

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ
Декан факультета информационных
технологий и компьютерной безопасности
 / П.Ю. Гусев /
подпись / *И.О. Фамилия*
«31» августа 2021 г.

РАБОЧАЯ ПРОГРАММА
дисциплины (модуля)
«Программирование мобильных платформ»
наименование дисциплины (модуля) в соответствии с учебным планом)

Направление подготовки 09.03.01 Информатика и вычислительная техника
код и наименование направления подготовки/специальности

Профиль (специализация) Вычислительные машины, комплексы, системы и сети
название профиля/программы

Квалификация выпускника бакалавр

Нормативный период обучения 4 года / 4 года и 11 м.
Очная/очно-заочная/заочная (при наличии)

Форма обучения Очная/Заочная

Год начала подготовки 2019 г.

Автор(ы) программы доцент  В.В. Сафронов
должность и подпись

**Заведующий кафедрой
Автоматизированных и
вычислительных систем**  В.Ф. Барабанов
наименование кафедры, реализующей дисциплину *подпись*

Руководитель ОПОП  С.Л. Подвальный

Воронеж 2021

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

Целью освоения дисциплины является теоретическая подготовка студентов в области разработки программ для мобильных устройств (смартфоны на Android, айфоны – Iphone, планшеты) с использованием современных языков программирования. Получение практических навыков по созданию пользовательских интерфейсов, сервисов и по использованию сигнализации, аппаратных сенсоров и стандартных хранилищ информации в рамках выбранной платформы (Android, Iphone).

Дисциплина включает:

- изучение основ и получение практических навыков программной инженерии в области разработки программного обеспечения для мобильных устройств;

- ознакомление студентов с основными принципами разработки приложений для операционных систем Android, WindowsPhone и iOS и технологией создания мобильных приложений с использованием различных языков программирования.

1.2. Задачи освоения дисциплины

- ознакомление с основными мобильными операционными системами;
- формирование знаний об особенностях программирования мобильных устройств и микрокомпьютеров;

- изучение методов программирования и установки программ для мобильных устройств и микрокомпьютеров;

- получение представления о жизненном цикле приложений и их структуре, программном манифесте и внешних ресурсах, основных доступных элементах пользовательского интерфейса, работе с файлами, базами данных, пользовательскими настройками, разделяемыми данными и межпрограммном взаимодействии;

- изучение инструментов для программирования и основ проектирования мобильных приложений;

- исследование программных интерфейсов, обеспечивающих функции телефонии, отправки/получения SMS, поддержку соединений посредством Wi-Fi/Bluetooth;

- исследование возможностей взаимодействия с геолокационными, картографическими сервисами;

- изучение способов создания фоновых служб, сигнализации и подключения механизма уведомлений;

- решение практических задач по созданию представлений, программированию сервисов, фоновых служб;

- получение практических навыков по разработке мобильного приложения с применением всех изученных принципов, методик, методов и средств разработки.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Программирование мобильных платформ» относится к дисциплинам базовой части блока ФТД.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Программирование мобильных платформ» направлен на формирование следующих компетенций:

ПК-2 - Способен проектировать и разрабатывать компоненты программных комплексов и информационных систем, используя современные технологии программирования и инструментальные средства разработки

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-2	Знать принципы построения архитектуры и виды архитектуры программного обеспечения мобильных платформ iOS, Android и Windows.
	Уметь применять методы и средства проектирования и разработки программного обеспечения мобильных платформ с использованием C++ и Objective-C для iOS и Android.
	Владеть методиками применения современных технологий программирования и инструментальных средств разработки мобильных платформ под Android платформу.

