АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ СОЦИАЛЬНЫЙ ИНСТИТУТ»

Утверждаю
Декан факультета
_____Ж.В. Игнатенко
«19» мая 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Разработка мобильных приложений

Направление подготовки: 09.03.02 Информационные системы и технологии

Направленность (профиль) программы: Проектирование информационных систем и их компонентов

Квалификация выпускника: Бакалавр

Форма обучения очная, заочная

год начала подготовки – 2025

Разработана
Канд. эконом. наук, доцент
_____ А.С. Березницкий

Согласована
зав. кафедрой ИС
_____А.Ю. Орлова

Рекомендована
на заседании кафедры ИС
от «19» мая 2025г.
протокол № 9
Зав. кафедрой _____ А.Ю. Орлова

Одобрена
на заседании учебно-методической
комиссии факультета
от «19» мая 2025 г.
протокол № 9
Председатель УМК
_____ Ж.В. Игнатенко

Ставрополь, 2025 г.

Содержание

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целями освоения дисциплины «Разработка мобильных приложений»являются:формирование представлений о современных технологиях программирования приложений для мобильных устройств, формирование набора профессиональных компетенций будущего бакалавра по направлению подготовки.

Задачи при изучении дисциплины:

1. Изучение базового устройства популярных мобильных платформ; изучение основных этапов жизненного цикла информационной системы для мобильных устройств.

2. Изучение технологии выбора современных операционных сред и информационно-коммуникационных технологий при проектировании, конструировании и отладке программных средств для мобильных устройств.

3. Овладение практическими навыками анализа рынка программно-технических средств, информационных продуктов и услуг для решения прикладных задач и создания информационных систем для мобильных устройств.

4. Получение практических навыков программирования, внедрения, адаптации и настройке мобильных гаджетов, пользовательских интерфейсов и сервисов под OS Android и WindowsPhone.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Разработка мобильных приложений» входит в Блок 1(Б.1.ДВ.2) «Дисциплины (модули)», часть, формируемую участниками образовательных отношений – обязательные дисциплины.

Предшествующие дисциплины (курсы, модули, практики)	Последующие дисциплины (курсы, модули, практики)
Безопасность информационных систем Технологии программирования Интеллектуальные информационные системы и технологии	

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Код и наименование компетенции	**Код и наименование индикатора (индикаторов) достижения компетенции**	**Результаты обучения**
ПК-8 Владение навыками использования различных технологий разработки программного обеспечения	ПК-8.1. Определяет формальные методы конструирования программного обеспечения	**Знает** методы конструирования программного обеспечения **Умеет** определять формальные методы конструирования программного обеспечения **Владеет навыками** конструирования программного обеспечения

	ПК-8.2. Выполняет работы и управляет работами по формализации и моделированию программного обеспечения	**Знает** методы управления работами по формализации и моделированию программного обеспечения **Умеет** выполнять работы и управляет работами по формализации и моделированию программного обеспечения **Владеет навыками** работы и управляет работами по формализации и моделированию программного обеспечения

4. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Общий объем дисциплины составляет 4 зачетных единиц,144 академических часа.

Вид учебной работы	Всего часов		Триместр	
			9	А
	ОФО	ЗФО	ОФО	ЗФО
Контактная работа (всего)	**40,2**	**16,2**	**40,2**	**16,2**
в том числе:				
1) занятия лекционного типа (ЛК)	**20**	**6**	**20**	**6**
из них				
-лекций	20	6	20	6
2) занятия семинарского типа (ПЗ)	**20**	**10**	**20**	**10**
-семинары (С)				
-практические занятия (ПР)	20	10	20	10
-лабораторные работы (ЛР)				
3) групповые консультации				
4) индивидуальная работа				
5) промежуточная аттестация	**0,2**	**0,2**	**0,2**	**0,2**
Самостоятельная работа (всего) (СР)	**103,8**	**123,8**	**103,8**	**127,8**
в том числе:				
Курсовой проект (работа)				
Расчетно-графические работы				
Контрольная работа				
Реферат				
Самоподготовка (самостоятельное изучение разделов, лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, контролю и т.д.)	100	120	100	120
Вид промежуточной аттестации (зачет)	3,8	3,8	3,8	3,8
Общий объем, час	**144**	**144**	**144**	**144**

5. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

5.1. Содержаниедисциплины

№ раздела (темы)	Наименование раздела (темы)	Содержание раздела (темы)
1.	История развития и современное состояние мобильных устройств	Общие сведения от истории развития мобильных устройств.

2.	Коммуникационные технологии	Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова. Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция. Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.
3.	Мобильные ОС	Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.
4.	Разработка мобильных приложений	Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления.
5.	Инструментальные средства программирования	Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций. Рекурсивные функции. Вычисление факториала. Числа Фибоначчи.
6.	Введение в мобильное программирование	Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.
7.	Введение в разработку Android-приложений	Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач. Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы. Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.
8.	Введение в разработку приложений для устройств на платформе iOS	Стиль программирования. Отладка программ.

5.2. Структурадисциплины

№	Наименование раздела		Количество часов

раздела (темы)	(темы)	Всего*	Л		ЛР		К		СР	
			ОФО	ЗФО	ОФО	ЗФО	ОФО	ЗФО	ОФО	ЗФО
1	История развития и современное состояние мобильных устройств	14/13	2	-	-	1	-	-	12	12
2	Коммуникационные технологии	16/14	2	1	2	1	-	-	12	12
3	Мобильные ОС	18/16	4	1	2	1	-	-	12	14
4	Разработка мобильных приложений	22/16	4	1	6	1	-	-	12	14
5	Инструментальные средства программирования	16/21	2	1	4	2	-	-	12	18
6	Введение в мобильное программирование	14/19	2	-	-	1	-	-	12	18
7	Введение в разработку Android-приложений	18/21	2	1	4	2	-	-	12	18
8	Введение в разработку приложений для устройств на платформе iOS	20/20	2	1	2	1	-	-	16	18
	Реферат	-	-	-	-	-	-	-	-	-
	Зачет	4/4	-	-			-	-	-	-
	Общий объем	**144/144**	**20**	**6**	**20**	**10**	**-**	**-**	**100**	**124**

5.3. Занятиясеминарскоготипа

№ п/п	№ раздела (темы)	Вид занятия	Наименование	Количество часов	
				ОФО	ЗФО
1	2	ЛР	Особенности интерфейсов для смартфонов. Принципы юзабилити	2	1
2	2	ЛР	Основы тестирования и отладки приложений на смартфоне	2	1
3	4	ЛР	Портирование приложений с использованием Intel XDK	2	1
4	4	ЛР	Использование мобильной связи в приложениях для смартфона	2	1
5	4	ЛР	Управление воспроизведением аудио	2	-
6	5	ЛР	Установка и настройка среды программирования ADT Bundle	4	1
7	7	ЛР	Основы разработки интерфейсов мобильных приложений	2	-
8	7	ЛР	Создание многоэкранного приложения	2	1
9	8	ЛР	Демонстрации распознавания стандартных жестов	2	-

5.4. Курсовой проект (курсовая работа, расчетно-графическая работа, реферат, контрольная работа)

5.5. Самостоятельная работа

№ темы	Виды самостоятельной работы	Количество часов	
		ОФО	ЗФО
1	Изучение источников информации по теме. Подготовка к лабораторной работе	12	12
2	Изучение источников информации по теме. Подготовка к лабораторной работе	12	12
3	Изучение источников информации по теме. Подготовка к лабораторной работе	12	14
4	Изучение источников информации по теме. Подготовка к лабораторной работе	12	14
5	Изучение источников информации по теме. Подготовка к лабораторной работе	12	18
6	Изучение источников информации по теме. Подготовка к лабораторной работе	12	18
7	Изучение источников информации по теме. Подготовка к лабораторной работе	12	18
8	Изучение источников информации по теме. Подготовка к лабораторной работе	16	18
	Реферат	-	-
	Подготовка к аттестации	4	4

6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Основные технологии обучения:

– работа с правовой информацией, в том числе с использованием современных компьютерных технологий, ресурсов сети Интернет;
– работа с текстами учебника, дополнительной литературой;
– работа с таблицами, схемами;
– выполнение тестовых заданий по темам;
– участие в дискуссиях;
– работа с документами.

