

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Луганский государственный университет имени Владимира Даля»  
(ФГБОУ ВО «ЛГУ им. В. Даля»)

Северодонецкий технологический институт  
Кафедра информационных технологий, приборостроения и электротехники

УТВЕРЖДАЮ:  
Врио. директора СТИ (филиал)  
ФГБОУ ВО «ЛГУ им. В. Даля»  
Ю.В. Бородач  
(подпись)  
«20» 2024 года



**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

«Р

»

По направлению подготовки: 09.03.02 «Информационные системы и технологии»

Про «Информационные ситемы и технологии»

## Лист согласования РПУД

Рабочая программа учебной дисциплины «  
» по направлению подготовки: 09.03.02 «Информационные системы и технологии»  
(программа бакалавриата «Информационные ситемы и технологии») – 43 с.

Рабочая программа учебной дисциплины «  
» разработана в соответствии Федеральным государственным образовательным  
стандартом высшего образования по направлению подготовки 09.03.02 «Информационные  
системы и технологии» утвержденный приказом Министерства науки и высшего образования  
Российской Федерации от 19 2017 . 926 (

1456

26.11.2020 ., 83 08.02.2021 ., 662 19.07.2022 ., 208 27.02.2023 .).

СОСТАВИТЕЛЬ:

. .

Рабочая программа дисциплины утверждена на заседании кафедры информационных  
технологий, приборостроения и электротехники « 05 » сентября 2024 г., протокол № 1.

Заведующий кафедрой ИТПЭ  В.Г. Чебан

Переутверждена: « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г., протокол № \_\_\_\_.

Рекомендована на заседании учебно-методической комиссии Северодонецкого  
технологического института (филиал) федерального государственного бюджетного  
образовательного учреждения высшего образования «Луганский государственный университет  
имени Владимира Даля» « 16 » сентября 2024 г., протокол № 1.

Председатель учебно-методической комиссии  
СТИ (филиал) ФГБОУ ВО «ЛГУ им. В.Даля»

 Ю.В. Бородач

## Структура и содержание дисциплины

### 1. Цели и задачи дисциплины, ее место в учебном процессе

Дисциплина «Разработка приложений для мобильных устройств» представляет собой изложение основных сведений о планировании, написании, тестировании, отладке и распространении приложений для мобильной платформы Android.

**Цель изучения дисциплины** - знакомство студентов с архитектурой Android приложений и формирование навыков работы с пакетами разработки, тестирования и отладки приложений для мобильных устройств.

**Задачи изучения дисциплины:**

- получение представления о жизненном цикле приложений, их структуре, программном манифесте и внешних ресурсах;
- изучение инструментов разработки; знакомство с элементами пользовательского интерфейса, работа с файлами, базами данных, пользовательскими настройками, разделяемыми данными и межпрограммном взаимодействии.

### 2. Место дисциплины в структуре ОПОП ВО.

Дисциплина «Разработка приложений для мобильных устройств» дисциплина входит в обязательную часть учебного плана по направлению подготовки 09.03.02 Информационные системы и технологии.

Содержание дисциплины является логическим продолжением содержания дисциплины «Кроссплатформенное программирование».

Является основой для написания выпускной квалификационной работы.

### 3. Требования к результатам освоения содержания дисциплины

Студенты, завершившие изучение дисциплины «Разработка приложений для мобильных устройств», должны

**знать:**

- архитектуру Android системы и Android приложения
- основные компоненты и правила их упаковки
- механизмы безопасности, связанные с загрузкой и отладкой Android приложений
- архитектуру визуальных компонентов Android
- номенклатуру интерактивных элементов и механизмов их размещения
- различные варианты организации обработчиков
- знать методы хранения настроек уровня приложения
- назначение компонента Activity, его жизненный цикл

- механизмы связи между компонентами приложений
- механизмы сохранения состояния Activity
- назначение механизмы Intent
- механизмы сериализации сложных данных для передачи их между компонентами
- особенности файловой системы и операций в Android
- механизмы чтения XML документов
- механизм касаний, реализованный в Android
- методы декларативного программирования графики и анимации в Android
- архитектуру Android процесса
- механизмы связи между фоновыми и UI потоком
- знать назначение компонента Service
- назначение механизма уведомлений

#### **уметь:**

- управлять ключами разработчика
- работать с визуальными элементами из кода
- назначать обработчики событий в визуальных элементах
- -уметь программировать меню и всплывающие сообщения
- программировать поведение Activity в Task
- программировать активацию компонента на основе механизма Uri
- программировать работу с файлами
- использовать стандартные диалоги Android
- обрабатывать одиночные и множественные касания
- работать с объектом Canvas из основного потока приложения
- программировать вывод графических примитивов и текста
- работать с компонентом Surface из фонового потока
- программировать фоновую обработку
- запускать сервис, передавать задачу для фоновой обработки
- программировать связывание с запущенным сервисом
- программировать обратную связь с использованием PendingIntent и BroadcastReceiver
- программировать уведомления в Activity и Service
- программировать кастомные уведомления и организовывать обратную связь

#### **владеть:**

- навыками авторизации и удаленной отладки на Android устройствах
- навыками конфигурации Android устройств для разработки приложений

- навыками загрузки и обновления приложений на Android устройствах
- навыками верстки интерфейса пользователя с применением различных типов механизмов размещения
- навыками верстки с поддержкой различных размеров, ориентаций экранов и пользовательских языков
- навыками программирования передачи данных и получения результата из Activity
- навыками формирования и запросов с использованием Uri
- навыками программирования пользовательских диалогов
- навыками программирования распознавания жестов
- навыками работы с классами Handler и AsyncTask

Перечисленные результаты образования являются основой для формирования следующих компетенций (в соответствии с ФГОС ВО и требованиями к результатам освоения основной образовательной программы (ОПОП ВО):

**общефессиональных:**

ОПК-2.1: Знать принципы работы современных принципы работы информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности

ОПК-2.2: Уметь выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности

ОПК-2.3: Иметь навыки применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности

ОПК-6.1: Знать методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий.

ОПК-6.2: Уметь применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий.

ОПК-6.3: Иметь навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач.

ОПК-7.1: Знать основные платформы, технологии и инструментальные программно-аппаратные средства для реализации информационных систем.

ОПК-7.2: Уметь осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем, применять современные технологии реализации информационных систем.

ОПК-7.3: Иметь навыки владения технологиями и инструментальными программно-аппаратными средствами для реализации информационных систем.

**профессиональных:**

ПК-01.1: Знает виды и методы предпроектного обследования объекта автоматизации, типы и особенности архитектур информационных систем

ПК-01.2: Умеет проводить анализ объекта автоматизации, осуществлять оценку и выбор архитектуры разрабатываемой информационной системы

ПК-01.3: Имеет навыки разработки технического задания на создание информационной системы или технологии, разработки информационной системы и макетов пользовательского интерфейса

ПК-03.1: Знает базовые приемы обработки информации, языки программирования, основные процедуры написания и отладки программ, угрозы безопасности информационных систем и способы их предотвращения, методы расчета экономической эффективности информационных систем и технологий

ПК-03.2: Умеет обоснованно выбирать средства языка программирования, необходимые для решения поставленных задач, выявлять угрозы безопасности, проводить расчет экономической эффективности информационных систем и технологий

ПК-03.3: Имеет навыки использования современных интегрированных сред разработки для создания программных продуктов, запуска процедуры резервного копирования, применения методов ценообразования для создаваемых информационных продуктов или услуг

**4. Структура и содержание дисциплины**

**4.1. Объем учебной дисциплины и виды учебной работы**

Вид учебной работы	Объем часов (з.е.)		
	Очная форма	Очно-заочная форма	Заочная форма
Объем учебной дисциплины (всего)	144 (4 з.е.)	-	144 (4 з.е.)
Обязательная аудиторная учебная нагрузка дисциплины (всего) в том числе:	70	-	14
Лекции	28	-	6
Семинарские занятия	-	-	-
Практические занятия	-	-	-
Лабораторные работы	42	-	8
Курсовая работа (курсовой проект)	-	-	-
Индивидуальное задание	-	-	-
Самостоятельная работа студента (всего)	38	-	121
Форма аттестации	экзамен	-	экзамен

## 4.2. Содержание разделов дисциплины

### Тема 1. Структура проекта

Что такое Package для приложения. Ресурсы приложения. Папка res\_values. Подписывание приложения. Утилиты keytool и jarsigner.

### Тема 2. Размещение элементов

Layout-файл в Activity. XML представление. Смена ориентации экрана. Виды Layouts. Ключевые отличия и свойства. Layout параметры для View-элементов.

### Тема 3. Связь с кодом

Работа с элементами экрана из кода. Обработчики событий на примере Button. Варианты реализации обработчиков

### Тема 4. Элементы интерфейса

Всплывающие сообщения. Простое меню. Группы, порядок. MenuInflater и xml-меню. Контекстное меню. Логи.

### Тема 5. Компонент Activity

Создание и вызов Activity. Intent, Intent Filter, Context. Activity Lifecycle, пример смены состояний с двумя Activity. Понятие Task. Формирование Task. Сохранение данных Activity при повороте экрана. Поведение.