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Программирование мобильных платформ» составляет 2 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры
		5
Аудиторные занятия (всего)	36	36
В том числе:		
Лекции	18	18
Лабораторные работы (ЛР)	18	18
Самостоятельная работа	36	36
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость:		
академические часы	72	72
зач.ед.	2	2

заочная форма обучения

Виды учебной работы	Всего часов	Семестры
		6
Аудиторные занятия (всего)	8	8
В том числе:		

Лекции	4	4
Лабораторные работы (ЛР)	4	4
Самостоятельная работа	60	60
Контрольная работа	+	+
Часы на контроль	4	4
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость: академические часы	72	72
зач.ед.	2	2

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Классификация мобильных устройств. Архитектура мобильных устройств и их компонентов.	<i>Особенности и назначение мобильных устройств.</i> История появления мобильных устройств, архитектура мобильных устройств, операционные системы для мобильных устройств, современные мобильные устройства на примере устройств для ОС iOS, Android и Windows. <i>Операционные системы для мобильных устройств.</i> Возможности современных ОС для мобильных устройств, API ОС мобильных устройств, Java для мобильных устройств и библиотеки ОС iOS и Android. <i>Программирование мобильных устройств.</i> Java: архитектура и возможности, применимость, недостатки и преимущества; программирование на C++ и Objective-C для OS iOS; программирование на Java и C++ для ОС Android; сторонние средства разработки мобильных приложений.	4	2	6	12
2	Пользовательский интерфейс.	Элементы разметки пользовательских приложений. Использование меню. Элементы управления пользовательского интерфейса. Манифест приложения. Жизненный цикл формы, приложения. Проектирование интерфейсов для мобильных устройств и планшетов. Проектирование интерфейсов, не зависящих от размера экрана. Фрагменты.	4	4	12	20
3	Использование сервисов мобильной платформы.	Сигнализация. Отложенная сигнализация. Картографические сервисы. Фоновые службы и процессы. Служба компоновки и создания фрагментов. Управление камерой. Управление сенсорами. Управление сетевыми соединениями. Получение информации об устройстве. Служба отправки и получения СМС. Поддержка протоколов Bluetooth/Wi-Fi. Установка шлюза через Wi-Fi Direct. Управление анимацией. Использование NFC. Служба push-нотификаций. Служба уведомлений и доставки. Управление потоками и асинхронными задачами.	6	6	12	24
4	Разработка	Анимация на основе ключевых кадров.	4	6	6	16

	эффективного приложения.	Техники программирования, сохраняющие заряд батареи. Энергосберегающие паттерны программирования. Межпроцессное взаимодействие. Рекомендации по написанию мобильных приложений.				
Итого			18	18	36	72

заочная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Классификация мобильных устройств. Архитектура мобильных устройств и их компонентов.	<i>Особенности и назначение мобильных устройств.</i> История появления мобильных устройств, архитектура мобильных устройств, операционные системы для мобильных устройств, современные мобильные устройства на примере устройств для ОС iOS, Android и Windows. <i>Операционные системы для мобильных устройств.</i> Возможности современных ОС для мобильных устройств, API ОС мобильных устройств, Java для мобильных устройств и библиотеки ОС iOS и Android. <i>Программирование мобильных устройств.</i> Java: архитектура и возможности, применимость, недостатки и преимущества; программирование на C++ и Objective-C для OS iOS; программирование на Java и C++ для ОС Android; сторонние средства разработки мобильных приложений.	1	1	10	12
2	Пользовательский интерфейс.	Элементы разметки пользовательских приложений. Использование меню. Элементы управления пользовательского интерфейса. Манифест приложения. Жизненный цикл формы, приложения. Проектирование интерфейсов для мобильных устройств и планшетов. Проектирование интерфейсов, не зависящих от размера экрана. Фрагменты.	1	1	20	22
3	Использование сервисов мобильной платформы.	Сигнализация. Отложенная сигнализация. Картографические сервисы. Фоновые службы и процессы. Служба компоновки и создания фрагментов. Управление камерой. Управление сенсорами. Управление сетевыми соединениями. Получение информации об устройстве. Служба отправки и получения СМС. Поддержка протоколов Bluetooth/Wi-Fi. Установка шлюза через Wi-Fi Direct. Управление анимацией. Использование NFC. Служба push-нотификаций. Служба уведомлений и доставки. Управление потоками и асинхронными задачами.	1	1	20	22
4	Разработка эффективного приложения.	Анимация на основе ключевых кадров. Техники программирования, сохраняющие заряд батареи. Энергосберегающие паттерны программирования. Межпроцессное взаимодействие. Рекомендации по написанию мобильных приложений.	1	1	10	12
Итого			4	4	60	68

5.2 Перечень лабораторных работ

Лабораторная работа №1. Минимальная Android программа.