Информационные технологии, используемые при осуществлении образовательного процесса по дисциплине:

– сбор, хранение, систематизация, обработка и представление учебной и научной информации;
– обработка различного рода информации с применением современных информационных технологий;
– самостоятельный поиск дополнительного учебного и научного материала, с использованием поисковых систем и сайтов сети Интернет, электронных энциклопедий и баз данных;
– использование электронной почты для рассылки и асинхронного общения, чата преподавателей и обучающихся, переписки и обсуждения возникших учебных проблем для синхронного взаимодействия;
– использование дистанционных образовательных технологий (при необходимости).

При чтении лекций используется компьютерная техника для демонстрации слайдов с помощью программного приложения MicrosoftPowerPoint. На практических занятиях студенты представляют результаты выполнения самостоятельной работы, подготовленные с помощью

программного продукта MicrosoftWord. При выполнении практических заданий на практических занятиях, студентами используется программное обеспечение: Windows 7, MicrosoftOffice.

Интерактивные и активные образовательные технологии

№ раздела (темы)	Вид занятия (Л, ПЗ, С, ЛР)	Используемые интерактивные образовательные технологии	Количество часов	
			ОФО	ЗФО
2	Л	Лекция-дискуссия	2	1
3	ЛР	Работа малыми группами	2	1
4	Л	Проблемная лекция.	2	1
5	ЛР	Работа малыми группами	2	1

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Описание показателей оценивания компетенций, формируемых в процессе освоения дисциплины (модуля), и используемые оценочные средства приведены в таблице 1.

Таблица 1 – Показатели оценивания и оценочные средствадля оценивания результатов обучения по дисциплине

Код и наименование формируемой компетенции	Код и наименование индикатора достижения формируемой компетенции	Показатели оценивания (результаты обучения)	Процедуры оценивания (оценочные средства)	
			текущий контроль успеваемости	промежуточная аттестация
ПК-8 Владение навыками использования различных технологий разработки программного обеспечения т	ПК-8.1. Определяет формальные методы конструирования программного обеспечения	**Знает** методы конструирования программного обеспечения	Контрольные вопросы Тестовоезадание	зачет (контрольные вопросы, тестовое задание)
		Умеет определять формальные методы конструирования программного обеспечения	Практическоезадание	зачет(ситуационнаязадача)
		Владеет навыками конструирования программного обеспечения	Практическоезадание	зачет(ситуационнаязадача)
	ПК-8.2. Выполняет работы и управляет работами по формализации и моделированию программного	**Знает** методы управления работами по формализации и моделированию программного обеспечения	Контрольные вопросы Тестовоезадание	зачет (контрольные вопросы, тестовое

Код и наименование формируемой компетенции	Код и наименование индикатора достижения формируемой компетенции	Показатели оценивания (результаты обучения)	Процедуры оценивания (оценочные средства)	
			текущий контроль успеваемости	промежуточная аттестация
	обеспечения			задание)
		Умеет выполнять работы и управляет работами по формализации и моделированию программного обеспечения	Практическоезадание	зачет(ситуационнаязадача)
		Владеет навыками работы и управляет работами по формализации и моделированию программного обеспечения	Практическоезадание	зачет(ситуационнаязадача)
ПК-8				зачет

7.1. ОЦЕНОЧНЫЕ СРЕДСТВА, КРИТЕРИИ И ШКАЛА ОЦЕНКИ

Типовые задания для текущего контроля успеваемости

Перечень типовых контрольных вопросов для подготовки к устному опросу

Устные опросы проводятся во время лекций, практических занятий и возможны при проведении промежуточной аттестации в качестве дополнительного испытания при недостаточности результатов тестирования. Основные вопросы для устного опроса доводятся до сведения обучающихся на предыдущем занятии.

Развернутый ответ обучающегося должен представлять собой связное, логически последовательное сообщение на заданную тему, показывать его умение применять определения, правила в конкретных случаях.

1. Как во время работы программы изменить цвет текста?
2. Что нужно сделать, чтобы при выборе пользователем текстового поля выводилась специальная клавиатура для ввода чисел?
3. Как при создании программы можно изменить свойства элемента Silverlight?
4. Каковы аппаратные требования к устройствам WindowsPhone?
5. В чём преимущество использования сенсорного экрана емкостного типа по сравнению с резистивным?
6. Какие устройства WindowsPhone позволяют телефону определять своё местоположение?

7. Какие аппартные кнопки есть у устройств WindowsPhone и какие функции они выполняют?

8. Какие типы сетевых подключений поддерживаются в WindowsPhone?

9. С какими программами и службами может взаимодействовать WindowsPhone? В чём идея "быстрого переключения приложений"?

10. Каковы отличия фоновых задач от программ?

11. Как выполняется компиляция и запуск программы в WindowsPhone? Для чего используется эмулятор WindowsPhone?

12. Как приложение может использовать функции телефона? Для чего используются технологии Silverlight и XNA?

13. Какие средства для хранения данных есть в WindowsPhone?

14. Какие инструменты можно использовать для создания приложений для WindowsPhone?

15. Длячегонужен Windows Phone Marketplace?

16. Какие средства разработки можно использовать для создания интерфейса приложений?

17. Что такое Metro-стиль?

18. Для чего предназначены свойства и методы?

19. Как создать интерфейс страницы приложения Silverlight?

20. Как можно изменить внешний вид элементов Silverlight на странице приложения? Для чего используется язык XAML?

21. Как обрабатывать события в приложении Silverlight?

22. Как можно создать на основе файлов исходного кода программы исполняемый

23. Как в VisualStudio создать новую программу? Что такое "пространство имен"?

24. Как совместно использовать несколько проектов? Каким образом можно добавить в проект ресурсы? Что такое "динамическая библиотека"?

25. Что такое "построение программы"?

26. Чем отличается термин "решение" от термина "проект"?

27. Какие типы решений и проектов приложений для WindowsPhone поставляются cWindowsPhone SDK?

28. Что представляют собой файлы с расширением .xap?

29. Как в VisualStudio запустить отладку приложения в эмуляторе WindowsPhone? Какие средства VisualStudio позволяют осуществлять отладку приложений?

30. Как в программе можно обработать ошибку, если пользователь введет вместо числа произвольный текст?

Критерии и шкала оценивания устного опроса

отлично	1) студент полно излагает материал, дает правильное определение основных понятий; 2) обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные; 3) излагает материал последовательно и правильно с точки зрения норм литературного языка.
хорошо	студент дает ответ, удовлетворяющий тем же требованиям, что и для отметки, но допускает 1–2 ошибки, которые сам же исправляет, и 1–2 недочета в последовательности и языковом оформлении излагаемого.
удовлетворительно	студент обнаруживает знание и понимание основных положений данной темы, но: 1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил; 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры;

	3) излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.
неудовлетворительно	студент обнаруживает незнание большей части соответствующего вопроса, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал. Оценка «неудовлетворительно» отмечает такие недостатки в подготовке, которые являются серьезным препятствием к успешному овладению последующим материалом.

Типовые практические задания

Общие сведения о платформе WindowsPhone 7.5

Ваша компания планирует создать приложение для адвокатов. Приложение должно хранить и управлять информацией о времени, которое тратит их персонал на работу с клиентами. Менеджер вашей компании побеседовал с потенциальными покупателями продукта и предоставил следующую информацию.

Система будет использоваться юридическим штатом, чтобы поминутно отслеживать действия сотрудников.

Первоочередные задачи системы — предоставление актуальной информации, её надёжное хранение и простота использования.

Программа будет подключаться к нашим серверам тайм-менеджмента и загружать расписания работы и информацию о клиентах в телефон в начале каждого дня.

В течение рабочего дня пользователи будут вводить информацию о своих действиях, и в конце дня телефон должен загружать эту информацию обратно на сервер.

Периодически будут возникать неотложные случаи, которым необходимо уделять внимание, и наша система должна отправлять сообщение сотрудникам о возникновении подобных ситуаций.

Иногда персонал будет участвовать во встречах за пределами офиса, и при этом система также должна отслеживать их действия.

Ваша компания планирует использовать в телефонах ещё несколько приложений, включая юридический словарь и интерактивную юридическую систему. Новое приложение должно работать одновременно с этими программами.

Мы также хотели бы использовать телефонные устройства в качестве диктофона, чтобы делать аудио- и видеозаписи встреч с клиентами с высоким качеством. К записям будут добавляться теги, и они будут загружаться на наши сервера. Записи будут занимать примерно 100 Мб данных за один час.

Наше приложение должно работать совместно с другими приложениями в устройстве, которые клиенты могут загрузить.

