателем на первом занятии. Отчет по лабораторной работе студент должен начать оформлять еще на этапе подготовки к ее выполнению. Допускаясь к лабораторной работе, каждый студент должен представить преподавателю «заготовку» отчета, содержащую: оформленный титульный лист или название и номер работы при ведении общего конспекта, цель работы, задание, проект решения, полученные результаты, выводы.

Изучение методических указаний к лабораторной работе – 2 часа перед выполнением лабораторной работы и в ходе разработки проекта и 2 часа для оформления отчета, отладки проекта и подготовки к сдаче работы.

После выполнения лабораторной работы необходимо согласовать полученные результаты с преподавателем. Важным этапом является защита лабораторной работы. В процессе защиты студент отвечает на вопросы преподавателя, касающиеся теоретического материала, относящегося к данной работе, и проекта, реализующего его задание, комментирует полученные в ходе работы результаты. При подготовке к защите лабораторной работы рекомендуется ознакомиться со списком вопросов по изучаемой теме и попытаться самостоятельно на них ответить, используя конспект лекций и рекомендованную

литературу. Кроме чтения учебной литературы рекомендуется активно использовать информационные ресурсы сети Интернет по изучаемой теме.

Подготовка к сдаче экзамена или зачета:

Экзамен/зачет – форма промежуточной проверки знаний, умений, навыков, степени освоения дисциплины. Главная задача экзамена/зачета состоит в том, чтобы у студента по окончании изучения данной дисциплины сформировались определенное представление об общем содержании дисциплины, определенные теоретические знания и практические навыки, определенный кругозор. Готовясь к экзамену/зачету, студент приводит в систему знания, полученные на лекциях, на практических и лабораторных занятиях, разбирается в том, что осталось непонятным, и тогда изучаемая им дисциплина может быть воспринята в полном объеме с присущей ей строгостью и логичностью, ее практической направленностью.

Экзамены/зачеты дают возможность преподавателю определить теоретические знания студента и его практические навыки при решении определенных прикладных задач. Оцениваются: понимание и степень усвоения теоретического материала; степень знакомства с основной и дополнительно литературой, а также с современными публикациями; умение применить теорию к практике, решать определенные практические задачи данной предметной области, правильно проводить расчеты и т. д.; знакомство с историей данной науки; логика, структура и стиль ответа, умение защищать выдвигаемые положения.

Значение экзаменов/зачетов не ограничивается проверкой знаний, являясь естественным завершением обучения студента по данной дисциплине, они способствуют обобщению и закреплению знаний и умений, приведению их в стройную систему, а также устранению возникших в процессе обучения пробелов.

Подготовка к экзамену/зачету – это тщательное изучение и систематизация учебного материала, осмысление и запоминание теоретических положений, формулировок, формул, установление и осмысление внутривидовых связей между различными темами и разделами дисциплины, закрепление теоретических знаний путем решения определенных задач.

Перед экзаменом назначается консультация, ее цель – дать ответы на вопросы, возникшие в ходе самостоятельной подготовки студента, студент имеет возможность получить ответ на все неясные ему вопросы, кроме того, преподаватель будет отвечать на вопросы других студентов, что будет способствовать повторению и закреплению знаний всех присутствующих. Преподаватель на консультации, как правило, обращает внимание на те разделы, по которым на предыдущих экзаменах ответы были

неудовлетворительными, а также фиксирует внимание на наиболее трудных разделах курса.

На непосредственную подготовку к экзамену обычно дается 3 – 5 дней. Этого времени достаточно для углубления, расширения и систематизации знаний, полученных в ходе обучения, на устранение пробелов в знании отдельных вопросов, для определения объема ответов на каждый из вопросов рабочей программы дисциплины.

Планируйте подготовку к зачету/экзамену, учитывая сразу несколько факторов: неоднородность в сложности учебного материала и степени его проработки в ходе обучения, свои индивидуальные способности. Рекомендуется делать перерывы в занятиях через каждые 50-60 минут на 10 минут. После 3-4 часов занятий следует сделать часовой перерыв. Чрезмерное утомление приведет к снижению тонуса интеллектуальной деятельности. Целесообразно разделять весь рабочий день на три рабочих периода – с утра до обеда, с обеда до ужина и с ужина до сна. Каждый рабочий период дня должен заканчиваться отдыхом не менее 1 часа. Работая в сессионном режиме, студент имеет возможность увеличить время занятий с 10 (как требовалось в семестре) до 12 часов в сутки.