### Тема 6. Activity и Intent

Intent Filter. Использование action из Intent. Передача данных с помощью Extras. Вызов Activity для получения результата. Метод startActivityForResult, onActivityResult. Параметры requestCode и resultCode.

### Тема 7. Взаимодействие между приложениями

Атрибут data Intent. Класс Uri. Вызов системных приложений. Расширение системных возможностей. Класс Parcel. Передача Parcelable объекты с помощью Intent.

### Тема 8. Хранение данных

Хранение данных. Preferences. Работа с файлами. XmlPullParser. Разбор XML документов.

### Тема 9. Диалоги

Встроенные диалоги TimePickerDialog и DatePickerDialog. Класс AlertDialog - Title, Message, Icon, Buttons. Кастомизация AlertDialog. Обработчики и операции. Управление из Activity. Класс ProgressDialog.

## Тема 10. Обработка касаний

Обработка одиночных касаний. Обработка множественных касаний

## Тема 11. Базовая графика

Анимация. Рисование. Доступ к Canvas. Простые фигуры, текст. Класс Picture. Графика. Drawable. Shape, Gradient, Bitmap, Layer List, State List.

## Тема 12. Мультиобработка и синхронизация

Класс Handler. Отложенные сообщения, удаление из очереди, Handler.Callback. Обработка Runnable. Обращение к UI-потoku. Класс AsyncTask. Параметры. Промежуточные результаты. Итоговый результат. Метод get. Отмена задачи в процессе выполнения. Статусы задачи. Поворот экрана.

## Тема 13. Компонент Service

Передача данных в сервис. Методы остановки сервиса. Метод onStartCommand. Обратная связь с помощью PendingIntent. Обратная связь с помощью BroadcastReceiver. Биндинг. ServiceConnection. Локальный биндинг.

## Тема 14. Механизм уведомлений

Уведомления. Основы. Режимы открытия Activity. Расширенные уведомления. Кастомные уведомления. Группировка уведомлений. Каналы.

### 4.3. Лекции

№ п/п	Название темы	Объем часов		
		Очная форма	Очно-заочная форма	Заочная форма
1	Структура проекта	2	-	0,5
2	Размещение элементов	2	-	0,5
3	Связь с кодом	2	-	0,5
4	Элементы интерфейса	2	-	0,5
5	Компонент Activity	2	-	0,5
6	Activity и Intent	2	-	0,5
7	Взаимодействие между приложениями	2	-	0,5
8	Хранение данных	2	-	0,5
9	Диалоги	2	-	0,5
10	Обработка касаний	2	-	0,5
11	Базовая графика	2	-	-
12	Мультиобработка и синхронизация	2	-	0,5
13	Компонент Service	2	-	0,5
14	Механизм уведомлений	2	-	-
Итого:		28	-	6

#### 4.4. Практические (семинарские) занятия

Не предусмотрены

#### 4.5. Лабораторные работы

№ п/п	Название темы	Объем часов		
		Очная форма	Очно-заочная форма	Заочная форма
1	Простая верстка	2	-	0,5
2	Программное обновление элементов отображения	2	-	0,5
3	Адаптация под устройство	2	-	0,5
4	Многоязыковая поддержка	2	-	0,5
5	Обработчики событий	2	-	0,5
6	Отслеживание состояний Activity	2	-	0,5
7	Меню и toast-уведомлениями	2	-	0,5
8	Сохранение настроек	2	-	0,5
9	Простая анимация интерфейса пользователя	2	-	0,5
10	Чтение системных файлов	2	-	0,5
11	Работа с файлами и каталогами	4	-	0,5
12	Обращение к компонентам приложения	4	-	0,5
13	Анимация переключения Activity	4	-	0,5
14	Обращение к системным компонентам	2	-	0,5
15	Обновление интерфейса из фонового потока	4	-	0,5
16	Организация фоновых вычислений	4	-	0,5
Итого:		42	-	8

#### 4.6. Самостоятельная работа студентов

№ п/п	Название темы	Вид СРС	Объем часов		
			Очная форма	Очно-заочная форма	Заочная форма
1	Структура проекта	подготовка к лабораторным занятиям, оформление отчетов	2	-	6
2	Размещение элементов	подготовка к лабораторным занятиям, оформление отчетов	2	-	6
3	Связь с кодом	подготовка к лабораторным занятиям, оформление отчетов	2	-	8
4	Элементы интерфейса	подготовка к лабораторным занятиям, оформление отчетов	2	-	8
5	Компонент Activity	подготовка к лабораторным занятиям, оформление отчетов	2	-	8
6	Activity и Intent	подготовка к лабораторным занятиям, оформление отчетов	2	-	8
7	Взаимодействие	подготовка к	2	-	8

№ п/п	Название темы	Вид СРС	Объем часов		
			Очная форма	Очно-заочная форма	Заочная форма
	между приложениями	лабораторным занятиям, оформление отчетов			
8	Хранение данных	подготовка к лабораторным занятиям, оформление отчетов	4	-	12
9	Диалоги	подготовка к лабораторным занятиям, оформление отчетов	4	-	12
10	Обработка касаний	подготовка к лабораторным занятиям, оформление отчетов	2	-	10
11	Базовая графика	подготовка к лабораторным занятиям, оформление отчетов	4	-	12
12	Мультиобработка и синхронизация	подготовка к лабораторным занятиям, оформление отчетов	2	-	5
13	Компонент Service	подготовка к лабораторным занятиям, оформление отчетов	4	-	10
14	Механизм уведомлений	подготовка к лабораторным занятиям, оформление отчетов	4	-	8
Итого:			38	-	121

#### **4.7. Курсовые работы/проекты**

Не предусмотрены

### **5. Образовательные технологии**

Преподавание дисциплины ведется с применением следующих видов образовательных технологий:

- традиционные объяснительно-иллюстративные технологии, которые обеспечивают доступность учебного материала для большинства студентов, системность, отработанность организационных форм и привычных методов, относительно малые затраты времени;
- технологии проблемного обучения, направленные на развитие познавательной активности, творческой самостоятельности студентов и предполагающие последовательное и целенаправленное выдвижение перед студентом познавательных задач, разрешение которых позволяет студентам активно усваивать знания (используются поисковые методы; постановка познавательных задач);

– технологии развивающего обучения, позволяющие ориентировать учебный процесс на потенциальные возможности студентов, их реализацию и развитие;

– технологии концентрированного обучения, суть которых состоит в создании максимально близкой к естественным психологическим особенностям человеческого восприятия структуры учебного процесса и которые дают возможность глубокого и системного изучения содержания учебных дисциплин за счет объединения занятий в тематические блоки;

– технологии модульного обучения, дающие возможность обеспечения гибкости процесса обучения, адаптации его к индивидуальным потребностям и особенностям обучающихся (применяются, как правило, при самостоятельном обучении студентов по индивидуальному учебному плану);

– технологии дифференцированного обучения, обеспечивающие возможность создания оптимальных условий для развития интересов и способностей студентов, в том числе и студентов с особыми образовательными потребностями, что позволяет реализовать в культурно-образовательном пространстве университета идею создания равных возможностей для получения образования

– технологии активного (контекстного) обучения, с помощью которых осуществляется моделирование предметного, проблемного и социального содержания будущей профессиональной деятельности студентов (используются активные и интерактивные методы обучения) и т.д.

Максимальная эффективность педагогического процесса достигается путем конструирования оптимального комплекса педагогических технологий и (или) их элементов на личностно-ориентированной, деятельностной, диалогической основе и использования необходимых современных средств обучения.

## **6. Формы контроля освоения дисциплины**

Текущая аттестация студентов производится в дискретные временные интервалы лектором и преподавателем, ведущим практические занятия в следующих формах:

- лабораторные работы;
- защита результатов практических занятий (устная форма).

Фонды оценочных средств, включающие типовые задания и методы контроля, позволяющие оценить результаты промежуточной аттестации обучающихся по данной дисциплине, помещаются в приложении к рабочей программе в соответствии с «Положением о фонде оценочных средств».

Промежуточный контроль по результатам освоения дисциплины проходит в форме экзамена, который включает в себя ответ на теоретические вопросы.

Студенты, выполнившие 75% текущих и контрольных мероприятий на «отлично», а остальные 25 % на «хорошо», имеют право на получение итоговой отличной оценки.

В экзаменационную ведомость и зачетную книжку выставляются оценки по шкале, приведенной в таблице.

Шкала оценивания	Характеристика знания предмета и ответов
отлично (5)	Студент глубоко и в полном объеме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.
хорошо (4)	Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.
удовлетворительно (3)	Студент знает только основной программный материал, допускает неточности, недостаточно четкие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.
неудовлетворительно (2)	Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы.

## **7. Учебно-методическое и программно-информационное обеспечение дисциплины:**

### **а) основная литература**

1. Семакова А., Введение в разработку приложений для смартфонов на ОС Android / Семакова А. - М.: Национальный Открытый Университет "ИНТУИТ", 2016. - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : [http://www.studentlibrary.ru/book/intuit\\_083.html](http://www.studentlibrary.ru/book/intuit_083.html) (дата обращения: 18.03.2023). - Режим доступа : по подписке.
2. Березовская Ю.В., Введение в разработку приложений для ОС Android / Березовская Ю.В., Юфрякова О.А., Володина В.Г., Озерова О.В., Куликов Э.Е., Латухина Е.А., Пархимович М.Н. - М.: Национальный Открытый Университет "ИНТУИТ", 2016. - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL :

- [http://www.studentlibrary.ru/book/intuit\\_082.html](http://www.studentlibrary.ru/book/intuit_082.html) (дата обращения: 18.03.2023). - Режим доступа : по подписке.
3. Ёранссон А., Эффективное использование потоков в операционной системе Android / Ёранссон А. - М. : ДМК Пресс, 2015. - 304 с. - ISBN 978-5-97060-168-6 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785970601686.html> (дата обращения: 18.03.2023). - Режим доступа : по подписке.