Лабораторная работа №2. Интерактивная Android программа.

Лабораторная работа №3. Android программа с несколькими активностями.

Лабораторная работа №4. Разработка мобильного приложения «Записная книжка».

- Лабораторная работа №5.** Разработка мобильного приложения «Карманный навигатор».
- Лабораторная работа №6.** Разработка мобильного приложения «Песочные часы».
- Лабораторная работа №7.** Разработка мобильного приложения «Обмен мгновенными сообщениями».
- Лабораторная работа №8.** Разработка мобильного приложения «Калькулятор».

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины для очной формы обучения не предусматривает выполнение курсового проекта (работы) или контрольной работы.

При освоении дисциплины по заочной форме обучения предусматривается выполнение контрольной работы в 6 семестре по теме «Разработка мобильного приложения под Android». Индивидуальная часть задания выполняется по вариантам.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ПК-2	Знать принципы построения архитектуры и виды архитектуры программного обеспечения мобильных платформ iOS, Android и Windows.	Активная работа на практических занятиях, отвечает на теоретические вопросы при защите лабораторных работ.	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь применять методы и средства проектирования и разработки программного обеспечения мобильных платформ с использованием C++ и Objective-C для iOS и Android.	Решение стандартных практических задач, выполнение лабораторных работ.	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть методиками применения современных технологий программирования и инструментальных средств разработки мобильных платформ под Android платформу.	Решение прикладных задач в конкретной предметной области, выполнение плана работ по лабораторным работам.	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 5 семестре для очной формы обучения, 6 семестре для заочной формы обучения по двухбалльной системе:

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Не зачтено
ПК-2	Знать принципы построения архитектуры и виды архитектуры программного обеспечения мобильных платформ iOS, Android и Windows.	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	Уметь применять методы и средства проектирования и разработки программного обеспечения мобильных платформ с использованием C++ и Objective-C для iOS и Android.	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть методиками применения современных технологий программирования и инструментальных средств разработки мобильных платформ под Android платформу.	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типичные контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

1. Какие методы отражают жизненный цикл Activity:

onCreate() (+)
onRestart() (+)
onResume() (+)
onRunning()
onClose()

2. Подробная информация о приложении содержится в файле:

AndroidManifest.xml (+)
main.xml *R.java*
default.properties

3. Для создания всплывающего уведомления необходимо инициализировать объект:

Toast (+)
Message
TextView
MessageBox

4. Единица измерения *dp* или *dip* - это 1/72 дюйма, определяется по физическому размеру экрана дюйм; определяется по физическому размеру экрана абстрактная ЕИ, позволяющая приложениям выглядеть одинаково на различных экранах и разрешениях; (+) физический элемент матрицы дисплея

5. Тип верстки при котором позиционирование элементов происходит относительно друг друга и относительно главного контейнера

AbsoluteLayout

FrameLayout

LinearLayout

RelativeLayout (+)

6. Какой класс можно использовать для перехода между Activity?

ActivityChanger

Activity

Intent (+)

Switcher

7. От какого класса наследуются все элементы управления?

Control

Controls

Element

View (+)

8. Для чего используется класс R?

Класс ресурсов; (+)

Класс для управления элементами управления;

Класс для работы с потоками;

Класс, предназначенный только для доступа к идентификаторам элементов

9. Как программно получить доступ к элементам управления через идентификатор?

getViewById(id)

findViewById(id) (+)

findViewById(id)

getView(id)

10. В какой библиотеке находится класс Activity?

android

android.Activities

android.app (+)

java.app

11. На основе какой операционной системы была разработана система Android:

Linux (+)

MiniX

Windows iOS

12. Как называется режим использования Android-системы с максимальными правами (аналог администратора Windows):

Суперпользователь (+)

Root-Home пользователь Android

user

Up-User

13. Как называется событие-клик объекта-кнопки:

OnClick (+)

OnClicked

Click

ClickEvent

14. Для добавления в приложение более одного экрана взаимодействия необходимо создание:

Activity (+)
View
Layout
XML-файла

15. Как называется класс для работы с SQLite-базами в Android:

SQLiteOpenHelper (+)
SQLProvider
SQLiteProvider
SQLiteContext

16. Как называется язык разметки Window Phone приложений:

XAML (+)
XML
Java
C#

17. С помощью какого класса можно вывести сообщение в Window Phone приложении:

Toast
MessageBox (+)
Message
ShowClass

18. Какой элемент предоставляет вариант разметки, элементы, помещенные в который, располагаются в стек горизонтально или вертикально:

StackPanel (+)
Pivot
Panorama
Canvas

19. Какой элемент позволят позиционировать элементы внутри себя максимально гибко предоставляя возможность размещать элементы, используя строки и столбцы.