Подготовку к экзаменам или зачетам следует начинать с общего планирования своей деятельности. С определения объема материала, подлежащего проработке, необходимо внимательно сверить свои конспекты с программой дисциплины, чтобы убедиться, все ли разделы отражены в лекциях, отсутствующие темы изучить по учебнику. Второй этап предусматривает системное изучение материала по данному предмету с обязательной записью всех выкладок, выводов, формул. На третьем этапе – этапе закрепления – полезно чередовать углубленное повторение особенно сложных вопросов с беглым повторением всего материала.

Рекомендации по работе с литературой:

Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта изучаются и книги по данному предмету. Литературу по дисциплине рекомендуется читать как в бумажном, так и в электронном виде (если отсутствует бумажный аналог). Полезно использовать несколько учебников и пособий по дисциплине. Рекомендуется после изучения очередного параграфа ответить на несколько вопросов по данной теме. Кроме того, полезно мысленно задать себе следующие вопросы (и попробовать ответить на них): «о чем этот параграф?», «какие новые понятия введены, каков их смысл?», «зачем мне это нужно по специальности?».

Рекомендуется самостоятельно изучать материал, который еще не прочитан на лекции и не применялся на лабораторном или практическом занятии, тогда занятия будут гораздо понятнее. В течение недели рекомендуется выбрать время (1 час) для работы с литературой.

Лабораторная работа №1. Знакомство с IDE Android Studio

Цель работы:

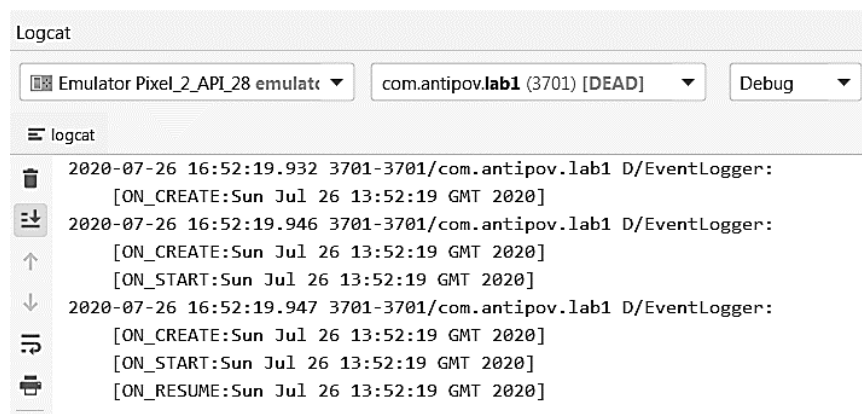
Изучение основ работы с IDE Android Studio. Использование LogCat.
Разработка и отладка первого мобильного приложения.

Задание:

Напишите статический класс **EventLogger**, который будет хранить в себе последовательность всех последних вызовов обработчиков событий, а также время их возникновения. Класс должен иметь следующие методы:

- **add()** – добавляет новое событие в класс.
- **print()** – выводит в лог список последних событий и время их возникновения.
- **clear()** – полностью очищает список событий.

Вывод списка событий в лог должен быть следующего вида:



```
Logcat
Emulator Pixel_2_API_28 emulatr  com.antipov.lab1 (3701) [DEAD]  Debug
logcat
2020-07-26 16:52:19.932 3701-3701/com.antipov.lab1 D/EventLogger:
[ON_CREATE:Sun Jul 26 13:52:19 GMT 2020]
2020-07-26 16:52:19.946 3701-3701/com.antipov.lab1 D/EventLogger:
[ON_CREATE:Sun Jul 26 13:52:19 GMT 2020]
[ON_START:Sun Jul 26 13:52:19 GMT 2020]
2020-07-26 16:52:19.947 3701-3701/com.antipov.lab1 D/EventLogger:
[ON_CREATE:Sun Jul 26 13:52:19 GMT 2020]
[ON_START:Sun Jul 26 13:52:19 GMT 2020]
[ON_RESUME:Sun Jul 26 13:52:19 GMT 2020]
```

Далее замените во всех обработчиках событий класса **MainActivity** вызовы **Log.d** на вызовы методов класса **EventLogger.add** для добавления нового события и следом за ним **EventLogger.print** для вывода всего сохранённого списка событий. В обработчике **onDestroy** также в конце добавьте вызов **EventLogger.clear** для очистки списка событий.