Компания иногда должна отправлять информационное "сообщение дня" с корпоративного веб-сайта. Приложение должно принимать и выводить эти сообщения на экран. В идеале, это должно происходить, даже когда приложение не запущено.

В будущем может понадобиться отслеживать местоположение тех сотрудников, которые выезжают за пределы офиса.

Ваш менеджер попросил вас исследовать платформу WindowsPhone как потенциальное устройство для выполнения поставленных задач. Вы должны предоставить ответы на следующие вопросы:

Могут ли возникнуть какие-либо проблемы при использовании телефона для решения этих задач?

На основе какой технологии следует создать приложение: Silverlight или XNA?

Какой тип сетевого подключения должен использоваться для распределения расписания и получения отчётов?

Следует ли ограничить владельцев телефонов возможностью запуска только одного нашего приложения?

Достаточно ли в телефоне памяти для хранения данных для обслуживания 10 встреч в день?

Есть ли в WindowsPhone какие-либо особенности, которые можно использовать для дальнейшего улучшения приложения?

Есть ли у платформы какие-либо ограничения, которые в будущем могут привести к изменениям технических характеристик устройств?

Какой язык программирования вы предлагаете использовать для разработки приложения? Можно ли использовать некоторые библиотеки для обработки данных, которые некоторое время назад были написаны на VisualBasic .NET?

Есть ли какие-то особые требования для создания и выполнения фоновых задач в телефоне?

Можно ли передать какие-то операции в облачный сервер?

При ответе на каждый вопрос вы должны определить подходящие средства и возможности платформы WindowsPhone.

Введение в Silverlight

1. Пользовательский интерфейс программы Калькулятор времени Ваша компания решила создать приложение для адвокатов. Приложение должно

хранить и управлять информацией о времени, которое тратит их персонал на работу с клиентами. В настоящее время адвокаты записывают в отчет количество минут, которые они потратили на работу с клиентом, и эти отчеты обрабатываются вручную. После того как адвокатам выдали телефоны на платформе WindowsPhone, было решено создать простой Калькулятор времени, который они смогут использовать для расчета времени.

Адвокаты будут вводить время запуска и время окончания (в часах и минутах), и программа должна выводит на экран количество минут между этими отметками времени.

Ваш менеджер попросил Вас разработать пользовательский интерфейс Silverlight для этого приложения:

Определите, какие элементы интерфейса Silverlight нужно разместить на главной странице приложения.

Нарисуйте схему расположения элементов на форме. Создайте "Руководство пользователя" для приложения.

Попытайтесь определить все возможные случаи ввода недопустимых значений. Добавьте описание действий приложения в случае ввода недопустимых данных.

Создайте набор тестовых данных и результатов, которые должна выдать программа, и оформите их в виде таблицы. Количество тестов должно быть не менее 10.

2. Пользовательский интерфейс калькулятора времени

Один из менеджеров по продукции считает, что телефон должен отправлять детализированную информацию о каждом временном интервале в корпоративную систему так, чтобы информация о проведенном времени могла обновляться автоматически. Менеджер решил, что нужно отправлять следующую информацию:

дата и время начала работы; дата и время окончания работы; название компании-клиента;

местоположение клиента, если встреча происходит не в помещении компании клиента;

оценка качества работы, поставленная клиентом — целое число в диапазоне от одного до пяти.

На сервер нужно отправлять отчеты, содержащие указанную информацию о каждой рабочей встрече.

Создайте пример XML-файла, содержащий описания трех встреч.

Удостоверьтесь, что файл содержит записи о встречах в помещении компании- заказчика, а также записи о встречах за пределами компании.

Управление решениями в VisualStudio

1. Создание приложения Калькулятор времени

Создание пустого приложения Silverlight

Запустите VisualStudio с установленным WindowsPhone SDK.

В главном меню выберите пункт Файл -> Создать проект…. Откроется диалоговое окно Создать проект.

В списке установленных шаблонов выберите SilverlightforWindowsPhone. В центральной части окна выберите Приложение WindowsPhone.

В нижней части окна в поле Имя введите TimeCalculator.

Выберите подходящий каталог для проекта. По умолчанию VisualStudio поместит проект в папку VisualStudio 2010 \ Projects в папке Документы.

Убедитесь, что флажок Создать каталог для решения установлен, и нажмите OK, чтобы создать новое приложение.

VisualStudio попросит выбрать целевую платформу для нового приложения. Выберите ОС WindowsPhone 7.1 и нажмите OK. VisualStudio создаст новое приложение в указанном местоположении.

Нажмите клавишу F5, чтобы запустить пустую программу. Запустится эмулятор WindowsPhone (если он еще не был запущен), и приложение будет выполняться в эмуляторе.

Щелкните по окну VisualStudio, чтобы его выбрать. Нажмите клавиши Shift + F5, чтобы остановить выполнение программы. Также можно остановить программу, нажав кнопку Назад в эмуляторе WindowsPhone.

Добавление элементов Silverlight

Убедитесь, что в VisualStudio открыто окно MainPage.xaml. Слева находится окно дизайнера (оно похоже на WindowsPhone), в центре — код XAML, справа — обозреватель решений. (Указанные области могут располагаться в окне VisualStudio по- другому.)

Измените текст имя страницы в элементе PageTitle на TimeCalculator. Измените текст МОЕ ПРИЛОЖЕНИЕ в элементе ApplicationTitle на ваше имя.

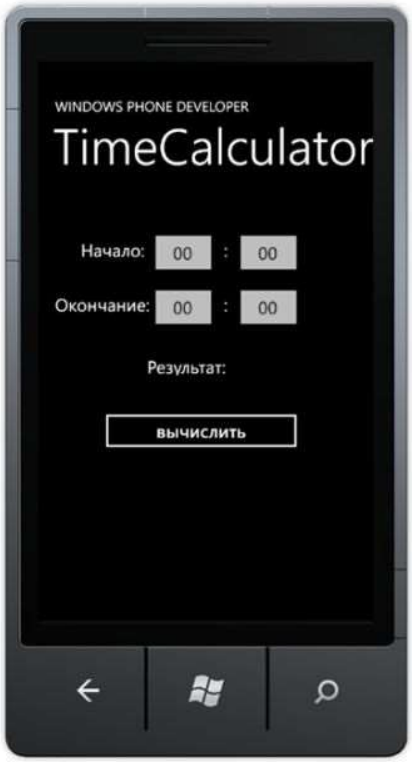
В главном меню выберите пункт Вид -> Панель элементов, чтобы отобразить на экране панель инструментов с доступными элементами. Элементы можно перетаскивать с панели инструментов в окно дизайнера.

Добавьте следующие элементы пользовательского интерфейса в окно дизайнера приложения и задайте их имена, как указано в таблице:

Назначениеэлемента	Типэлемента	Имяэлемента
Часы времени начала TextBoxstartHourTextBox Минуты времени начала TextBoxstartMinTextBox Часы времени окончания TextBoxendHourTextBox Минуты времени окончания TextBoxendMinTextBox Кнопка для вычисления Кнопка calcButton РезультатTextBlockresultTextBlock		

Можно также добавить дополнительные элементы TextBlock, чтобы разместить в окне приложения пояснительный текст.

Настройте свойства добавленных элементов, чтобы приложение выглядело приблизительно таким образом (рекомендуется установить размер текста элементов TextBlock равным 25):



Запустите программу. Обратите внимание, что можно вводить текст в текстовые поля и нажимать на кнопку.

Добавление поведения в приложение

В обозревателе решений откройте файл MainPage.xaml.cs.

Добавьте после конструктора MainPage следующее описание метода calculateMinutes:

```
privatevoidcalculateMinutes()
{
// здесь нужно будет добавить проверку данных и обработку исключений

int startMin = int.Parse(startMinTextBox.Text); int startHour = int.Parse(startHourTextBox.Text);

int endMin = int.Parse(endMinTextBox.Text); int endHour = int.Parse(endHourTextBox.Text);

// здесь нужно будет добавить код для вычисления времени

int result = 99;

resultTextBlock.Text = "Результат: " + result.ToString();
}
```

Вернитесь в окно редактора файла MainPage.xaml.

Дважды щелкните мышью по кнопке вычислить. VisualStudio создаст обработчик нажатия на кнопку и откроет код обработчика, созданный в файле MainPage.xaml.cs.

```
private void calcButton_Click(object sender, RoutedEventArgs e)
{
}
```

Добавьте вызов метода calculateMinutes в этот метод-обработчик.