#### **б) дополнительная литература**

1. Медникс З., Дорнин Л., Мик Б., Накамура М. Программирование под Android. 2-е изд. — СПб.: Питер, 2013. — 560 с.: ил. — (Серия «Бестселлеры О’Reilly»). — ISBN 978-5-496-00526-5. - Текст : электронный [сайт]. - URL : [https://www.studmed.ru/medniks-z-dornin-l-mik-b-nakamura-m-programmirovanie-pod-android\\_076930dba9a.html](https://www.studmed.ru/medniks-z-dornin-l-mik-b-nakamura-m-programmirovanie-pod-android_076930dba9a.html) (дата обращения: 18.03.2023). - Режим доступа : свободный
2. Хашими С., Коматинени С., Маклин Д. Разработка приложений для Android. СПб.: Питер, 2011. — 736 с: ил. ISBN 978-5-459-00530-1 - Текст : электронный [сайт]. - URL : [https://www.studmed.ru/hashimi-s-komatineni-s-maklin-d-razrabotka-prilozheniy-dlya-android\\_6226f74c5c3.html](https://www.studmed.ru/hashimi-s-komatineni-s-maklin-d-razrabotka-prilozheniy-dlya-android_6226f74c5c3.html) (дата обращения: 18.03.2023). - Режим доступа : свободный
3. Дейтел П., Дейтел Х., Уолд А. Android для разработчиков. 3-е издание. — СПб.: Питер, 2016. — 512 с.: ил. — (Библиотека программиста). — ISBN 978-5-496-02371-9. - Текст : электронный [сайт]. - URL : [https://www.studmed.ru/deytel-p-deytel-h-uold-a-android-dlya-razrabotchikov\\_22d45659c69.html](https://www.studmed.ru/deytel-p-deytel-h-uold-a-android-dlya-razrabotchikov_22d45659c69.html) (дата обращения: 18.03.2023). - Режим доступа : свободный
4. Нортон П. Руководство Питера Нортон. Программирование на Java. В 2 кн. Кн. 1 [Текст] / П. Нортон, У. Станек. - М. : СК Пресс, 1998. - 552 с. : ил. - ISBN 1-57521-088-6 (США). - ISBN 5-89233-012-5 (Россия)
5. Нортон П. Руководство Питера Нортон. Программирование на Java. В 2 кн. Кн. 2 [Текст] / П. Нортон, У. Станек. - М. : СК Пресс, 1998. - 400 с. : ил. - ISBN 1-57521-088-6 (США). - ISBN 5-89233-012-5 (Россия)

#### **в) методические указания**

1. Конспект лекций по дисциплине «Разработка приложений для мобильных устройств» (электронное издание) / Составители: Д.О. Синепольский, В.В. Макогон – Луганск: изд-во ЛГУ им. В. Даля, 2023. – 90 с.
2. Методические указания к лабораторным работам по дисциплине «Разработка приложений для мобильных устройств» (электронное издание) / Составители: Д.О. Синепольский, В.В. Макогон – Луганск: изд-во ЛГУ им. В. Даля, 2023. – 44 с.

### г) Интернет-ресурсы:

#### Электронные библиотечные системы и ресурсы

Электронно-библиотечная система «Консультант студента» – <http://www.studentlibrary.ru/cgi-bin/mb4x>

**Информационный ресурс библиотеки образовательной организации**

Научная библиотека имени А. Н. Коняева – <http://biblio.dahluniver.ru/>

### 8. Материально-техническое обеспечение дисциплины

Освоение дисциплины предполагает использование академических аудиторий, соответствующих действующим санитарным и противопожарным правилам и нормам.

Прочее: рабочее место преподавателя, оснащенное компьютером с доступом в Интернет.

Программное обеспечение:

Функциональное назначение	Бесплатное программное обеспечение	Ссылки
Операционная система	UBUNTU 19.04	<a href="https://ubuntu.com/">https://ubuntu.com/</a> <a href="https://ru.wikipedia.org/wiki/Ubuntu">https://ru.wikipedia.org/wiki/Ubuntu</a>
Интегрированная среда разработки на языке Java	Eclipse MARS	<a href="https://www.eclipse.org/downloads/packages/release/mars/2">https://www.eclipse.org/downloads/packages/release/mars/2</a>
Android SDK	Android SDK 22	<a href="https://developer.android.com/studio/index.html">https://developer.android.com/studio/index.html</a>
Плагин для интеграции SDK и среды разработки	Android Development Tools 15	<a href="https://dl.google.com/android/ADT-15.0.0.zip">https://dl.google.com/android/ADT-15.0.0.zip</a>
Офисный пакет	Libre Office 6.3.1	<a href="https://www.libreoffice.org/">https://www.libreoffice.org/</a> <a href="https://ru.wikipedia.org/wiki/LibreOffice">https://ru.wikipedia.org/wiki/LibreOffice</a>
Отображение файлов в формате PDF	Foxit Reader	<a href="https://www.foxitsoftware.com/downloads/">https://www.foxitsoftware.com/downloads/</a> (Free Download)
Браузер	Firefox Mozilla	<a href="http://www.mozilla.org/ru/firefox/fx">http://www.mozilla.org/ru/firefox/fx</a>

**Паспорт  
фонда оценочных средств по учебной дисциплине  
«Разработка приложений для мобильных устройств»**

**Перечень компетенций (элементов компетенций),  
формируемых в результате освоения учебной дисциплины**

№ п/п	Код контролируемой компетенции	Формулировка контролируемой компетенции	Контролируемые темы учебной дисциплины	Этапы формирования (семестр изучения)
1	ОПК-2.1	Знать: принципы работы современных принципы работы информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности	Тема 1. Структура проекта Тема 13. Компонент Service Тема 14. Механизм уведомлений	начальный (8)
2	ОПК-2.2	Уметь: выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности	Тема 7. Взаимодействие между приложениями Тема 8. Хранение данных	начальный (8)
3	ОПК-2.3	Иметь навыки: применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности	Тема 5. Компонент Activity Тема 6. Activity и Intent	начальный (8)
4	ОПК-6.1	Знать: методы алгоритмизации, языки и технологии программирования, пригодные для	Тема 5. Компонент Activity Тема 6. Activity и Intent Тема 9. Диалоги	начальный (8)

№ п/п	Код контролируемой компетенции	Формулировка контролируемой компетенции	Контролируемые темы учебной дисциплины	Этапы формирования (семестр изучения)
		практического применения в области информационных систем и технологий.	Тема 12. Мультиобработка и синхронизация	
5	ОПК-6.2	Уметь: применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий.	Тема 5. Компонент Activity Тема 6. Activity и Intent Тема 9. Диалоги Тема 12. Мультиобработка и синхронизация	начальный (8)
6	ОПК-6.3	Иметь навыки: программирования, отладки и тестирования прототипов программно-технических комплексов задач.	Тема 12. Мультиобработка и синхронизация	начальный (8)
7	ОПК-7.1	Знать: основные платформы, технологии и инструментальные программно-аппаратные средства для реализации информационных систем.	Тема 3. Связь с кодом Тема 4. Элементы интерфейса Тема 10. Обработка касаний	начальный (8)
8	ОПК-7.2	Уметь: осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем, применять современные технологии реализации информационных систем.	Тема 3. Связь с кодом Тема 4. Элементы интерфейса Тема 10. Обработка касаний	начальный (8)

№ п/п	Код контролируемой компетенции	Формулировка контролируемой компетенции	Контролируемые темы учебной дисциплины	Этапы формирования (семестр изучения)
10	ОПК-7.3	Иметь навыки: владения технологиями и инструментальными программно-аппаратными средствами для реализации информационных систем.	Тема 11. Базовая графика Тема 12. Мультиобработка и синхронизация	начальный (8)
11	ПК-01.1	Знает виды и методы предпроектного обследования объекта автоматизации, типы и особенности архитектур информационных систем	Тема 1. Структура проекта	начальный (8)
12	ПК-01.2	Умеет проводить анализ объекта автоматизации, осуществлять оценку и выбор архитектуры разрабатываемой информационной системы	Тема 1. Структура проекта	начальный (8)
13	ПК-01.3	Имеет навыки разработки технического задания на создание информационной системы или технологии, разработки информационной системы и макетов пользовательского интерфейса	Тема 2. Размещение элементов Тема 3. Связь с кодом Тема 4. Элементы интерфейса	начальный (8)
14	ПК-03.1	Знает базовые приемы обработки информации, языки программирования, основные процедуры написания и отладки программ, угрозы безопасности информационных	Тема 1. Структура проекта Тема 7. Взаимодействие между приложениями Тема 8. Хранение данных	начальный (8)