Grid (+)
Pivot
Panorama
Canvas

20. Какой элемент предоставляет наиболее простой вариант разметки. Он может быть использован для абсолютного позиционирования элементов с использованием координат.

Grid
Pivot
Panorama
Canvas (+)

7.2.2 Примерный перечень заданий для решения стандартных задач

Задание 1. Имеется следующий код:

```
public class Overload {  
    public void method(Object o) {  
        System.out.println("Object");  
    }  
    public void method(java.io.FileNotFoundException f) {  
        System.out.println("FileNotFoundException");  
    }  
}
```

```

public void method(java.io.IOException i) {
    System.out.println("IOException");
}
public static void main(String args[]) {
    Overload test = new Overload();
    test.method(null);
}
}

```

Результатом его компиляции и выполнения будет:

1. Ошибка компиляции
2. Ошибка времени выполнения
3. «Object»
4. «File Not Found Exception»
5. «IO Exception»

Задание 2.

```

Float f1 = new Float(Float.NaN);
Float f2 = new Float(Float.NaN);
System.out.println( "" + (f1 == f2) + " " + f1.equals(f2) + " " + (Float.NaN ==
Float.NaN) );

```

Что будет выведено в результате выполнения данного куска кода:

1. falsefalsefalse
2. falsetruefalse
3. truetruefalse
4. Falsetruetrue
5. truetruetrue

Задание 3.

```

class Mountain {
    static String name = "Himalaya";
    static Mountain getMountain() {
        System.out.println("GettingName ");
        return null;
    }
}
public static void main(String[] args) {
    System.out.println( getMountain().name );
}
}

```

Что произойдет при попытке выполнения данного кода:

1. Будет выведено «Himalaya» но НЕ будет выведено «GettingName ,,
2. Будет выведено «Getting Name » и «Himalaya»
3. Ничего не будет выведено
4. Будет выброшен NullPointerException
5. Будет выведено «GettingName », а потом выброшено NullPointerException

Задание 4.

```

Integer a = 120;
Integer b = 120;

```

```
Integer c = 130;
Integer d = 130;
System.out.println(a==b);
System.out.println(c==d);
```

В результате выполнения данного кода будет выведено:

1. true true
2. false false
3. false true
4. true false
5. произойдет ошибка времени выполнения

Задание 5.

```
//In File Other.java
package other;
public class Other { public static String hello = "Hello"; }
//In File Test.java
package testPackage;
import other.*;
class Test{
public static void main(String[] args) {
String hello = "Hello", lo = "lo";
System.out.print((testPackage.Other.hello == hello) + " ");
System.out.print((other.Other.hello == hello) + " ");
System.out.print((hello == ("Hel"+"lo")) + " ");
System.out.print((hello == ("Hel"+lo)) + " ");
System.out.println(hello == ("Hel"+lo).intern());
}
}
class Other { static String hello = "Hello"; }
```

В результате мы получим:

1. false true true false true
2. false false true false true
3. true true true true true
4. true true true false true
5. Все ответы неверны

Задание 6. Дана сигнатура метода:

```
public static <E extends CharSequence> List<? super E> doIt(List<E> nums)
```

Который вызывается:

```
result = doIt(in);
```

Какого типа должны быть result и in?