Запустите программу. Обратите внимание, что при нажатии на кнопку значение результата изменяется на 99.

Завершение создания приложения

В редакторе кода XAML добавьте во все текстовые поля следующий атрибут: InputScope="Number"

Это укажет WindowsPhone, что в текстовые поля будут вводиться числа.

Добавьте в метод calculateMinutes код для вычисления разности времени между введенными моментами времени (создайте код самостоятельно).

Запустите программу и проверьте правильность ее работы. Упражнение 2. Отладка приложения

В этом упражнении вы выполните отладку приложения Калькулятор времени, которое создал другой программист. Программа иногда выдает неправильные результаты. Необходимо выполнить отладку программы, чтобы обнаружить проблему.

Откройте проект TimeCalculator, который находится в папке Lab3 TimeCalculator. Выполните программу. Введите значения Начало: 00:00 и Окончание: 01:00.

Нажмите кнопку вычислить. Обратите внимание, что программа выводит на экран 60 минут, что является правильным результатом.

Введите значения Начало: 00:00 и Окончание: 01:30.

Нажмите кнопку вычислить. Обратите внимание, что программа выводит на экран 60 минут, что является неверным результатом.

Во время работы программы откройте исходный файл MainPage.xaml.cs в обозревателе решений.

Установите точку останова в следующей строке, щелкнув в поле слева от этой строки:

resultTextBlock.Text = "Результат: " + result.ToString();

Нажмите кнопку вычислить еще раз. Теперь программа должна приостановить работу в этой строке.

Просмотрите значения переменных. Попробуйте установить причину возникающей проблемы. Если вы не можете обнаружить ошибку, выполните следующий шаг.

Нажмите клавишу F5 для возобновления программы. Введите значения Начало: 01:02 и Окончание: 03:04. Теперь используйте отладчик, чтобы просмотреть значения переменных startHour, startMin, endHour и endMin.

Определите и исправьте причину ошибок.

Удалите контрольную точку, щелкнув мышью в левом поле еще раз, и проверьте правильность работы программы на тестовых данных.

Создание приложений Silverlight

1. Добавление обработки ошибок

Существующая версия программа Калькулятор времени работает при вводе правильных числовых значений, но выдает исключение, если пользователь введет значения, лежащие вне диапазона, или произвольный текст, который не является числом. Необходимо добавить в программу код для обработки следующих ошибок:

ввод произвольного текста вместо чисел;

ввод значения часа, которое больше 23 или меньше 0; ввод значения минут, которое больше 59 или меньше 0; ввод времени начала, которое больше времени окончания.

В этом упражнении Вы улучшите версию программы, созданную другим программистом.

Откройтев Visual Studio проектTimeCalculatorвпапке Lab4 TimeCalculator.

Запустите программу. Введите значения Начало: 0a:00 и Окончание: 00:00 и нажмите кнопку вычислить.

Приложение сгенерирует исключение, поскольку значение 0a не может быть преобразовано в число.

Измените код программы, чтобы учесть приведенные выше замечания.

Используйте метод TryParse для преобразования текста в число, чтобы программа могла обнаружить недопустимые значения.

Выделите на экран недопустимые значения красным цветом, а допустимые значения — цветом текста по умолчанию.

Выведите на экран окно с сообщением, если время начала и окончания события недопустимо.

Упражнение 2. Улучшение пользовательского интерфейса

Элементы TextBox в приложении могут генерировать событие TextChanged, когда пользователь вводит новые значения. Связанный с этими событиями код может автоматически обновлять значения на экране.

Добавьте обработчик события TextChanged так, чтобы не нужно было использовать кнопку вычислить.

Удалите кнопку вычислить.

Лучший способ это сделать заключается в том, чтобы создать один метод, который будет вызываться для отображения нового значения результата при изменении любого текстового поля. Этот метод можно использовать в обработчиках элементов TextBox.

Упражнение 3. Использование привязки данных

Другой программист создал класс TimeClass для приложения. Вклассепятьсвойств:

StartHourStartMinuteEndHourEndMinuteMinuteDifference

Вам нужно выполнить привязку данных для связывания элементов на экране с этими свойствами. Первые четыре свойства будут использовать двунаправленную привязку, а пятое свойство является выходным значением, которое используется для вывода на экран результата.

Добавление класса TimeClass в качестве ресурса

Сначала нужно добавить TimeClass в проект так, чтобы его могли использовать элементы Silverlight.

Откройтев Visual Studio проектTimeCalculatorв Lab4 Data Binding TimeCalculator. Откройте страницу MainPage.xaml.

Сначала нужно добавить пространство имен к ресурсам страницы MainPage. После строки, которая начинается на

xmlns:mc = "http://schemas.openxmlformats....добавьте строку

xmlns:local = "clr-namespace:TimeCalculator"

Теперь нужно добавить связь к классом, который мы хотим использовать. Послестрокиshell:SystemTray.IsVisible="True">добавьтестроки

<phone:PhoneApplicationPage.Resources>

<local:TimeClass x:Key="TimeClass" />

</phone:PhoneApplicationPage.Resources>

Теперь можно добавить ресурс к элементу Grid, который содержит элементы

Silverlight пользовательского интерфейса приложения. В элемент Grid с именем LayoutRoot добавьте следующий атрибут:

DataContext = "{StaticResourceTimeClass}" Связывание данных со свойствами элементов

В редакторе VisualStudio щелкните по элементу startHourTextBox. В области свойств будут отображаться свойства этого элемента.

Щелкните правой кнопкой мыши по свойству Text, и выберите в контекстном меню пункт Применить привязку данных…. Появится список доступных свойств.

Дважды щелкните по элементу StartHour. Обратите внимание, что в параметрах задан режим TwoWay.

Повторите эти действия для привязки свойств StartMin, EndHour и Endmin к соответствующим элементам.

В настройках привязки элемента resultTextBlock к свойству MinuteDifference установите режим привязки OneWay.

Запустите программу и введите значение времени окончания. Проверьте, что значение результата обновляется, когда пользователь переходит к другому элементу.

Теперь программа использует привязку данных, но часть обработка ошибок при вводе не выполняется. Попробуйте добавить в программу код для обработки ошибок. Для этого можно добавить в класс TimeClass свойства с сообщениями об ошибке и связать их с визуальными элементами.

Упражнение 4. Связывание данных со списками

Вашему начальнику нужно приложение для работы с информацией о клиентах, которое может выводить на экран список встреч с каждым клиентом. Программист начал писать эту программу, но не закончил ее. Вам поручено дописать программу.

Класс Session

Этот класс содержит текстовое описание встречи и ее продолжительность в минутах.

```
public class Session
{
public string Description { get; set; } public int LengthInMins { get; set; }

public Session(string inDescription, int inLength)
{
Description = inDescription; LengthInMins = inLength;
}
}
```

Класс Customer содержитсписоквстречсклиентом. public List<Session> Sessions;

Программа создает тестовый набор встреч для каждого клиента, но Ваш начальник считает, что этого недостаточно.

Увеличение количества тестовых данных

Для начала нужно изменить метод MakeTestCustomers класса Customers, чтобы создать больше тестовых данных.

Откройтев Visual Studio проектCustomerManagerвпапке Lab4 CustomerManager. ОткройтефайлCustomers.cs.

НайдитевклассеCustomers метод MakeTestCustomers. Этот метод создает тестовый набор клиентов, и для каждого клиента создается несколько тестовых встреч. Количество встреч для каждого клиента определяется случайным образом, но диапазон значений является слишком маленьким.

Найдитестроку

```
intnoOfSessions = sessionRand.Next(1,4);
```

Метод Next возвращает случайное значение из диапазона 1—3, что позволит создать максимум 3 тестовых встречи с одним клиентом.

Измените параметры метода так, чтобы для каждого клиента было создано 20—30 встреч. Это позволит проверить, что выводимый на экран список клиентов можно прокручивать.

Добавление навигации по страницам

Страница с информацией о клиенте теперь содержит кнопку встречи, которая используется для просмотра встреч с клиентом. Но пока кнопке не назначен обработчик событий, и при ее нажатии ничего не происходит. Необходимо создать обработчик нажатия на кнопку и добавить код для перехода к странице встреч.

Откройте в редакторе VisualStudio страницу CustomerDetailPage.xaml.