№ п/п	Код контролируемой компетенции	Формулировка контролируемой компетенции	Контролируемые темы учебной дисциплины	Этапы формирования (семестр изучения)
		систем и способы их предотвращения, методы расчета экономической эффективности информационных систем и технологий		
15	ПК-03.2	Умеет обоснованно выбирать средства языка программирования, необходимые для решения поставленных задач, выявлять угрозы безопасности, проводить расчет экономической эффективности информационных систем и технологий	Тема 1. Структура проекта Тема 11. Базовая графика	начальный (8)
16	ПК-03.3	Имеет навыки использования современных интегрированных сред разработки для создания программных продуктов, запуска процедуры резервного копирования, применения методов ценообразования для создаваемых информационных продуктов или услуг	Тема 1. Структура проекта	начальный (8)

**Показатели и критерии оценивания компетенций,  
описание шкал оценивания**

№ п/п	Код контролируемой компетенции	Показатель оценивания (знания, умения, навыки)	Контролируемые темы учебной дисциплины	Наименование оценочного средства
1	ОПК-2.1	Знать: - архитектуру Android процесса - знать назначение компонента Service - назначение механизма уведомлений	Тема 1. Тема 13. Тема 14.	Вопросы лабораторной работе Контрольные работы Вопросы экзамену
2	ОПК-2.2	Знать: - механизмы сериализации сложных данных для передачи их между компонентами - особенности файловой системы и операций в Android - механизмы чтения XML документов Уметь: - программировать работу с файлами	Тема 7. Тема 8.	Вопросы лабораторной работе Контрольные работы Вопросы экзамену
3	ОПК-2.3	Уметь: - программировать поведение Activity в Task - программировать активацию компонента на основе механизма Uri Владеть: - навыками формирования и запросов с использованием Uri	Тема 5. Тема 6.	Вопросы лабораторной работе Контрольные работы Вопросы экзамену
4	ОПК-6.1	Знать: - назначение компонента Activity, его жизненный цикл - механизмы связи между компонентами приложений - механизмы сохранения состояния Activity - назначение механизмы Intent	Тема 5. Тема 6. Тема 9. Тема 12.	Вопросы лабораторной работе Контрольные работы Вопросы экзамену
5	ОПК-6.2	Уметь:	Тема 5.	Вопросы

		<ul style="list-style-type: none"> <li>- использовать стандартные диалоги Android</li> <li>- программировать фоновую обработку</li> <li>- запускать сервис, передавать задачу для фоновой обработки</li> <li>- программировать связывание с запущенным сервисом</li> <li>- программировать обратную связь с использованием PendingIntent и BroadcastReceiver</li> <li>- программировать уведомления в Activity и Service</li> <li>- программировать кастомные уведомления и организовывать обратную связь</li> </ul>	<p>Тема 6. Тема 9. Тема 12.</p>	<p>лабораторной работе Контрольные работы Вопросы к экзамену</p>
6	ОПК-6.3	<p>Владеть:</p> <ul style="list-style-type: none"> <li>- навыками работы с классами Handler и AsyncTask</li> </ul>	<p>Тема 12.</p>	<p>Вопросы к лабораторной работе Контрольные работы Вопросы к экзамену</p>
7	ОПК-7.1	<p>Знать:</p> <ul style="list-style-type: none"> <li>- архитектуру Android процесса</li> <li>- различные варианты организации обработчиков</li> <li>- механизм касаний, реализованный в Android</li> </ul> <p>Владеть:</p> <ul style="list-style-type: none"> <li>- навыками программирования распознавания жестов</li> </ul>	<p>Тема 3. Тема 4. Тема 10.</p>	<p>Вопросы к лабораторной работе Контрольные работы Вопросы к экзамену</p>
8	ОПК-7.2	<p>Уметь:</p> <ul style="list-style-type: none"> <li>- работать с визуальными элементами из кода</li> <li>- назначать обработчики событий в визуальных элементах</li> <li>- уметь программировать меню и всплывающие</li> </ul>	<p>Тема 3. Тема 4. Тема 10.</p>	<p>Вопросы к лабораторной работе Контрольные работы Вопросы к экзамену</p>

		сообщения - обрабатывать одиночные и множественные касания		
9	ОПК-7.3	Владеть: - работать с объектом Canvas из основного потока приложения - программировать вывод графических примитивов и текста - работать с компонентом Surface из фонового потока	Тема 11. Тема 12.	Вопросы лабораторной работе Контрольные работы Вопросы экзамену к
10	ПК-01.1	Знать: - архитектуру Android системы и Android приложения - основные компоненты и правила их упаковки - механизмы безопасности, связанные с загрузкой и отладкой Android приложений	Тема 1.	Вопросы лабораторной работе Контрольные работы Вопросы экзамену к
11	ПК-01.2	Знать: - архитектуру Android системы и Android приложения - основные компоненты и правила их упаковки - механизмы безопасности, связанные с загрузкой и отладкой Android приложений	Тема 1.	Вопросы лабораторной работе Контрольные работы Вопросы экзамену к
12	ПК-01.3	Знать: - архитектуру визуальных компонентов Android - номенклатуру интерактивных элементов и механизмов их размещения - методы декларативного программирования графики и анимации в Android - механизмы связи между фоновыми и UI	Тема 2. Тема 3. Тема 4.	Вопросы лабораторной работе Контрольные работы Вопросы экзамену к

		<p>потоком</p> <p>Уметь:</p> <ul style="list-style-type: none"> <li>- навыками верстки интерфейса пользователя с применением различных типов механизмов размещения</li> <li>- навыками верстки с поддержкой различных размеров, ориентаций экранов и пользовательских языков</li> </ul> <p>Владеть:</p> <ul style="list-style-type: none"> <li>- навыками программирования пользовательских диалогов</li> </ul>		
13	ПК-03.1	<p>Знать:</p> <ul style="list-style-type: none"> <li>- основные компоненты и правила их упаковки</li> <li>- знать методы хранения настроек уровня приложения</li> <li>- механизмы безопасности, связанные с загрузкой и отладкой Android приложений</li> </ul> <p>Владеть:</p> <ul style="list-style-type: none"> <li>- навыками программирования передачи данных и получения результата из Activity</li> </ul>	Тема 1. Тема 7. Тема 8.	<p>Вопросы лабораторной работе</p> <p>Контрольные работы</p> <p>Вопросы экзамену</p> <p>к</p> <p>к</p>
14	ПК-03.2	<p>Знать:</p> <ul style="list-style-type: none"> <li>- методы декларативного программирования графики и анимации в Android</li> <li>- механизмы безопасности, связанные с загрузкой и отладкой Android приложений</li> </ul> <p>Уметь:</p> <ul style="list-style-type: none"> <li>- управлять ключами разработчика</li> </ul> <p>Владеть:</p> <ul style="list-style-type: none"> <li>- навыками авторизации и удаленной отладки на Android устройствах</li> <li>- навыками конфигурации Android</li> </ul>	Тема 1. Тема 11.	<p>Вопросы лабораторной работе</p> <p>Контрольные работы</p> <p>Вопросы экзамену</p> <p>к</p> <p>к</p>

		устройств для разработки приложений		
15	ПК-03.3	<p>Знать:</p> <ul style="list-style-type: none"> <li>- архитектуру Android системы и Android приложения</li> <li>- основные компоненты и правила их упаковки</li> <li>- механизмы безопасности, связанные с загрузкой и отладкой Android приложений</li> </ul> <p>Уметь:</p> <ul style="list-style-type: none"> <li>- управлять ключами разработчика</li> </ul> <p>Владеть:</p> <ul style="list-style-type: none"> <li>- навыками авторизации и удаленной отладки на Android устройствах</li> <li>- навыками конфигурации Android устройств для разработки приложений</li> <li>- навыками загрузки и обновления приложений на Android устройствах</li> </ul>	Тема 1.	<p>Вопросы к лабораторной работе</p> <p>Контрольные работы</p> <p>Вопросы к экзамену</p>

**Фонды оценочных средств по дисциплине  
«Разработка приложений для мобильных устройств»**

**Вопросы для защиты лабораторных работ**

Лабораторная работа № 1.

Простая верстка

Цель работы: Познакомиться с процессом верстки пользовательского интерфейса на основе layout-ов; процессом подготовки приложения от исходного кода до запуска на мобильном устройстве.

Вопросы для защиты:

- 1) Какие виды layout-ов вы знаете? Как определить ориентацию компонент в LinearLayout?
- 2) Что означает атрибут android:weight для Linear... и TableLayout?
- 3) Сколько компонентов можно разместить внутри ScrollView? Какой компонент там обычно размещается?
- 4) Нужно ли использовать ScrollView вместе с TextView? Почему?
- 5) Как поместить Drawable слева от текста? Нужен ли отдельный layout?
- 6) Какие теги псевдо-HTML допускаются внутри текста TextView?

- 7) В каком файле проекта указывается идентификатор приложения? С чем он должен совпадать?
- 8) В каком файле задаются темы? В каком файле темы привязываются к компонентам приложения?
- 9) В какой папке проекта следует располагать картинки, не привязанные к характеристикам устройства?
- 10) В каком файле задаются ресурсы строк? Какой синтаксис используется?
- 11) Как сослаться на строку, заданную в ресурсах, из других xml файлов проекта?