1. ArrayList<String> in; List<CharSequence> result;
2. List<String> in; List<Object> result;
3. ArrayList<String> in; List result;
4. List<CharSequence> in; List<CharSequence> result;
5. ArrayList<Object> in; List<CharSequence> result;

Задание 7.

```

public static void doIt(String String) { //1
inti = 10;
i : for (int k = 0 ; k < 10; k++) { //2
System.out.println( String + i); //3
if( k*k > 10) continue i; //4
}
}

```

Данный код:

1. Не скомпилируется из-за строки 1
2. Не скомпилируется из-за строки 2
3. Не скомпилируется из-за строки 3
4. Не скомпилируется из-за строки 4
5. Скомпилируется и запустится без проблем

Задание 8.

```

public class Main {
static void method(int... a) {
    System.out.println("inside int...");
}
static void method(long a, long b) {
System.out.println("inside long");
}
static void method(Integer a, Integer b) {
System.out.println("inside INTEGER");
}
public static void main(String[] args) {
int a = 2;
int b = 3;
method(a,b);
}
}

```

В результате мы получим:

1. Ошибку компиляции
2. Ошибку времени выполнения
3. «insideint...»
4. «insidelong»
5. «inside INTEGER»

Задание 9.

```

class Super { static String ID = "QBANK"; }
class Sub extends Super {
static { System.out.print("InSub"); }
}
class Test {
public static void main(String[] args) {
System.out.println(Sub.ID);
}
}

```

В результате выполнения данного кода:

1. Он даже не скомпилируется
2. Результат зависит от реализации JVM
3. Будет выведено «QBANK»
4. Будет выведено «InSub» и «QBANK»
5. Все ответы неверны

Задание 10. Имеется два класса:

```
//in file A.java
package p1;
public class A{
protected inti = 10;
public intgetI() { return i; }
}
//in file B.java
package p2;
import p1.*;
public class B extends A{
public void process(A a) {
a.i = a.i*2;
}
public static void main(String[] args) {
A a = new B();
B b = new B();
b.process(a);
System.out.println( a.getI() );
}
}
```

В результате выполнения класса В мы получим:

1. Будет выведено «20»
2. Будет выведено «10»
3. Код не скомпилируется
4. Возникнет ошибка времени выполнения
5. Все ответы неверны

7.2.3 Примерный перечень заданий для решения прикладных задач

Задание 1. Создать программу, которая выводит в элемент TextView надпись, введённую пользователем в текстовом поле EditText после нажатия на кнопку Button. Помимо этого, в Activity должен быть TextView с ФИО студента и группой.

Задание 2. Создать приложение, которое состоит из нескольких activities. Первое activity содержит элемент TextView с названием или номером activity, текстовое поле EditText для ввода какой-то информации, кнопку Button с названием "Next" или "Перейти на 2 activity/экран/окно" или просто "2". Помимо этого, в 1 activity должен быть TextView с ФИО студента и группой. После нажатия на эту кнопку происходит переход на второе activity, где содержится TextView с названием или номером activity, TextView с надписью: "В первом окне вы напечатали:" и под ним - ещё один TextView с содержимым EditText с первого activity, и кнопка "Вернуться к вводу текста", нажав на которую пользователь может перейти обратно к 1 activity.

Задание 3. Написать приложение, работающее с разными темами/стилями. Сначала создать свой стиль и применить его к какому-нибудь интерфейвному элементу, затем - свою тему, которая применяется ко всем интерфейсным элементам.

Задание 4. Создать пользовательский список. Например, получить доступ в приложении к контактам (Permissions-закладка в AndroidManifest.xml) и скопировать контакты телефона в свой список, который отобразить после запуска приложения. Или создать свой список в виде твиттера (картинка+текст), элементы которого просто статически задать в массиве (как и картинки).

Задание 5. Создать приложение, содержащее анимированные интерфейсные элементы (например, увеличивающиеся при клике на них кнопки, вращающиеся TextView и т.д.).

Задание 6. Создать приложение, отображающее после запуска карты Google или другие карты.

Задание 7. Создать собственный виджет с настройками. Например, виджет, который открывает какой-то сайт, адрес которого можно менять в настройках.

Задание 8. Создать приложение, использующее опциональное меню (меню настроек) и контекстное меню для интерфейсного элемента. Выбор пунктов меню должен что-то менять в интерфейсных элементах или их отображении. Например, очистить поле ввода через контекстное меню, или отобразить невидимые интерфейсные элементы через установку галок в опциональном меню.