Дважды щелкните по кнопке встречи, чтобы создать обработчик событий и перейти к файлу кода CustomerDetailPage.xaml.cs.

В созданный метод sessionButton_Click добавьте следующий код для выполнения перехода на страницу.

```
NavigationService.Navigate(new Uri("/SessionDetailPage.xaml", UriKind.RelativeOrAbsolute));
```

Запустите программу. Выберите клиента и нажмите на кнопку встречи. Произойдет переход на страницу встреч, но на экран ничего не будет выводиться, потому что к странице не привязаны никакие данные.

Добавление привязки данных на страницу Sessions

Откройте в VisualStudio файл с исходным кодом SessionDetailPage.xaml.cs.

Переместите курсор в класс сразу после закрывающей фигурной скобки конструктора SessionDetailsPage. Введите слово override и нажмите пробел.

Intellisense выведет на экран список методов, которые могут быть переопределены в этом классе. Выберите метод OnNavigatedTo и нажмите Enter.

Добавьте следующий код в метод OnNavigatedTo.

// получить ссылку на страницу, содержащую информацию о выбранном клиенте AppthisApp = Application.CurrentasApp;

CustomerName.DataContext = thisApp.ActiveCustomer; SessionList.ItemsSource = thisApp.ActiveCustomer.Sessions;

Запустите программу. Выберите клиента и нажмите на кнопку встречи. На экран будет выведен список встреч для выбранного клиента. При нажатии на кнопку Назад произойдет переход на страницу информации о клиенте. Если нажать кнопку Назад еще раз, откроется страница со списком клиентов.

Хранение данных приложений

1. Использование изолированного хранилища

В предыдущем упражнении мы создавали простое приложение TimeTracker, которое сохраняло информацию о времени встреч с клиентами. Предыдущая версия программы при запуске генерировала набор тестовых данных. Необходимо использовать хранилище файлов, чтобы приложение сохраняло информацию о встречах.

Загрузка сохраненных данных

Программист внес в программу изменения, однако, тестировщик обнаружил, что программа не сохраняет изменения данных. Необходимо выявить причину этой проблемы.

Откройте в VisualStudio проект CustomerManager в папке Lab5 CustomerTimeLoggerStorage. Этот проект содержит версию программы, в которой обнаружена ошибка.

Убедитесь в том, что эмулятор WindowsPhone не запущен. Укажите эмулятор WindowsPhone в качестве целевой платформы.

Запустите программу. При первом запуске программа создаст новый набор тестовых данных.

Выберите клиента и измените информацию о нем.

Нажмите кнопку сохранить. Обратите внимание на то, что содержимое экрана изменилось, поскольку в программе используется привязка данных для отображения обновленных значений.

Остановите программу, нажав в эмуляторе WindowsPhone кнопку Назад. Не останавливайте выполнение программы в VisualStudio.

Запустите программу еще раз. При повторном запуске программа должна загрузить список клиентов из изолированного хранилища.

Найдите клиента, информацию о котором вы изменили. Обратите внимание, что сделанные изменения не сохранились.

Список клиентов загружается в файле App.xaml.cs. Откройте этот файл в обозревателе решений VisualStudio и найдите конструктор App. В конце этого метода находится код, который должен загружать данные из изолированного хранилища. Попробуйте определить причину возникающей проблемы.

Конструктор содержит только код, который создает тестовый список при каждом запуске программы:

ActiveCustomerList = Customers.MakeTestCustomers();

Необходимо изменить этот код, чтобы информация о клиентах загружалась из файла. Откройте в файле App.xaml.cs область CustomerManagerValues и найдите методы LoadCustomers и SaveCustomers.

Метод LoadCustomers возвращает список клиентов или значение null, если список не может быть считан. Если метод возвращает null, программа должна создать набор тестовых данных. Замените указанный в шаге 10 код на следующий:

// загрузить список клиентов из файла ActiveCustomerList = LoadCustomers(ListFilename);

// если загрузить список не удалось, создать тестовые данные if (ActiveCustomerList == null)

ActiveCustomerList = Customers.MakeTestCustomers(); Запуститепрограмму. Измените информацию о любом клиенте.

Остановите программу, нажав в эмуляторе WindowsPhone кнопку Назад. Не останавливайте выполнение программы в VisualStudio.

Запустите программу еще раз. Убедитесь в том, что теперь сделанные изменения были сохранены.

Пропадание информации о встрече

Тестировщик нашел в программе еще одну проблему: программа некорректно сохраняет информацию о встрече. Необходимо исследовать эту проблему.

Программа создает объекты для работы с потоками StreamReader и StreamWriter для загрузки и сохранения элементов в изолированном хранилище. Каждый объект может сохранить свое состояние в поток и загрузить информацию из потока. Ниже приведен код методов для сохранения информации в поток и для загрузки информации из потока:

```
public void SaveToStream(StreamWriter output)
{
output.WriteLine(CustomerList.Count);

foreach (Customer c in CustomerList)
{
c.SaveToStream(output);
}
}

public Customers(StreamReader input)
{
CustomerList = new List<Customer>();
int noOfCustomers = int.Parse(input.ReadLine()); for (int i = 0; i<noOfCustomers; i++)
{
CustomerList.Add(new Customer(input));
}
}
```

Этот код выглядит правильно. Необходимо проверить класс Customer, правильно ли он использует эти методы.

Вернитесь в программу и откройте файл Customers.cs. Найдите класс Customers и просмотрите метод SaveToStream: publicvoidSaveToStream(StreamWriteroutput)

```
{
output.WriteLine(Name); output.WriteLine(Address); output.WriteLine(ID);
}
```

Этот метод не содержит код для сохранения информации о встрече. Аналогичная проблема возникает и в методе Customer:

```
public Customer(StreamReader input)
{
Name = input.ReadLine(); Address = input.ReadLine();
ID = int.Parse(input.ReadLine());
int noOfSessions = int.Parse(input.ReadLine());
```

```
}
```

Необходимо использовать тот же шаблон для загрузки и сохранения списка встреч, который использует объект CustomerList для хранения информации о клиентах. Измените методы для сохранения и загрузки, чтобы можно было сохранять информацию о нескольких встречах:

```
public void SaveToStream(StreamWriter output)
{
output.WriteLine(Name); output.WriteLine(Address); output.WriteLine(ID); output.WriteLine(Sessions.Count); foreach (Session session in Sessions)
{
session.SaveToStream(output);
}
}

public Customer(StreamReader input)
{
Name = input.ReadLine(); Address = input.ReadLine();
ID = int.Parse(input.ReadLine());
int noOfSessions = int.Parse(input.ReadLine()); for (int i = 0; i<noOfSessions; i++)
{
Sessions.Add(new Session(input));
}
}
```

Остановите эмулятор WindowsPhone, чтобы при следующем запуске программы она создала новый набор тестовых данных. Запустите программу еще раз.

Остановите программу, нажав в эмуляторе WindowsPhone кнопку Назад. Не останавливайте выполнение программы в VisualStudio.

Запустите программу еще раз.

Выберите любого клиента и просмотрите информацию о встречах, которая должна была быть сохранена.

Упражнение 2. Использование базы данных

В программу, созданную в предыдущем упражнении, внесли изменения, чтобы она могла работать с базой данных. Однако, программа компилируется, но не запускается. Необходимо исправить ошибку.

Откройте в VisualStudio проект CustomerManager в папке Lab5 CustomerTimeLoggerDatabase. Этот проект содержит версию программы, в которой обнаружена ошибка.

Убедитесь в том, что эмулятор WindowsPhone не запущен.

Запустите программу. Программа сгенерирует тестовые данные и попытается их использовать. После этого будет сгенерировано исключение, в котором база данных сообщает, что проблема связана с внешним ключом. Это касается связей между классами Session и Customer. Сообщение об ошибке информирует о том, что невозможно вставить внешний ключ, так как не существует соответствующий первичный ключ.

При создании связи встречи с клиентом запись в таблице встреч должна содержать значение первичного ключа, которое идентифицирует клиента. Однако, если этот ключ еще не был задан, база данных не может добавить запись. При создании связи между этими двумя таблицами необходимо:

добавить встречу к списку встреч клиента;

установить значение customerID для встречи, чтобы идентифицировать клиента для этой встречи.