### Лабораторная работа № 2.

#### Программное обновление элементов отображения

Цель работы: Познакомиться с механизмами доступа к элементам загруженного отображения и изменения их свойств.

Вопросы для защиты:

- 1) Что такое класс R приложения? Как и когда он создается?
- 2) Что такое класс android.R? Когда он используется? Почему импорт этого к класса «плохая идея»?
- 3) Как задать идентификатор элементу вида? Всегда ли нужен идентификатор?
- 4) Как получить ссылку на элемент вида, если известен его идентификатор?
- 5) Какой метод позволяет загрузить вид из его xml описания? В какой момент он обычно вызывается?
- 6) Какой метода TextView позволяет изменить его содержимое?
- 7) Какие методы получения текущего времени доступны из Java приложения?
- 8) Какой класс Java предназначен для манипуляций с датами? Как получить экземпляр этого класса для текущего момента?

### Лабораторная работа № 3.

#### Адаптация под устройство

Цель работы: Познакомится с процессом создания версий ресурсов с поддержкой различных ориентаций и разрешений экрана.

Вопросы для защиты:

- 1) Каким образом указывается, что данная версия ресурсов предназначена для устройств с определенным разрешением дисплея? Что такое `-mdp` и `-ldp` суффиксы?
- 2) Какие ориентации дисплея устройств поддерживаются Android? Под какую ориентацию идет верстка по-умолчанию?
- 3) Каким образом указываются `layout-ы` для альбомной ориентация дисплея устройства?
- 4) Что происходит при повороте дисплея? Какие методы Activity вызываются?

5) Каким образом задать иконку приложения? Что нужно учитывать в случае растровой иконки?

Лабораторная работа № 4.  
Многоязыковая поддержка

Цель работы: Познакомится с процессом создания версий ресурсов с поддержкой различных системных языков.

Вопросы для защиты:

- 1) Каким образом рекомендуется оформлять культурно-зависимые строки? Как использовать такую строку в контексте, где требуется Java тип String?
- 2) Как сослаться на строку, определенную в ресурсе из файла AndroidManifest.xml? Все ли строки из AndroidManifest.xml можно вынести в ресурсы?
- 3) В каком файле по умолчанию сохраняются строковые ресурсы? Как задать версию файла для определенного языка?
- 4) Можно ли сделать специальные версии Drawable ресурсов, зависящих от языка? Можно ли при этом оставить зависимость от разрешения дисплея?
- 5) Какой приоритет у квалификатора локализации ресурсов по отношению к другим квалификаторам, таким как ориентация или размер экрана?

## Лабораторная работа № 5.

### Обработчики событий

Цель работы: Научиться задавать обработку событий в элементах вида.

Вопросы для защиты:

- 1) Назовите два главных способа закрепить обработчик события `onClick` за компонентом.
- 2) В каком классе должен находиться и как быть описан обработчик события, если в `xml layout`-е использовался атрибут `android:onClick`? Если это условие нарушено, в какой момент будет обнаружена ошибка?
- 3) Требуется ли кнопке назначать атрибут `android:id`, если обработчик события задается атрибутом `android:onClick`?
- 4) Можно ли использовать одинаковое значение атрибута `android:onClick` с несколькими компонентами? Как в этом случае различать компоненты? Нужно ли задавать компонентам атрибут `android:id`?
- 5) Если вы решили имплементировать интерфейс `View.OnClickListener` на уровне класса `Activity`, то каким образом отличать события, произошедшие в разных компонентах?
- 6) Какие преимущества можно получить, реализуя интерфейс `View.OnClickListener` в локальных классах, по сравнению с другими технологиями? Какой недостаток у данного метода?
- 7) Если требуется разместить элемент вида определенным образом по отношению к родительскому или соседним элементам, то какой тип (типы) `layout`-а следует предпочесть?

## Лабораторная работа № 6.

### Отслеживание состояний Activity

Цель работы: Научиться обрабатывать изменение состояния `Activity`, работать с классом `Bundle`.

Вопросы для защиты:

- 1) При каких обстоятельствах система пересоздает `Activity` путем его уничтожения и создания заново? Как при этом сохраняется состояние `Activity`?
- 2) Методы `onCreate()` и `onRestoreInstanceState()` имеют параметр `Bundle`. В каком случае этот параметр может быть `null`?
- 3) Нужно ли вызывать `super.onSaveInstanceState()` и `super.onRestoreInstanceState()`? Зачем?
- 4) Какую часть состояния `Activity` система сохраняет автоматически? Какое обязательное условие для сохранения состояния элемента вида?
- 5) Если у элемента `TextView` будет установлен атрибут `android:id`, будет ли его состояние автоматически сохраняться системой при изменении конфигурации? Почему?
- 6) Если состояние храниться в полях класса `Activity`, а отображается через элементы вида, какое действие нужно предпринять в обработчике `onRestoreInstanceState()`?

- 7) Какие типы данных могут быть сохранены в Bundle в `onSaveInstanceState()`?
- 8) Какие требования предъявляются к классу, чтобы его экземпляр мог быть сохранен в Bundle?

### Лабораторная работа №7.

#### Меню и toast-уведомлениями

Цель работы: Научиться создавать меню, управлять состоянием меню с помощью групп, использовать toast-уведомлениями.

Вопросы для защиты:

- 1) Как определить, был ли метод `onCreate()` вызван в результате изменения конфигурации устройства?
- 2) Для чего предназначен класса `GestureDetector`? Написание какого обработчика он упрощает?
- 3) Что такое длинное нажатие? Как задать его обработчик?
- 4) Что такое контекстное меню? Как задать обработчик его создания? Что делает `Activity.registerForContextMenu()`?
- 5) Как программно показать контекстное меню для элемента вида? Что делает `Activity.openContextMenu()`?
- 6) Какая связь между обработчиком длинного нажатия и контекстным меню? Что будет если обработчик вернет `true`?
- 7) Для чего предназначены группы меню? Как группы меню отражаются в xml файле?
- 8) Какой класс предназначен для загрузки меню из xml ресурсов? Как получить его экземпляр?
- 9) Для чего предназначены методы `onSaveInstanceState()` и `onRestoreInstanceState()`?
- 10) Что такое toast-уведомление? Как показать простое текстовое сообщение?

### Лабораторная работа №8.

#### Сохранение настроек

Цель работы: Научиться сохранять настройки приложения с применением `SharedPreferences`.

Вопросы для защиты:

- 1) Что такое `SharedPreferences`? Где сохраняются данные? В каком формате?
- 2) В чем разница между `Activity.getPreferences()` и `Activity.getSharedPreferences()`?
- 3) Что задает параметр `mode` метода `get*Preferences()`?
- 4) Как прочитать данные из `SharedPreferences`? Как получить все данные, не зная ключей?
- 5) Для чего предназначен класс `SharedPreferences.Editor`? Как получить его экземпляр?
- 6) В чем разница между методами `Editor.apply()` и `Editor.commit()`?

7) Что делает метод `Editor.clear()`?

### Лабораторная работа №9.

#### Простая анимация интерфейса пользователя

Цель работы: Познакомится с применением анимации в пользовательском интерфейсе на уровне отдельных элементов и при переключении между Activity.

Вопросы для защиты:

- 1) Что обеспечивает класс `ViewAnimator`? Какие его наследники вы знаете?
- 2) Какую функциональность обеспечивает `ViewSwitcher`? В чем его отличие от `ViewFlipper`?
- 3) Сколько видов можно поместить внутри `ViewFlipper`?
- 4) Какую функциональность обеспечивает `ImageSwitcher`?
- 5) Что задают атрибуты `android:inAnimation` и `android:outAnimation`?
- 6) В каком каталоге ресурсов должны располагаться описания анимации?
- 7) Как в файле `layout` сослаться на анимацию определенную в Android, а не в проекте приложения?
- 8) Как переключиться на заданный вид `ViewFlipper`? Как задать автоматическую прокрутку через временной интервал?

### Лабораторная работа №10.

#### Чтение системных файлов

Цель работы: Научиться получать информацию от операционной системы Linux через виртуальную файловую систему.

Вопросы для защиты:

- 1) Что такое файловая система `proc`. К какому уровню Android системы она относится? Где искать описание ее компонент?
- 2) Что такое каталоги вида `/proc/id`, где `id` – десятичное значение? Что такое каталог `/proc/self`?
- 3) Какие классы Java можно использовать для доступа к элементам файловой системы?
- 4) Какие классы Java можно использовать для доступа к содержимому файлов?
- 5) Как программно разрешить прокрутку текста по горизонтали в `TextView`?
- 6) Что обеспечивает установка `movement method` для `TextView`?
- 7) Как используя функции файловой системы узнать идентификатор процесса, в котором выполняется Android приложение?
- 8) Что содержится в файле `/proc/*/cmdline` Android процесса?

### Лабораторная работа №11.

#### Работа с файлами и каталогами

Цель работы: Научиться работать с файловой системой в Android приложении.

Вопросы для защиты:

- 1) Какие Java классы позволяют получить содержимое каталога?
- 2) Что такое символические ссылки в Linux? Как получить полное имя файла, зная имя ссылки?
- 3) Как узнать, ссылается ли строка на каталог?
- 4) Что означает `canWrite()` для каталога? Кто может изменить содержимое файла?
- 5) Как установить разрешение на чтение файла всеми, независимо от владельца?