Задание 9. Создать приложение, отображающее после некоторых действий (нажатия на кнопку, или проверки корректности ввода текста в EditText) диалоговое окно, свидетельствующее об ошибке/информирующее/предупреждающее пользователя.

Задание 10. Создать приложение, помещающее по нажатию на кнопку сообщение со звуком в панель уведомлений/статус-панель.

Задание 11. Создать приложение, по нажатию кнопки в котором проигрывается звук.

Задание 12. Создать приложение, при запуске которого проигрывается видео.

Задание 13. Создать приложение, при запуске которого активируется фотокамера телефона, производится снимок, и этот снимок помещается в ImageView интерфейса приложения.

Задание 14. Создать приложение, работающее с SharedPreferences и сохраняющее настройки, а также работающее с БД SQLite - заполняющее БД по нажатию кнопки 1 с помощью EditText, и выводящее все записи этой БД в интерфейсный элемент ниже с помощью кнопки 2 (в виде списка, datagrid или просто правильно настроенного Text View).

7.2.4 Примерный перечень вопросов для подготовки к зачету

1. Классификация видов мобильных приложений
2. Преимущества использования мобильных приложений в сравнении с веб-приложениями

3. Недостатки использования мобильных приложений в сравнении с десктопными приложениями
4. Инструментальные среды разработки мобильных приложений для операционной системы Apple iOS
5. Инструментальные среды разработки мобильных приложений для операционной системы Android
6. Инструментальные среды разработки мобильных приложений для операционной системы Windows
7. Структура операционной системы iOS
8. Структура операционной системы Android
9. Структура приложения iOS
10. Структура приложения Android
11. Основные требования к интерфейсу приложений iOS
12. Основные требования к интерфейсу приложений Android
13. Взаимодействие Android-приложения с сетью
14. Работа Android-приложения с локальной базой данных
15. Считывание информации Android-приложением с XML-файла
16. Вызов приложения из другого приложения в ОС Android
17. Проблемы безопасности мобильных операционных систем
18. Бизнес-модели распространения мобильных приложений
19. HTML5 и мобильные приложения
20. Проектирование мобильных приложений с использованием C++
21. Технологии фреймворков в проектировании мобильных приложений
22. Проблемы масштабирования СУБД в мобильных приложениях
23. История появления мобильных устройств и их архитектура.
24. Операционные системы для мобильных устройств.
25. Клиент-серверное взаимодействие мобильных приложений.
26. Доступ к оборудованию в ОС Android.
27. C++ программы для ОС Android. Преимущества и недостатки.
28. C++ программы для ОС Android. Задачи, для которых целесообразно применять C++.
29. Концепция закрытой экосистемы Apple.

7.2.5 Примерный перечень вопросов для подготовки к экзамену

Не предусмотрено учебным планом

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Зачет проводится по тест-билетам, каждый из которых содержит 10 вопросов, 10 стандартных задач и 10 прикладных задач. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом. Максимальное количество набранных баллов – 30.

Оценка «не зачтено» ставится в случае, если студент набрал менее 16 баллов.

Оценка «зачтено» ставится в случае, если студент набрал от 16 до 30 баллов.

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Классификация мобильных устройств. Архитектура мобильных устройств и их компонентов.	ПК-2	Тест, защита лабораторных работ, устный опрос, зачет
2	Пользовательский интерфейс.	ПК-2	Тест, защита лабораторных работ, устный опрос, зачет
3	Использование сервисов мобильной платформы.	ПК-2	Тест, защита лабораторных работ, устный опрос, зачет
4	Разработка эффективного приложения.	ПК-2	Тест, защита лабораторных работ, устный опрос, зачет

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1. Введение в разработку приложений для ОС Android [Электронный ресурс]/ Ю.В. Березовская [и др.]. - Электрон. текстовые данные. - М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 433 с. - Режим доступа: <http://www.iprbookshop.ru/73669.html>. - ЭБС «IPRbooks».

2. Семакова А. Введение в разработку приложений для смартфонов на ОС Android [Электронный ресурс]/ Семакова А. - Электрон. текстовые данные. - М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 102 с. - Режим доступа: <http://www.iprbookshop.ru/73670.html>. - ЭБС «IPRbooks».