Лабораторная работа №2. Создание адаптивного интерфейса мобильного приложения

Цель работы:

Изучение принципов создания адаптивного интерфейса мобильного приложения при помощи редактора разметок, а также базовых элементов управления.

Задание:

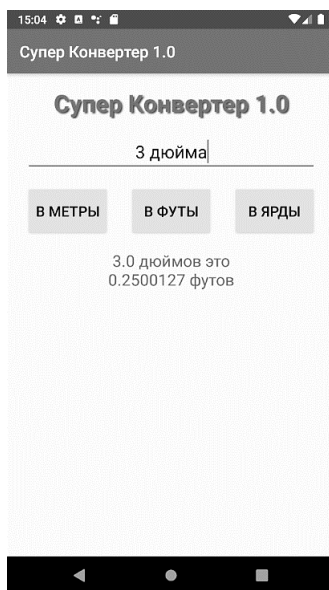
Напишите класс-синглетон **UnitConverter**, который будет конвертировать значения из одних единиц измерения в другие, и возвращать результат. Необходимые единицы измерения для конвертации представлены в таблице ниже:

Длина: 1 метр = 39,37 дюйма 1 метр = 3,281 фута 1 метр = 1,094 ярда	Масса: 1 килограмм = 2,205 фунта 1 килограмм = 35,274 унций 1 килограмм = 0,1574 стона	Объём: 1 Литр = 0,264172 галлона 1 Литр = 1,05669 кварта 1 Литр = 2,113 пинта
---	--	---

Интерфейс программы должен быть реализован следующим образом: пользователь вводит в текстовом поле значение и меру измерения (например, «25 метров» или «15 килограмм»), нажимает Enter, и далее выбирает необходимую единицу измерения для конвертации, нажимая на соответствующую кнопку:

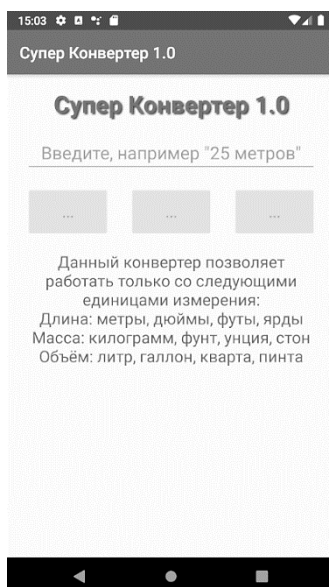


При наборе пользователем другой единицы измерения, варианты конвертации на кнопках меняются:



Если пользователь допустил ошибку при вводе, сообщается об ошибке и выводится подсказка.

По умолчанию, при запуске программы или если ничего не введено, в текстовом поле высвечивается подсказка, кнопки конвертации не доступны, а ниже выводится подсказка с возможными вариантами конвертации:



Лабораторная работа №3.

Применение различных типов разметок.

Иерархия элементов управления.

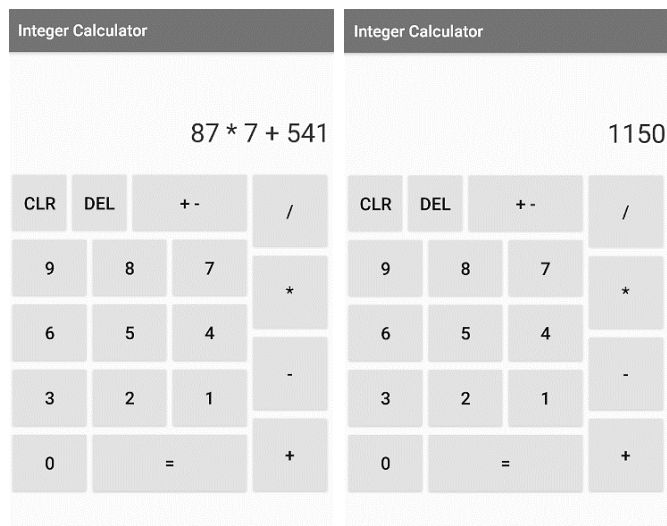
Цель работы:

Продолжение изучения принципов создания адаптивного интерфейса мобильного приложения, использование различных типов разметок, создание иерархии элементов управления.