Код, который выполняет эти действия, выглядит следующим образом: intsessionLength = sessionRand.Next(5,120);

string sessionDesc = "Встреча " + i.ToString(); Session newSession = new Session(); newSession.Description = sessionDesc; newSession.LengthInMins = sessionLength; newSession.SessionCustomer = c; newDB.SessionTable.InsertOnSubmit(newSession); c.Sessions.Add(newSession);

Добавьте этот код в программу и запустите ее повторно. Теперь программа работает правильно и позволяет управлять данными.

Тестирование хранения данных

В настоящее время при каждом запуске программа создает новую базу данных. Измените программу так, чтобы она использовала существующую базу данных и создавала новую, если база данных не существует. Обратите внимание, что новая база данных создается при выполнении следующей строки кода в файле App.xaml.cs:

CustomerDB.MakeTestDB("Data Source=isostore:/Sample.sdf");

Средства WindowsPhone для работы с сетью

Для выполнения необходимо работающее сетевое подключение, а также среда VisualStudio для создания службы WCF.

1. Создание службы TimeTracker

В этом упражнении вы создадите службу, которая получает и хранит отчёты о встречах, получаемые от приложения TimeTrackerClient, работающего на удалённом устройстве. Служба будет предоставлять два метода: один — для получения отчётов о встречах, другой — для предоставления отчётов о встречах.

Создание службы Откройте VisualStudio.

Выберите в главном меню пункт Файл -> Создать проект. Откроется диалоговое окно Создать проект.

В списке Установленные шаблоны выберите в группе WCF шаблон Приложение службы WCF.

Назовите проект TimeTrackerService и нажмите OK, чтобы создать проект. VisualStudio создаст новый проект и откроет файл Service1.svc.cs.

Задайте службе и её интерфейсу более удобные имена. Для этого выберите в обозревателе решений файл IService1.cs, нажмите F2 и введите новое имя ITimeTrackerService.cs. При нажатии клавиши EnterVisualStudio предложит переименовать интерфейс и все ссылки на него, чтобы его имя соответствовало новому имени файла. Нажмите OK для подтверждения.

Аналогичным образом переименуйте файл службы от Service1.svc в TimeTrackerService.svc.

Откройте файл TimeTrackerService.svc.cs и щёлкните правой кнопкой по названию класса Service1. В контекстном меню выберите пункт Рефакторинг -> Переименовать…, укажите в открывшемся окне новое имя класса TimeTrackerService и подтвердите переименование.

Создание интерфейсов методов службы

Наша служба будет содержать два метода: один метод будет использовать удалённый клиент для сохранения отчёта о встрече, а другой будет возвращать список сохранённых встреч. VisualStudio автоматически создаёт некоторые методы, которые мы удалим из проекта.

Откройте файл ITimeTrackerService.cs. Он содержит интерфейс, который описывает методы, предоставляемые службой.

Удалите весь код внутри интерфейса и добавьте следующее описание методов: [OperationContract]

void SaveSession(string employeeName, string companyName, int startHour, int startMin,
int endHour, int endMin);

код:

[OperationContract] string GetSessionList();

УдалитеописаниеклассаCompositeType. Созданиеметодовслужбы

ОткройтефайлTimeTrackerService.svc.cs.

Удалите всё содержимое класса TimeTrackerService и добавьте в него следующий

```
class Session
{
public string EmployeeName{ get; set; } public string CompanyName { get; set; } public int StartHour { get; set; }
public int StartMin{ get; set; } public int EndHour { get; set; } public int EndMin { get; set; }

public override string ToString()
{
return EmployeeName + " " + CompanyName;
}
}

static List<Session> sessions = new List<Session>();
```

Этот класс будет хранить информацию о каждой встрече, а также содержит список встреч, который будет формироваться во время работы службы. В реальном приложении этот список должен заполняться информацией из базы данных.

Добавьте в класс TimeTrackerService следующие методы:

```
public void SaveSession(string employeeName, string companyName, int startHour, int startMin, int endHour, int endMin)
{
sessions.Add(new Session{
EmployeeName = employeeName, CompanyName = companyName, StartHour = startHour,
StartMin = startMin, EndHour = endHour, EndMin = endMin
});
}

public string GetSessionList()
{
StringBuilder result = new StringBuilder();

for (int i = 0; i<sessions.Count; i++)
{
result.AppendLine(sessions[i].ToString());
}
returnresult.ToString();
}
```

Метод SaveSession сохраняет информацию о сеансе на сервере. Метод GetSessionList возвращает сохранённую информацию о встречах в виде многострочного текста.

Развёртывание службы

Выберите в обозревателе решений файл TimeTrackerService.svc и нажмите F5 для запуска проекта. После построения решения откроется окно тестового клиента WCF.

Дважды щёлкните мышью по методу SaveSession(). Введите следующие значения параметров метода:

employeeName — Test Employee companyName — Test Company startHour — 1

startMin — 2

endHour —3

endMin — 4

Нажмите на кнопку Вызвать, чтобы вызвать метод службы с указанными параметрами. Метод вернёт пустое значение.

Дважды щёлкните мышью по методу GetSessionList() и нажмите на кнопку Вызвать. Метод вернёт строку, соответствующую встрече, добавленной в систему при вызове метода SaveSession.

Нажмите клавиши Shift + F5 для остановки службы.

В обозревателе решений щёлкните правой кнопкой мыши по файлу TimeTrackerService.svc и выберите в контекстом меню пункт Просмотр в обозревателе. Обратите внимание на URL службы в адресной строке браузера — это значение понадобится для подключения к службе клиента:

http://localhost:30975/TimeTrackerService.svc

В вашем случае номер порта может отличаться, поскольку VisualStudio генерирует его автоматически.

Закройте браузер и запустите службу в VisualStudio ещё раз. Запущенная служба будет использоваться в следующем упражнении.

Упражнение 2. Создание клиента службы

В этом упражнении мы создадим приложение, которое будет подключаться к созданной в предыдущем упражнении службе и использовать её методы.

Подключение клиентского приложения к службе

Убедитесь в том, что созданная в предыдущем упражнении служба запущена.

Запустите другую копию VisualStudio и откройте в ней проект TimeTrackerClient в папке Lab6 TimeTrackerClient. В текстовые поля пользователь может ввести имя сотрудника, название компании, с которой проводится встреча, а также время начала и окончания встречи. При нажатии на кнопку сохранить, программа должна сохранить информацию о встрече на сервере, и при нажатии на кнопку список встреч — вывести информацио о встречах, полученную от службы.

В обозревателе решений щёлкните правой кнопкой мыши по элементу Ссылки и выберите пункт Добавить ссылку на службу….

В открывшемся диалоговом окне Добавить ссылку на службу в поле Адрес введите URL службы, полученный в предыдущем упражнении в результате запуска службы, и нажмите кнопку Перейти. VisualStudio загрузит описание службы с сервера и выведет на экран информацию о службе.

Откройте описание службы, щёлкнув по стрелке слева от её имени, и выберите имя интерфейса ITimeTrackerService. В поле операции будут отображены имена двух созданных методов службы.

Измените пространство имён на TimeTrackerService и нажмите кнопку OK. VisualStudio добавит в проект ссылку на службу.

Щёлкните правой кнопкой мыши по созданной ссылке на службу TimeTrackerService и выберите пункт Настроить ссылку на службу….

В открывшемся диалоговом окне снимите галочку Повторно использовать типы в сборках, на которые есть ссылки и нажмите OK.

Использование метода службы SaveSession в приложении Откройте файл MainPage.xaml.cs.

Добавьте в начало объявления класса следующую строку: privateTimeTrackerService.TimeTrackerServiceClienttimeTracker;

Добавьте в конструктор класса после вызова метода InitializeComponent следующий код для создания связанного со службой объекта и обработчиков событий службы (для создания обработчиков событий можно использовать Intellisense — в этом случае автоматически будет создана большая часть кода):

```
timeTracker = new TimeTrackerService.TimeTrackerServiceClient();

timeTracker.SaveSessionCompleted +=
new EventHandler<System.ComponentModel.AsyncCompletedEventArgs>(
timeTracker_SaveSessionCompleted);
```

timeTracker.GetSessionListCompleted +=
new EventHandler<TimeTrackerService.GetSessionListCompletedEventArgs>(timeTracker_GetSessionListCompleted);

Добавьте код метода saveButton_Click для асинхронного вызова метода службы и передачи введённых пользователем значений:

```
private void saveButton_Click(object sender, RoutedEventArgs e)
{
int startHour, startMin, endHour, endMin;

if (int.TryParse(startHourTextBox.Text, out startHour) &&int.TryParse(startMinTextBox.Text, out startMin) &&int.TryParse(endHourTextBox.Text, out endHour) &&int.TryParse(endMinTextBox.Text, out endMin))
{
timeTracker.SaveSessionAsync(employeeNameTextBox.Text,
companyNameTextBox.Text, startHour, startMin, endHour, endMin);
}
}
```

Добавьте код метода-обработчика timeTracker_SaveSessionCompleted, который будет вызываться при завершении выполнения метода службы:

```
void timeTracker_SaveSessionCompleted(object sender, System.ComponentModel.AsyncCompletedEventArgs e)
{
if (!e.Cancelled)
{
employeeNameTextBox.Text = "Введитеимясотрудника..."; companyNameTextBox.Text = "Введитеназваниекомпании...";
}
}
```

Если метод службы выполнится успешно, то текст в текстовых полях приложения будет изменён на первоначальный.