### Лабораторная работа №12.

#### Обращение к компонентам приложения

Цель работы: Познакомится с применением механизма Intent для обращения к компонентам, реализованным внутри приложения.

Вопросы для защиты:

- 1) Что такое Intent? Как явно указать имя класса Activity, которую требуется запустить?
- 2) Какие типы данных можно передать через Intent? Что такое Extras?
- 3) Какие требования предъявляются к классу, чтобы его экземпляр можно было передать через Intent? Какие дополнительные ограничения возникают, если исходная и целевая Activity находятся в разных приложениях?
- 4) Какой метод используется для запуска Activity через Intent?
- 5) Как получить Intent, с помощью которого была запущена данная Activity?
- 6) Как узнать, какой из пунктов одиночного выбора активен сейчас?
- 7) Как поместить данные в Intent Extras?
- 8) В чем отличие `getExtras().get*()` от `getExtra*()`?

### Лабораторная работа №13.

#### Анимация переключения Activity

Цель работы: Научиться программировать tween-анимацию, задавать эффекты переключения Activity.

Вопросы для защиты:

- 1) В каком каталоге должны располагаться ресурсы tween-анимации?
- 2) Как задать в анимации перемещение по горизонтали? В каких единицах задаются точки?
- 3) Какие модификации объекта, помимо перемещения, могут быть заданы в tween-анимации?
- 4) Что делает `Activity.overridePendingTransition()`? В какой момент должен вызываться этот метод?

- 5) Когда вызывается метод `finish()`? Как он связан с сенсорной кнопкой `Back` устройства?
- 6) Почему метод `overridePendingTransition()` принимает ссылки на два ресурса анимации?

#### Лабораторная работа №14.

##### Обращение к системным компонентам

Цель работы: Познакомится с применением механизма `Intent` для обращения к компонентам системы или других приложений.

Вопросы для защиты:

- 1) Что задает атрибут `action` класса `Intent`?
- 2) В чем отличие атрибутов `data` и `extras` класса `Intent`?
- 3) Как соотносятся между собой атрибуты `data` и `type` класса `Intent`? Можно ли задать оба атрибута одновременно?
- 4) Какой стандартный ACTION используется когда нужно передать данные за пределы приложения? В каком классе содержится эта константа?
- 5) Что будет, если обнаружится несколько приложений, способных обработать указанный `Intent`?
- 6) Какой `Intent` позволяет пользователю явно выбрать приложение, используемое для обработки данных?
- 7) Как указать, что данные требуется передать по электронной почте? В чем особенность задания адреса получателя?
- 8) Как задать заголовок и содержание письма для отправки по электронной почте?

#### Лабораторная работа №15.

##### Обновление интерфейса из фонового потока

Цель работы: Научиться программировать взаимодействие основного и фоновых потоков приложения.

Вопросы для защиты:

- 1) Как приложение должно вести себя в UI потоке?
- 2) Допускается ли в Android обращения к UI элементам из не-UI потоков?
- 3) Для чего предназначен класс `Handler`?
- 4) Что обеспечивает метод `Activity.runOnUiThread()`? Как он реализован?
- 5) Какая проблема может возникнуть с `Handler.postDelayed()`?
- 6) Как запустить новый поток? Какая связь между `Thread` и `Runnable`?
- 7) Как сохранить ссылку на объект, когда `Activity` пересоздается при изменении конфигурации? Как получить ссылку назад?
- 8) Почему метод `Thread.stop()` является `deprecated`? Какие условия должны быть соблюдены, чтобы корректно вызвать `Thread.interrupt()`? Как проще всего останавливать поток?

## Лабораторная работа №16. Организация фоновых вычислений

Цель работы: Научиться выносить длительные вычисления в фоновые потоки.

Вопросы для защиты:

- 1) Должен ли поток фоновых вычислений быть активен, когда Activity его создавшая не находится на переднем плане? Почему?
- 2) Что делать с потоком фоновых вычислений, если пользователь нажал кнопку Back? Будет ли ответ таким же, если перенести фоновые вычисления в Service?
- 3) Как должен располагаться элемент вида ProgressBar по отношению к остальным элементам?
- 4) Как получить ссылку на объект, ранее сохраненную `onRetainNonConfigurationInstance()`?
- 5) Как в `onDestroy()` отличить ситуацию пересоздания Activity от завершения приложения?

Критерии и шкала оценивания по оценочному средству защита лабораторных работ

Шкала оценивания	Критерий оценивания
5	Ответы на вопросы к защите лабораторных работ даны на высоком уровне (правильные ответы даны на 90-100% вопросов)
4	Ответы на вопросы к защите лабораторных работ даны на среднем уровне (правильные ответы даны на 75-89% вопросов)
3	Ответы на вопросы к защите лабораторных работ даны на низком уровне (правильные ответы даны на 50-74% вопросов)
2	Ответы на вопросы к защите лабораторных работ даны на неудовлетворительном уровне (правильные ответы даны менее чем на 50%)

### Вопросы для контрольных работ

1. В чем разница между методами `Editor.apply()` и `Editor.commit()`?
2. Какой метода `TextView` позволяет изменить его содержимое?
3. Какие типы данных можно передать через `Intent`? Что такое Extras?
4. Сколько компонентов можно разместить внутри `ScrollView`? Какой компонент там обычно размещается?
5. Как установить разрешение на чтение файла всеми, независимо от владельца?
6. Что будет, если обнаружится несколько приложений, способных обработать указанный `Intent`?
7. Что делать с потоком фоновых вычислений, если пользователь нажал кнопку Back? Будет ли ответ таким же, если перенести фоновые вычисления в Service?

8. Можно ли использовать одинаковое значение атрибута `android:onClick` с несколькими компонентами? Как в этом случае различать компоненты? Нужно ли задавать компонентам атрибут `android:id`?
9. В каком файле задаются ресурсы строк? Какой синтаксис используется?
10. Как программно показать контекстное меню для элемента вида? Что делает `Activity.openContextMenu()`?
11. Для чего предназначен класса `GestureDetector`? Написание какого обработчика он упрощает?
12. Нужно ли использовать `ScrollView` вместе с `TextView`? Почему?
13. Что такое класс `android.R`? Когда он используется? Почему импорт этого к класса «плохая идея»?
14. Что задает атрибут `action` класса `Intent`?
15. Как должен располагаться элемент вида `ProgressBar` по отношению к остальным элементам?
16. Как поместить `Drawable` слева от текста? Нужен ли отдельный `layout`?
17. Какие виды `layout`-ов вы знаете? Как определить ориентацию компонент в `LinearLayout`?
18. Что означает `canWrite()` для каталога? Кто может изменить содержимое файла?
19. Что означает атрибут `android:weight` для `Linear...` и `TableLayout`?
20. При каких обстоятельствах система пересоздает `Activity` путем его уничтожения и создания заново? Как при этом сохраняется состояние `Activity`?
21. В каком классе должен находиться и как быть описан обработчик события, если в `xml layout`-е использовался атрибут `android:onClick`? Если это условие нарушено, в какой момент будет обнаружена ошибка?
22. В чем отличие атрибутов `data` и `extras` класса `Intent`?
23. Как сослаться на строку, заданную в ресурсах, из других `xml` файлов проекта?
24. Для чего предназначен класс `Handler`?
25. Если у элемента `TextView` будет установлен атрибут `android:id`, будет ли его состояние автоматически сохраняться системой при изменении конфигурации? Почему?
26. Каким образом указываются `layout`-ы для альбомной ориентация дисплея устройства?
27. Какая связь между обработчиком длинного нажатия и контекстным меню? Что будет если обработчик вернет `true`?
28. Как задать заголовок и содержание письма для отправки по электронной почте?
29. Как программно разрешить покрутку текста по горизонтали в `TextView`?
30. Как сохранить ссылку на объект, когда `Activity` пересоздается при изменении конфигурации? Как получить ссылку назад?
31. Какую функциональность обеспечивает `ViewSwitcher`? В чем его отличие от `ViewFlipper`?

32. Что такое файловая система `proc`. К какому уровню Android системы она относится? Где искать описание ее компонент?
33. Что такое каталоги вида `/proc/id`, где `id` – десятичное значение? Что такое каталог `/proc/self`?
34. Почему метод `Thread.stop()` является `deprecated`? Какие условия должны быть соблюдены, чтобы корректно вызвать `Thread.interrupt()`? Как проще всего останавливать поток?
35. Должен ли поток фоновых вычислений быть активен, когда Activity его создавшая не находится на переднем плане? Почему?
36. Каким образом рекомендуется оформлять культурно-зависимые строки? Как использовать такую строку в контексте, где требуется Java тип `String`?
37. Какую часть состояния Activity система сохраняет автоматически? Какое обязательное условие для сохранения состояния элемента вида?
38. Нужно ли вызывать `super.onSaveInstanceState()` и `super.onRestoreInstanceState()`? Зачем?
39. Что такое длинное нажатие? Как задать его обработчик?
40. Что такое `toast`-уведомление? Как показать простое текстовое сообщение?
41. Какие методы получения текущего времени доступны из Java приложения?
42. Как определить, был ли метод `onCreate()` вызван в результате изменения конфигурации устройства?
43. Как в `onDestroy()` отличить ситуацию пересоздания Activity от завершения приложения?
44. Что такое `Intent`? Как явно указать имя класса Activity, которую требуется запустить?
45. Что обеспечивает установка `movement method` для `TextView`?
46. Какие требования предъявляются к классу, чтобы его экземпляр можно было передать через `Intent`? Какие дополнительные ограничения возникают, если исходная и целевая Activity находятся в разных приложениях?
47. Как получить ссылку на объект, ранее сохраненную `onRetainNonConfigurationInstance()`?
48. Что такое класс `R` приложения? Как и когда он создается?
49. Как узнать, какой из пунктов одиночного выбора активен сейчас?
50. Какие Java классы позволяют получить содержимое каталога?
51. Что такое контекстное меню? Как задать обработчик его создания? Что делает `Activity.registerForContextMenu()`?
52. Как сослаться на строку, определенную в ресурсе из файла `AndroidManifest.xml`? Все ли строки из `AndroidManifest.xml` можно вынести в ресурсы?
53. Для чего предназначены группы меню? Как группы меню отражаются в `xml` файле?
54. Допускается ли в Android обращения к UI элементам из не-UI потоков?