Задание:

Создать класс **IntegerCalculator**, позволяющий с помощью посылаемых ему команд осуществлять ввод арифметического выражения, состоящего из базовых арифметических операций (сложение, вычитание, умножение, деление) и **целых** чисел (вводимых цифрами по-разрядно). Используя данный класс создать интерфейс калькулятора для ввода арифметического выражения и получения результата его вычисления (**приоритет операций можно не учитывать**). У пользователя также должна быть возможность смены знака (+-) у текущего набираемого числа, очистки введённого выражения (CLR), удаление из выражения последней введённой операции либо числа (DEL) и вычисление введённого выражения (=).

Требуемый интерфейс приложения представлен ниже:



Лабораторная работа №4.

Создание многоэкранного мобильного приложения.

Локализация интерфейса.

Цель работы:

Изучение принципов создания многоэкранного мобильного приложения с возможностью локализации интерфейса.

Задание:

Создать приложение «Анкеты соискателей», позволяющее создавать анкеты для соискателей работы, просматривать их содержимое, редактировать, удалять, а также просматривать статистику по всем анкетам. Приложение состоит из 4 активностей – экранов. В программе все текстовые данные должны быть вынесены в ресурсы, а также должна быть добавлена английская локализация (можно использовать Google Translate).

Первый экран – главное меню. Содержит заголовок приложения, кнопки перехода к другим экранам, а также краткую информацию по количеству уже добавленных анкет:



Второй экран – добавление новой анкеты (или редактирование существующей):

Новая анкета

Ф.И.О.:

Фамилия

Имя

Отчество

Образование:

- Неполное среднее
 Среднее
 Высшее

Отрасль:

Государственные организации

Опыт работы:

НАЗАД

ОЧИСТИТЬ

ДОБАВИТЬ

На этом экране можно ввести следующие поля:

Ф.И.О.:

Фамилия

Имя

Отчество

Образование:

- Неполное среднее
 Среднее
 Высшее

Отрасль:

Государственные организации

Добывающая отрасль

Информационные технологии

Услуги

Опыт работы:

Нет опыта

От 1 до 3 лет

От 3 до 6 лет

Более 6 лет

Желаемый уровень зарплаты:

38 550 руб



Тип желаемой занятости:

Полная занятость Разовая работа

Частичная занятость Стажировка

НЕТ ФИЗИЧЕСКИХ ОГРАНИЧЕНИЙ

НЕТ СУДИМОСТИ

Дополнительная информация от
соискателя:

Укажите здесь дополнительную информацию

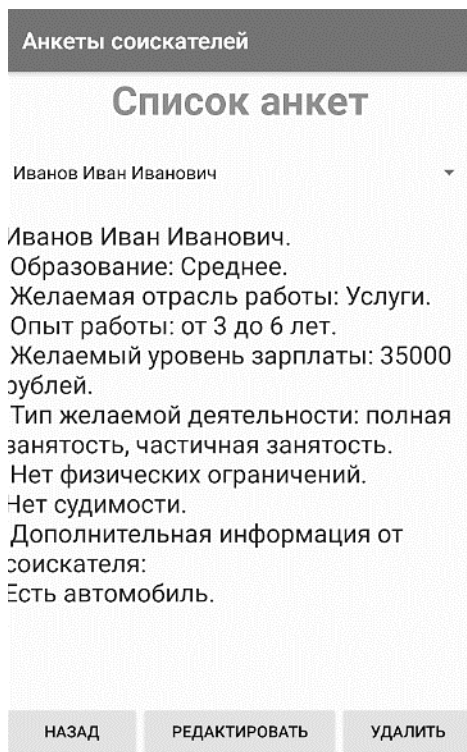
Я подтверждаю подлинность
предоставленной информации и даю
свое согласие на обработку своих
персональных данных



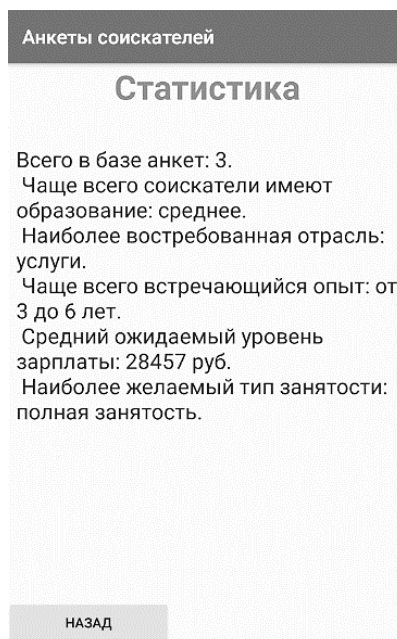
Все эти элементы должны быть в контейнере **ScrollView**. Без последнего пункта (подтверждение) не давать пользователю добавить анкету (кнопка «добавить» должна быть не активной).