Тестирование метода SaveSession

Нажмите F5, чтобы запустить программу в эмуляторе.

Введите в текстовые поля информацию о встрече и нажмите на кнопку сохранить.

Текст в текстовых полях должен измениться на первоначальный. Таким же образом добавьте информацию о второй встрече. Остановите выполнение программы.

Использование метода службы GetSessionList

Добавьте в приложение код метода listButton_click для вызова метода службы: privatevoidlistButton_Click(objectsender, RoutedEventArgs e)

```
{
timeTracker.GetSessionListAsync();
}
```

Добавьте в приложение код метода timeTracker_GetSessionListCompleted для вывода на экран результата выполнения метода службы:

```
void timeTracker_GetSessionListCompleted(object sender, TimeTrackerService.GetSessionListCompletedEventArgs e)
{
if (!e.Cancelled)
{
sessionsTextBlock.Text = e.Result;
}
}
```

Запустите программу и нажмите кнопку список встреч. На экран будет выведена информация о двух введённых встречах.

Создание приложений XNA

Для выполнения рекомендуется использовать физическое устройство WindowsPhone для возможности мультисенсорного ввода и использования акселерометра.

1. Создание многопользовательской игры

В этом упражнении вы доработаете созданную на лекции игру. Ограничение движения игрового объекта

На текущий момент левая платформа может перемещаться за пределы экрана.

Необходимо предотвратить такую возможность.

Откройте в VisualStudio проект PongGame в папке Lab7 PongGame.

Добавьте в метод Draw следующий код, который не позволит левой платформе уйти за пределы экрана:

```
if (lPaddleY< 0)
{
lPaddleY = 0;
}

if (lPaddleY + lPaddleRectangle.Height>GraphicsDevice.Viewport.Height)
{
lPaddleY = GraphicsDevice.Viewport.Height - lPaddleRectangle.Height;
}
```

Добавьте аналогичный код для правой платформы. Запустите игру и проверьте правильность её работы. Добавление возможности играть вдвоём

На текущий момент правая платформа управляется компьютером. Благодаря возможности мультисенсорного ввода, можно добавить в игру возможность управления правой платформы вторым игроком.

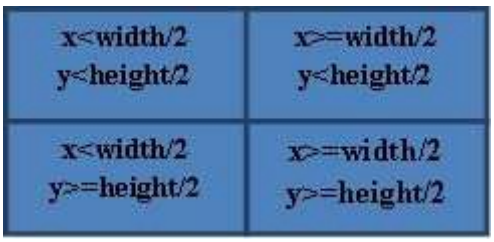
Измените код для управления движением платформы следующим кодом для обработки нескольких одновременных касаний экрана:

```
TouchCollection touches = TouchPanel.GetState();

foreach (TouchLocation touch in touches)
{
if (touch.Position.Y>GraphicsDevice.Viewport.Height / 2)
{
lPaddleY = lPaddleY + lPaddleSpeed;
}
else
{
lPaddleY = lPaddleY - lPaddleSpeed;
}
}
```

Запустите программу и убедитесь в том, что управление платформой работает правильно.

Теперь нужно изменить программный код для управления правой платформой, чтобы ей мог управлять второй игрок. Для этого экран можно поделить на четыре части. При обнаружении касания в каждой из областей определяется направление перемещения соответствующей платформы. Условия принадлежности касания соответствующей области показаны на рисунке:



Измените код программы, чтобы управление обеими платформами осуществлялось с сенсорного экрана. (Решите эту задачу самостоятельно.)

Удалите из программы код, который автоматически управляет правой платформой. Запустите игру и проверьте правильность её работы.

Упражнение 2. Использование акселерометра для управления в игре

В этом упражнении вы создадите приложение, которое использует акселерометр WindowsPhone для определения положения телефона в пространстве. Существующая программа выводит на экран зелёный шарик, расположенный над красной точкой, которая расположена в центре экрана. Шарик должен управляться с помощью акселерометра.

Откройте в VisualStudio проект Level в папке Lab7 Level. Откройте файл Game1.cs.

Добавьте следующий код в конструктор класса Game1 для указания поддерживаемой в приложении ориентации телефона, размеров экрана и задания полноэкранного режима.

graphics.SupportedOrientations = DisplayOrientation.LandscapeLeft; graphics.PreferredBackBufferWidth = 480;

graphics.PreferredBackBufferHeight = 800; graphics.IsFullScreen = true;

Добавьте в проект ссылку на библиотеку Microsoft.Devices.Sensors.

Добавьте в файл Game1.cs ссылку на пространство имён Microsoft.Devices.Sensors.

Добавьте в метод Initialise метод следующий код для создания и запуска объекта для работы с акселерометром:

```
Accelerometer acc = new Accelerometer();

acc.ReadingChanged +=
new EventHandler<AccelerometerReadingEventArgs>(acc_ReadingChanged);

acc.Start();
```

Добавьте сразу после метода Initialise следующий код для определения обработчика события акселерометра:

```
Vector3 accVector = Vector3.Zero;

void acc_ReadingChanged(object sender, AccelerometerReadingEventArgs e)
{
accVector.X = (float)e.X; accVector.Y = (float)e.Y; accVector.Z = (float)e.Z;
}
```

Для управления движением шарика можно использовать значения, получаемые от акселерометра и сохраняемые в переменную accVector. Переменные bubbleX и bubbleY содержат координаты шарика, когда он находится в центре экрана.

В методе Update удалите следующие строки кода: bubbleRectangle.X = (int)(bubbleX + 0.5f); bubbleRectangle.Y = (int)(bubbleY + 0.5f);

и добавьте код для вычисления новых координат шарика, в которые он будет перемещаться:

float accBubbleX = bubbleX - (accVector.X * 200); float accBubbleY = bubbleY + (accVector.Y * 200);

bubbleRectangle.X = (int)(accBubbleX + 0.5f); bubbleRectangle.Y = (int)(accBubbleY + 0.5f);

Запустите программу и проверьте правильность её работы.

Использование системных функций в приложениях

1. Редактирование значков приложения

Создание приложения

Откройте VisualStudio, создайте новое приложение Silverlight и назовите его Icons.

Выберите в качестве целевого устройства эмулятор WindowsPhone и запустите программу, нажав клавишу F5. VisualStudio развернёт программу в эмуляторе и запустит её.

Нажмите в эмуляторе кнопку Назад, чтобы остановить программу. В эмуляторе откроется меню Пуск, содержащее единственную плитку, связанную с программой InternetExplorer.

Щёлкните по стрелке вправо в верхнем правом углу экрана эмулятора. Откроется список приложений эмулятора. Список содержит приложение Icons, которое мы только что создали, InternetExplorer и приложение Настройки эмулятора.

Щёлкните по значку приложения Icons, чтобы запустить это. Приложение запустится в эмуляторе.

Нажмите кнопку Назад, чтобы остановить приложение.

Эмулятор WindowsPhone сохраняет все запускаемые в нём программы в списке приложений, пока его работа не будет остановлена. Можно открыть в VisualStudio другой проект и запустить его в эмуляторе — приложение добавится в список приложений эмулятора.

Прикрепление приложения к меню Пуск Откройте в эмуляторе список приложений.

Нажмите и не отпускайте левую кнопку мыши на приложении Icons. Появится меню, которое позволяет удалить приложение из эмулятора или прикрепить приложение к меню Пуск.

В открывшемся меню выберите пункт на рабочий стол. Приложение Icons будет прикреплено к меню Пуск рядом с плиткой приложения InternetExplorer.