3. Разработка приложений для мобильных интеллектуальных систем на платформе Intel Atom [Электронный ресурс]/ К.С. Амелин [и др.]. - Электрон. текстовые данные. - М.: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019. - 201 с. - Режим доступа: <http://www.iprbookshop.ru/79719.html>. - ЭБС «IPRbooks».

4. Верескун Д.М. Разработка мобильных приложений для бизнеса [Электронный ресурс]: учебное пособие/ Верескун Д.М. - Электрон. текстовые данные. - Саратов: Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2012. - 51 с. - Режим доступа: <http://www.iprbookshop.ru/76508.html>. - ЭБС «IPRbooks».

5. Методические рекомендации по выполнению контрольных работ для бакалавров направления 09.03.01 профиля «Вычислительные машины, комплексы, системы и сети», магистров профиля 09.04.01 Информатика и вычислительная техника, программа: Распределенные автоматизированные системы очной формы обучения / ФГБОУ ВО «Воронежский государственный технический университет»; сост. А.М. Нужный, Ю.С. Акинина, Н.И. Гребенникова. Воронеж: Изд-во ВГТУ, 2020. – 8с.

6. Организация самостоятельной работы обучающихся : методические указания для студентов, осваивающих основные образовательные программы высшего образования – бакалавриата, специалитета, магистратуры: методические указания / сост. В.Н. Почечихина, И.Н. Крючкова, Е.И. Головина, В.Р. Демидов; ФГБОУ ВО «Воронежский государственный технический университет ». – Воронеж, 2020. – 14 с

8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

Лицензионное ПО:

- Windows Professional 7 Single Upgrade MVL A Each Academic
- Microsoft Office Word 2007
- Microsoft Office Power Point 2007

Свободно распространяемое ПО:

- Microsoft Visual Studio Community Edition
- Microsoft SQL Server Express
- OpenJDK
- Qt
- PostgreSQL
- SQLite

Отечественное ПО:

- Яндекс.Браузер
- Архиватор 7z

- Astra Linux

Ресурс информационно-телекоммуникационной сети «Интернет»:

- Образовательный портал ВГТУ

- <http://www.edu.ru/>

- <https://metanit.com/>

- <https://developer.android.com/index.html> - Сайт для разработчиков под Android

- <https://android-arsenal.com/> - Коллекция библиотек под Android

- <https://android-developers.blogspot.ru/> - Блог разработчиков под Android

Информационно-справочные системы:

- <http://window.edu.ru>

- <https://wiki.cchgeu.ru/>

Современные профессиональные базы данных:

- <https://proglib.io>

- <https://msdn.microsoft.com/ru-ru/>

- <https://docs.microsoft.com/>

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Для проведения лекционных занятий необходима аудитория, оснащенная оборудованием для лекционных демонстраций и проекционной аппаратурой.

Для проведения лабораторных работ необходима лаборатория с ПК, оснащенными программами для проведения лабораторного практикума и обеспечивающими возможность доступа к локальной сети кафедры и Интернет, из следующего перечня:

- 322 (Лаборатория распределённых вычислений)

- 324 (Специализированная лаборатория сетевых систем управления (научно-образовательный центр «АТОС»))

Лаборатории расположены по адресу: 394066, г. Воронеж, Московский проспект, 179 (учебный корпус №3).

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Программирование мобильных платформ» читаются лекции, проводятся лабораторные работы.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Вид учебных занятий	Деятельность студента
Лекция	<p>Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.</p>
Лабораторная работа	<p>Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.</p>
Самостоятельная работа	<p>Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие:</p> <ul style="list-style-type: none"> - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	<p>Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом три дня эффективнее всего использовать для повторения и систематизации материала.</p>

Лист регистрации изменений

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП
1	Актуализирован раздел 8.1 Перечень учебной литературы, необходимой для освоения дисциплины. Актуализирован раздел 8.2 в части состава используемого лицензионного программного обеспечения, современных профессиональных баз данных и справочных информационных систем.	31.08.2020	
2	Внесены изменения в части состава используемого лицензионного программного обеспечения, современных профессиональных баз данных и справочных информационных систем, учебной литературы, необходимой для освоения дисциплины.	31.08.2021	