При нажатии кнопки «очистить» все поля принимают значения по умолчанию.

На третьем экране можно просмотреть списком все добавленные анкеты, а также удалить или редактировать выбранную анкету (редактирование осуществляется в той же активности, что и при добавлении новой анкеты):



И наконец на четвертом экране выводится вся статистика по добавленным анкетам:



Практическое занятие по теме № 1

Вопросы:

1. Что такое мобильная операционная система и какие задачи она решает?
2. Какие аппаратные функции обычно присутствуют в мобильной ОС?
3. Какие достоинства и недостатки есть у Android?
4. Какие достоинства и недостатки есть у iOS?
5. Какие задачи решает Android Studio?
6. Что такое уникальный идентификатор приложения Android?
7. Для чего устанавливается минимальный уровень API приложения?
8. Что такое навигатор проекта?
9. Какими способами можно тестировать приложения в Android Studio?
10. Зачем нужна панель сборки?
11. Зачем нужна панель LogCat?
12. Каким образом в коде приложения можно выводить сообщения в логи и что такое уровни важности сообщений?
13. Зачем нужна метка (TAG) сообщения в логе?
14. Зачем нужен файл AndroidManifest.xml?
15. Что такое Активность (Activity)?
16. Что такое Представления (View)?
17. Какие состояния Активности бывают?
18. Что такое стек Активностей?
19. Когда вызываются обработчики событий onCreate и onDestroy?
20. Когда вызываются обработчики событий onPause и onResume?
21. Когда вызываются обработчики событий onStart, onRestart и onStop?
22. Для чего нужны обработчики событий onSaveInstanceState и onRestoreInstanceState?

Практическое занятие по теме № 2

Вопросы:

1. Что такое разметка (Layout) и для чего она нужна?
2. Для чего нужен класс представлений (View)?
3. Для чего нужен класс группы представлений (ViewGroup)?
4. Какие режимы отображения есть в редакторе разметки?

5. В чем отличие отображение интерфейса в режиме design и в режиме blueprint?
6. Для чего нужно окно палитры компонентов?
7. Для чего нужно окно дерева компонентов?
8. Что такое атрибуты компонента?
9. Что такое Constraint Layout?
10. Для чего задаются ограничения в Constraint Layout?
11. Какие единицы измерения используются в Android?
12. Для чего используется привязка компонента?
13. Для чего используются значения ограничения 0dp для компонентов?
14. Для чего нужны направляющие (guidelines)?
15. Что делают функции Organize, Align и Center?
16. Для чего применяется функция цепь (chains)?
17. Какие есть стили цепи (chains)?
18. Для чего нужен компонент TextView?
19. Каким образом из Java-кода можно обращаться к компонентам интерфейса?
20. Для чего нужен компонент EditText?
21. Какие виды компонента EditText есть в палитре и для чего они нужны?
22. Для чего нужен компонент Button?
23. Какие есть способы обработки событий компонентов?
24. Каким образом задаётся ширина и высота компонентов?
25. Для чего нужны значения 0dp (match constraints), wrap_content и match_parent для размеров компонентов?
26. Зачем задаётся вес (weight) компонентов?

Практическое занятие по теме № 3

Вопросы:

1. Что такое LinearLayout?
2. Для чего используется константа match_parent?
3. Для чего указывается вес компонента?
4. Для чего используется атрибут android:weightSum?
5. Зачем нужен компонент распорка (Space)?