55. Если состояние храниться в полях класса Activity, а отображается через элементы вида, какое действие нужно предпринять в обработчике `onRestoreInstanceState()`?
56. Как поместить данные в Intent Extras?
57. В каком файле проекта указывается идентификатор приложения? С чем он должен совпадать?
58. Какие типы данных могут быть сохранены в Bundle в `onSaveInstanceState()`?
59. В каком файле задаются темы? В каком файле темы привязываются к компонентам приложения?
60. Как в файле layout сослаться на анимацию определенную в Android, а не в проекте приложения?
61. Какой метод используется для запуска Activity через Intent?
62. Если вы решили имплементировать интерфейс `View.OnClickListener` на уровне класса Activity, то каким образом отличать события, произошедшие в разных компонентах?
63. Когда вызывается метод `finish()`? Как он связан с сенсорной кнопкой Back устройства?
64. Какие теги псевдо-HTML допускаются внутри текста `TextView`?
65. Как задать идентификатор элементу вида? Всегда ли нужен идентификатор?
66. Как задать в анимации перемещение по горизонтали? В каких единицах задаются точки?
67. Какой класс предназначен для загрузки меню из xml ресурсов? Как получить его экземпляр?
68. В какой папке проекта следует располагать картинки, не привязанные к характеристикам устройства?
69. В каком файле по умолчанию сохраняются строковые ресурсы? Как задать версию файла для определенного языка?
70. Какие требования предъявляются к классу, чтобы его экземпляр мог быть сохранен в Bundle?
71. Что делает метод `Editor.clear()`?
72. Какой стандартный АСТИОН используется когда нужно передать данные за пределы приложения? В каком классе содержится эта константа?
73. Какой Intent позволяет пользователю явно выбрать приложение, используемое для обработки данных?
74. Как узнать, ссылается ли строка на каталог?
75. Как соотносятся между собой атрибуты `data` и `type` класса Intent? Можно ли задать оба атрибута одновременно?
76. Что обеспечивает метод `Activity.runOnUiThread()`? Как он реализован?
77. Как получить Intent, с помощью которого была запущена данная Activity?
78. Как получить ссылку на элемент вида, если известен его идентификатор?
79. Почему метод `overridePendingTransition()` принимает ссылки на два ресурса анимации?

80. Как указать, что данные требуется передать по электронной почте? В чем особенность задания адреса получателя?
81. Какие классы Java можно использовать для доступа к содержимому файлов?
82. Сколько видов можно поместить внутри ViewFlipper?
83. Методы onCreate() и onRestoreInstanceState() имеют параметр Bundle. В каком случае этот параметр может быть null?
84. Какие модификации объекта, помимо перемещения, могут быть заданы в tween-анимации?
85. В чем отличие getExtras().get\*() от getExtra\*()?
86. Как используя функции файловой системы узнать идентификатор процесса, в котором выполняется Android приложение?
87. Как приложение должно вести себя в UI потоке?
88. Какой метод позволяет загрузить вид из его xml описания? В какой момент он обычно вызывается?
89. Какие классы Java можно использовать для доступа к элементам файловой системы?
90. Можно ли сделать специальные версии Drawable ресурсов, зависящих от языка? Можно ли при этом оставить зависимость от разрешения дисплея?
91. В каком каталоге должны располагаться ресурсы tween-анимации?
92. Для чего предназначены методы onSaveInstanceState() и onRestoreInstanceState()?
93. Какая проблема может возникнуть с Handler.postDelayed()?
94. Что делает Activity.overridePendingTransition()? В какой момент должен вызываться этот метод?
95. Какой класс Java предназначен для манипуляций с датами? Как получить экземпляр этого класса для текущего момента?
96. Как запустить новый поток? Какая связь между Thread и Runnable?
97. Что такое SharedPreferences? Где сохраняются данные? В каком формате?
98. В чем разница между Activity.getPreferences() и Activity.getSharedPreferences()?
99. Каким образом указывается, что данная версия ресурсов предназначена для устройств с определенным разрешением дисплея? Что такое -mdp и -ldp суффиксы?
100. Что содержится в файле /proc/\*/cmdline Android процесса?
101. Какой приоритет у квалификатора локализации ресурсов по отношению к другим квалификаторам, таким как ориентация или размер экрана?
102. Какие ориентации дисплея устройств поддерживаются Android? Под какую ориентацию идет верстка по-умолчанию?
103. Что задает параметр mode метода get\*Preferences()?
104. Что происходит при повороте дисплея? Какие методы Activity вызываются?
105. Что такое символические ссылки в Linux? Как получить полное имя файла, зная имя ссылки?

106. Каким образом задать иконку приложения? Что нужно учитывать в случае растровой иконки?
107. В каком каталоге ресурсов должны располагаться описания анимации?
108. Назовите два главных способа закрепить обработчик события onClick за компонентом.
109. Как прочитать данные из SharedPreferences? Как получить все данные, не зная ключей?
110. Требуется ли кнопке назначать атрибут android:id, если обработчик события задается атрибутом android:onClick?
111. Какие преимущества можно получить, реализуя интерфейс View.OnClickListener в локальных классах, по сравнению с другими технологиями? Какой недостаток у данного метода?
112. Если требуется разместить элемент вида определенным образом по отношению к родительскому или соседним элементам, то какой тип (типы) layout-а следует предпочесть?
113. Для чего предназначен класс SharedPreferences.Editor? Как получить его экземпляр?
114. Что обеспечивает класс ViewAnimator? Какие его наследники вы знаете?
115. Какую функциональность обеспечивает ImageSwitcher?
116. Что задают атрибуты android:inAnimation и android:outAnimation?
117. Как переключиться на заданный вид ViewFlipper? Как задать автоматическую прокрутку через временной интервал?

### *Типовые варианты контрольных работ*

#### ВАРИАНТ 1

1. Какие классы Java можно использовать для доступа к содержимому файлов?
2. Сколько видов можно поместить внутри ViewFlipper?
3. Методы onCreate() и onRestoreInstanceState() имеют параметр Bundle. В каком случае этот параметр может быть null?
4. Какие модификации объекта, помимо перемещения, могут быть заданы в tween-анимации?
5. В чем отличие getExtras().get\*() от getExtra\*()?
6. Как используя функции файловой системы узнать идентификатор процесса, в котором выполняется Android приложение?

## ВАРИАНТ 2

1. Какая проблема может возникнуть с `Handler.postDelayed()`?
2. Что делает `Activity.overridePendingTransition()`? В какой момент должен вызываться этот метод?
3. Какой класс Java предназначен для манипуляций с датами? Как получить экземпляр этого класса для текущего момента?
4. Как запустить новый поток? Какая связь между `Thread` и `Runnable`?
5. Что такое `SharedPreferences`? Где сохраняются данные? В каком формате?
6. В чем разница между `Activity.getPreferences()` и `Activity.getSharedPreferences()`?

## ВАРИАНТ 3

1. Почему метод `Thread.stop()` является `deprecated`? Какие условия должны быть соблюдены, чтобы корректно вызвать `Thread.interrupt()`? Как проще всего останавливать поток?
2. Должен ли поток фоновых вычислений быть активен, когда `Activity` его создавшая не находится на переднем плане? Почему?
3. Каким образом рекомендуется оформлять культурно-зависимые строки? Как использовать такую строку в контексте, где требуется Java тип `String`?
4. Какую часть состояния `Activity` система сохраняет автоматически? Какое обязательное условие для сохранения состояния элемента вида?
5. Нужно ли вызывать `super.onSaveInstanceState()` и `super.onRestoreInstanceState()`? Зачем?
6. Что такое длинное нажатие? Как задать его обработчик?