6. Что такое `TableLayout`?
7. Что такое `TableRow`?
8. Какие правила существуют для `TableLayout`?
9. Для чего используется атрибут `layout_span`?
10. Что такое `FrameLayout`?
11. Для чего используется атрибут `layout_margin` в `FrameLayout`?
12. Для чего используется атрибут `layout_gravity` в `FrameLayout`?
13. В чем отличие атрибутов `layout_gravity` и `gravity`?
14. Для чего нужен класс `CompoundButton`?
15. Для чего нужен компонент `CheckBox`?
16. Для чего нужен компонент `ToggleButton`?
17. Как поменять текст для состояний `ToggleButton`?
18. Для чего нужны компоненты `RadioButton` и `RadioGroup`?
19. Как в `RadioGroup` узнать `id` выбранного переключателя?
20. Для чего нужен компонент `Switch`?
21. В чем отличие в сценарии использования компонентов `Switch`, `CheckBox` и `ToggleButton`?
22. Для чего нужен компонент `ProgressBar`?
23. В чем отличие в сценарии использования детерминированного и недетерминированного `ProgressBar`?
24. Для чего нужен компонент `SeekBar`?
25. Зачем нужен атрибут `android:tag`?

Практическое занятие по теме № 4

Вопросы:

1. Что такое ресурсы приложения и где они хранятся?
2. Для чего отделяют код от ресурсов, какую выгоду это даёт разработчику?
3. Какие виды ресурсов есть и в каких подкаталогах они хранятся?
4. Какие типы простых значений (`values`) вы знаете?
5. Как добавить ресурсный файл?
6. Как получить идентификатор ресурса из кода?
7. Как получить экземпляр ресурса из кода?

8. Как обратиться к ресурсу из другого ресурса?
9. Как обращаться к системным ресурсам?
10. Зачем нужны стили и как их задавать?
11. Что такое альтернативные ресурсы и для чего они нужны?
12. Как в андроиде происходит локализация приложения?
13. Как добавить изображение в ресурсы?
14. Что такое менеджер ресурсов и для чего он нужен?
15. Что такое ListView?
16. Как заполняются данными компоненты ListView?
17. Что такое класс Адаптера и для чего он нужен?
18. Зачем классу адаптера нужен шаблон разметки?
19. Зачем нужен метод notifyDataSetChanged классу Адаптера?
20. Какой метод позволяет прокрутить ListView до нужной позиции?
21. Зачем ListView нужны методы addHeaderView() и addFooterView()?
22. Зачем нужен атрибут android:entries?
23. Что такое Spinner?
24. Как в компоненте Spinner получить выбранный элемент?
25. Как в компоненте Spinner получить номер выбранного элемента?
26. Зачем нужны ScrollView и HorizontalScrollView?
27. Как добавить несколько компонентов в ScrollView?
28. Какие методы позволяют управлять пролистыванием компонента ScrollView?
29. Что такое намерения и зачем они нужны?
30. Что нужно сделать, чтобы создать и запустить вторую активность?
31. Как получать результаты работы активности?
32. В каких случаях возвращается Activity.RESULT_OK, а в каких Activity.RESULT_CANCELED?

Библиографический список

1. Операционная система Android. Учебное пособие / М.А. Дмитриев, А.В. Зуйков, А.А. Кузин, П.Е. Минин, А.М. Рапетов, А.С. Самойлов, М.И. Фроимсон, В.Б. Холявин, Д.В. Шевченко; по ред. И.Ю. Жукова. М.: НИЯУ МИФИ, 2012. 64 с.

2. Дейтел Х. Android для разработчиков. – СПб.: Питер, 2016. – 512 с.
3. Делессо К Создание приложений для Android. – М.: Эксмо, 2015. – 528 с.
4. Гриффитс Д., Гриффитс Д., Head First. Программирование для Android. – СПб.: Питер, 2016. – 704 с.
5. Варакин М. Разработка мобильных приложений под Android. – Москва, УЦ
6. «Специалист» при МГТУ им. Баумана, 2012. – 128
7. Цехнер М. Программирование игр под Android. – СПб.: Питер, 2013. – 668 с.
8. Майер Н Android. Программирование приложений для планшетных компьютеров и смартфонов. – М.: Эксмо, 2013. — 815 с.
9. Харди Б., Филлипс Б. Android. Программирование для профессионалов. – СПб.: Питер, 2016. – 640 с.
10. Нахавандипур В. iOS Разработка приложений для iPhone, iPad и iPod. – СПб.: Питер, 2013 — 864 с.
11. Пайлон Д., Пайлон Т. Программируем для iPhone и iPad — СПб.: Питер, 2014. — 336 с.

Составил: к.т.н., доцент кафедры
«Вычислительная и прикладная
математика»

Антипов О.В.

Заведующий кафедрой вычислительной и
прикладной математики, д-р техн. наук,
профессор

Овечкин Г.В.