Критерии и шкала оценивания по оценочному средству контрольная работа

Шкала оценивания	Критерий оценивания
5	Контрольная работа выполнена на высоком уровне (правильные ответы даны на 90-100% вопросов)
4	Контрольная работа выполнена на среднем уровне (правильные ответы даны на 75-89% вопросов)
3	Контрольная работа выполнена на низком уровне (правильные ответы даны на 50-74% вопросов)
2	Контрольная работа выполнена на неудовлетворительном уровне (правильные ответы даны менее чем на 50%)

## Оценочные средства для промежуточной аттестации (экзамен)

1. Как приложение идентифицируется в системе?
2. Что хранится в папке res/values?
3. Как в проекте организовано хранение ресурсов, в чем отличие папок res и assets?
4. Что такое debug.keystore?
5. Назначение файлов adbkey и adbkey.pub
6. Почему наличие только исходных кодов может быть недостаточно для генерации обновлений приложения?
7. В чем назначение файла AndroidManifest.xml?
8. Как активировать отладку через WiFi на Android устройстве?
9. Как подключить ADB к устройству через сеть, зачем это нужно?
10. Что хранится в файле res/layouts?
11. Что такое View и ViewGroup, в чем назначение классов \*Layout?
12. Что такое padding, margin и gravity для визуальных элементов? В каких единицах задаются?
13. Как задаются размеры визуальных компонентов, что значит wrap\_content?
14. В каких единицах задаются размеры шрифтов?
15. Какой смысл имеет параметр weight для LinearLayout?
16. Как в RelativeLayout задать положение компонента относительно родителя? Относительно другого компонента?
17. Как организовать прокрутку пользовательского интерфейса при нехватке места на экране?
18. Как задать отдельную верстку для альбомной и планшетной ориентации устройства?
19. Как организовать поддержку нескольких языков в пользовательском интерфейсе?
20. Нужен ли отдельный Layout для реализации «картинки с подписью»?
21. Что такое R класс, как он создается? Как назначить id компоненту?
22. Как получить ссылку на визуальный элемент по его id? Как получить id, имея ссылку на визуальный компонент?
23. Как задать обработчик события click в файле layout? Какие требования предъявляются к обработчику?
24. Как задать обработчик события в коде? В чем преимущества анонимных классов?
25. Можно ли назначить один обработчик нескольким элементам? Как отличить в каком из элементов произошло событие?
26. Когда в приложении отображается меню? Какой обработчик отвечает за наполнение меню приложения?
27. Как программно добавить пункты в меню приложения? Может ли обработчик предотвратить отображение меню? Как?

28. Какой метод позволяет приложению подредктировать меню перед отображением?
29. Что такое группа меню? Какой метод позволяет добавить меню в конкретную группу? В чем преимущество групп меню?
30. Какой метод отвечает за обработку команд, выбранных в меню приложения? Что это метод получает на вход? Как определить какой пункт меню был выбран? Как получить ссылку на визуальный элемент по его id?
31. Что общего и в чем отличие в поведении методов onCreateOptionsMenu и onCreateContextMenu?
32. В каком методе происходит обработка команд контекстного меню?
33. Опишите два способа задания контекстного меню элемента? Какой из них применяется, для случая когда обработчиком является Activity?
34. Как вывести сообщение в системный лог? Какие уровни важности определены для логов?
35. Что такое Activity, как задается в коде и в манифесте
36. Что такое Task. Связь между приложением, процессом и Task-ом
37. При каком условии Activity разных приложений оказываются в одном Task
38. Могут ли Activity разных приложений оказаться в одном процессе
39. Жизненный цикл Activity, чем отличаются состояния Start и Resume
40. Что происходит при изменении конфигурации устройства, какие данные сохраняются автоматически
41. Как сохранить внутренние данные Activity с учетом жизненного цикла
42. Как запускается Activity, что такое action?
43. Что такое intent filter, его параметры
44. Intent, явное и косвенное указание Activity
45. Intent, передача данных через Extra
46. Возврат результата в вызывающий Activity, назначение requestCode
47. Параметры Data и Type класса Intent, указание объекта через Uri.
48. Как задать intent filter для определенного типа данных?
49. Что такое Parcel?
50. Сравните интерфейсы Serializable и Parcelable.
51. ACTION\_SEND, передача текста в другое приложение, отправка письма.
52. Где приложение может размещать собственные изменяемые данные
53. Где хранятся пользовательские данные
54. Классы для чтения данных в формате XML
55. Классы для работы с каталогами и файлами
56. Хранение данных. Работа с файлами. XmlPullParser. Разбор XML документов.
57. Как задать Activity поведение диалога
58. В чем отличие диалогов от Activity
59. Встроенные диалоги для работы с датой и временем
60. Встроенные ProgressDialog

- 61.Классы для построения диалогов. `AlertBuilder.Dialog`.
- 62.Кастомизация `AlertBuilder.Dialog`. Верстка внутреннего пространства диалога
- 63.Жизненный цикл диалога. Повторная инициализация
- 64.Обработка событий в диалогах
- 65.Интерфейс `onTouchListener`. Класс `MotionEvent`.
- 66.Метод `MotionEvent.getAction`. События при одиночном касании.
- 67.Класс `MotionEvent`. Индексы и идентификаторы касаний.
- 68.События при множественном касании.
- 69.Ресурсы tween-анимации. Описание анимации в xml файлах. Scale-анимация.
- 70.Анимация. Трансформации и вращения. Наборы анимации.
- 71.Программная загрузка и применение анимации к элементам вида.
- 72.Назначение анимации в layout-файлах
- 73.Переопределение анимации переключения `Activity`
- 74.Класс `Canvas`. Рисование в основном потоке.
- 75.Класс `SurfaceView`. Доступ к канве из фонового потока.
- 76.Класс `Paint`. Настройка параметров рисования.
- 77.Класс `Paint`. Рисование элементарных фигур. Обводка и закрашивание.
- 78.Класс `Paint`. Вывод текста. Получение параметров результирующего растра.
- 79.Класс `Paint`. Вывод текста. Применение шрифтов и стилей.
- 80.Класс `Picture`. Внеэкранный отрисовка. Вывод растра.
- 81.Класс `Handler`. Посылка и обработка сообщений.
- 82.Класс `Handler`. Отложенная обработка и удаление из очереди.
- 83.Интерфейс `Handler.Callback`.
- 84.Передача `Runnable` в `Handler`.
- 85.Методы `Activity` и `View` для обращения к UI потоку.
- 86.Класс `AsyncTask`. Параметры шаблона.
- 87.Класс `AsyncTask`. Методы `onPreExecute`, `doInBackground`, `onPostExecute`.
- 88.Промежуточные результаты `AsyncTask`. Методы `publishProgress` и `onProgressUpdate`
- 89.Окончательный результат `AsyncTask`. Метод `get`.
- 90.Статусы задачи. Поведение `AsyncTask` при отмене задачи.
- 91.Асинхронная обработка и смена конфигурации устройства.
- 92.Компонент `Service`. Методы `onCreate`, `onDestroy`, `onStartCommand`
- 93.Компонент `Service`. Регистрация в манифесте. Запуск по ACTION.
- 94.Запуск и остановка сервиса, методы, настройка `Intent`
- 95.Компонент `Service`. Основной и вспомогательные потоки.
- 96.Передача данных в сервис через `Intent`.
- 97.Метод `stopSelf` сервиса. Порядок завершения вызовов.
- 98.Метод `onStartCommand`. Возвращаемое значение. Автоматический перезапуск и повторные вызовы.
- 99.Запуск сервиса в отдельном процессе. Параметры манифеста.

100. Возврат данных из сервиса. Механизм PendingIntent. Методы createPendingResult и stopSelfResult.
101. Возврат данных из сервиса. Механизм BroadcastReceiver. Метод sendBroadcast.
102. Подключение к сервису. Методы bindService и unbindService. Режим BIND\_AUTO\_CREATE. Интерфейс ServiceConnection.
103. Связь с сервисом через интерфейс IBinder. Метод getService. Локальный биндинг.
104. Классы Notification, Notification.Builder и NotificationService. Получение ссылки на сервис. Добавление уведомления
105. Идентификатор уведомления. Обновление и удаление уведомлений. Автоматическое удаление уведомлений
106. Обработка нажатия на уведомление. Механизм PendingIntent и флаги.
107. Механизм PendingIntent. Класс TaskBuilder. Отображение стека Activity.
108. Отображение индикатора прогресса в уведомлении. Неопределенное состояние. Скрытие индикатора.
109. Расширенные уведомления. Стилль BigPictureStyle.
110. Расширенные уведомления. Стилль InboxStyle.
111. Расширенные уведомления. Стилль MessagingStyle.
112. Дополнительные кнопки в уведомлениях. Механизм PendingIntent.
113. Кастомизация уведомлений. Механизм RemoteViews.

Типовой экзаменационный билет.

## **ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. В.ДАЛЯ**

### **Кафедра информационных и управляющих систем**

#### **Вариант 1**

1. Как сохранить внутренние данные Activity с учетом жизненного цикла?
2. Что такое padding, margin и gravity для визуальных элементов? В каких единицах задаются?
3. Могут ли Activity разных приложений оказаться в одном процессе?

Утверждено на заседании кафедры информационных и управляющих систем \_\_\_\_\_, протокол \_\_\_\_\_

Заведующий кафедрой

А.И. Горбунов

Экзаменатор

Д.О. Синепольский

Критерии и шкала оценивания результатов промежуточной аттестации  
«Экзамен»

Шкала оценивания (интервал баллов)	Критерий оценивания
отлично (5)	Студент глубоко и в полном объеме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.
Хорошо (4)	Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.
удовлетворительно (3)	Студент знает только основной программный материал, допускает неточности, недостаточно четкие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.
неудовлетворительно (2)	Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы

### Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)