Приложение также можно переместить в другое место меню Пуск или удалить из меню Пуск.

Редактирование значка приложения

В обозревателе решений щёлкните правой кнопкой мыши по файлу ApplicationIcon.png.

В открывшемся контекстном меню выберите пункт Открыть с помощью…, чтобы вывести на экран список программ, которые могут использоваться для работы с выбранным файлом.

Выберите в списке программу Paint и нажмите OK. Откроется окно программы Paint, в котором будет открыто изображение из файла.

Измените изображение и закройте программу Paint, сохранив сделанные изменения.

В главном меню VisualStudio выберите пункт Построение -> Перестроить. VisualStudio заново создаст файлы программы и будет использовать изменённую версию значка.

Запустите программу. VisualStudio удалит старую версию программы с эмулятора и развернёт новую версию с изменённым значком.

Аналогичным образом измените изображения в файлах Background.png и SplashScreenImage.jpg. После повторного построения приложения будут изменены значок в меню Пуск и экран-заставка программы.

2. Исследование выгрузки приложения

В этом упражнении вы рассмотрите, как приложения выгружаются из памяти. Создание приложения

Откройте VisualStudio, создайте новый проект приложения Silverlight и назовите его Tombstone.

Откройте файл App.xaml.cs и найдите методы, которые запускаются, когда происходят события запуска, закрытия, выгрузки и возобновления работы приложения. Добавьте в методы код для выдачи диагностических сообщений:

```
// Код для выполнения при активации приложения (переводится в основной режим)
// Этот код не будет выполняться при первом запуске приложения privatevoidApplication_Activated(objectsender, ActivatedEventArgs e)
{
```

```
System.Diagnostics.Debug.WriteLine("Активировано");
}

// Код для выполнения при деактивации приложения (отправляется в фоновый режим)
// Этот код не будет выполняться при закрытии приложения
private void Application_Deactivated(object sender, DeactivatedEventArgs e)
{
System.Diagnostics.Debug.WriteLine("Деактивировано");
}

// Код для выполнения при закрытии приложения (например, при нажатии пользователем
// кнопки "Назад")
// Этот код не будет выполняться при деактивации приложения privatevoidApplication_Closing(objectsender, ClosingEventArgs e)
{
System.Diagnostics.Debug.WriteLine("Закрывается");
}
```

Выберите в качестве целевого устройства эмулятор WindowsPhone и запустите приложение.

Выберите в главном меню VisualStudio пункт Вид -> Вывод. Откроется окно вывода диагностических сообщений. Последним сообщением будет Запущено.

В эмуляторе щёлкните кнопку Назад, чтобы остановить программу. В окне вывода диагностических сообщений появится сообщение Закрывается.

Деактивация приложения Запустите программу в эмуляторе.

Нажмите в эмуляторе кнопку Пуск для перехода к экрану меню Пуск. В окне вывода появится сообщение Деактивировано.

Запустите в эмуляторе программу InternetExplorer.

Нажмите в эмуляторе кнопку Назад, чтобы вернуться на экран меню Пуск.

Нажмите в эмуляторе кнопку Назад ещё раз, чтобы вернуться на страницу приложения. В окне вывода появится сообщение Активировано.

Закройте приложение. Возобновление работы приложения Откройте файл App.xaml.cs.

Найдите метод Application_Activated и измените его следующим образом, чтобы по выводимому сообщению можно было определить, из какого состояния приложение возобновляет свою работу:

```
private void Application_Activated(object sender, ActivatedEventArgs e)
{
if (e.IsApplicationInstancePreserved)
{
System.Diagnostics.Debug.WriteLine("АктивированоизсостоянияБездействует");
}
else
{
System.Diagnostics.Debug.WriteLine("АктивированоизсостоянияВыгружено");
}
}
```

Выполните все действия предыдущей части упражнения. Обратите внимание, что приложение было активировано из состояния "бездействует" не выгружалось из памяти.

Выгрузка приложения из памяти

В VisualStudio можно настроить, чтобы приложение при деактивации автоматически выгружалось из памяти.

Убедитесь в том, что приложение не запущено.

В обозревателе решений щёлкните правой кнопкой мыши по файлу проекта и в открывшемся меню выберите пункт Свойства. Откроется окно свойств проекта.

Выберите в окне свойств вкладку Отладка и установите галочку Отметить как удаленный в результате деактивации во время.

Выполните пункт 3 предыдущей части упражнения. Обратите внимание, что приложение было активировано из состояния "выгружено".

Наиболее вероятно, что работа приложения будет возобновлена из состояния покоя. Однако, необходимо проверять поведение программы, если она всё же будет выгружена из памяти.

Публикация приложений в WindowsPhoneMarketplace

1. Использование инструмента для анализа производительности

Откройте в VisualStudio один из созданных ранее проектов приложения Silverlight или XNA.

В главном меню VisualStudio выберите пункт Отладка -> Начать анализ производительности WindowsPhone. Откроется окно параметров.

Выберите необходимые параметры или оставьте значения по умолчанию.

Выберите в качестве целевого устройства эмулятор WindowsPhone и нажмите ссылку Запустить приложение. Приложение запустится в эмуляторе, и VisualStudio начнёт собирать сведения о работе приложения.

Выполните в приложении какие-нибудь операции и закройте его, нажав в эмуляторе кнопку Назад, находясь в главном окне приложения. VisualStudio закончит сбор данных и откроет созданный файл с расширением .sap в виде диаграммы.

Проанализируйте диаграмму. Обратите внимание, в какие моменты были зафиксированы всплески активности приложения, и вспомните, какие действия вы выполняли в это время.

Упражнение 2. Использование инструмента MarketplaceTestKit

В этом упражнении вы выполните тестирование приложения с помощью инструмента MarketplaceTestKit.

Откройте в VisualStudio один из созданных ранее проектов приложения Silverlight или XNA.

Выберите в раскрывающемся списке правее от выбора целевой платформы пункт Release.

Выполните построение проекта, нажав кливишиCtrl + Shift + B.

В главном меню VisualStudio выберите пункт Проект -> Открыть в MarketplaceTestKit. Откроется окно инструмента MarketplaceTestKit.

На вкладке Автоматизированные тесты нажмите на кнопку Запуск тестов.

Просмотрите результаты теста. Попробуйте определить, почему приложение не прошло некоторые тесты, и попытайтесь устранить недостатки.

Типовые задания для промежуточного контроля

Перечень типовых контрольных вопросов для устного опроса на промежуточной аттестации (экзамен)

1. Как вывести на экран сообщение, чтобы приложение могло получить ответ от пользователя?
2. Какие настройки можно задать при добавлении в проект ресурса? В чем разница между элементом контента и внедренным ресурсом?
3. Что нужно сделать, чтобы добавить к программу код, который должен выполняться, когда происходит определенное событие?
4. Для чего используется привязка данных?
5. Как выполнить привязку объекта данных к визуальному элементу Silverlight?
6. В чем разница между однонаправленной и двунаправленной привязкой данных?

Как можно указать поддерживаемый страницей приложения тип ориентации?

7. Для чего предназначены объекты-контейнеры? Как можно вывести в приложение список данных?

8. Как осуществить переход к другой странице приложения? Как можно передать данные другой странице приложения? Как создать и использовать в приложении класс ViewModel? Для чего используются наблюдаемые коллекции?

9. Где приложение может сохранять данные?

10. Каким образом программа может сохранять файлы?

11. Как программа может сохранить настройки, которые можно представить в виде пар "имя—значение"?

12. Для чего предназначена программа IsolatedStorageExplorer?

13. Как можно создать базу данных в приложении для WindowsPhone?

14. Как связать результаты запроса LINQ с визуальными элементами Silverlight?

15. Как с помощью LINQ задать связи между таблицами?

16. Как выполняются запросы LINQ?

17. Как выполняется добавление и удаление записей таблиц в LINQ?

18. Какие типы подключений к сети доступны в WindowsPhone?

19. Какие образом данные передаются по сети?

20. Для чего используется система доменных имён?

21. Каким образом несколько взаимодействующих с сетью служб могут работать на одном сервере только с теми сообщениями, которые для них предназначены?

22. В чём отличие сеансов подключений от дейтаграмм?

23. Для чего используется класс Socket?

24. Как в программе можно создать подключение по протоколу UDP?

25. Как в программе можно создать подключение по протоколу TCP?

26. Для чего используется класс WebClient?

27. Как получить необходимые данные из XML-структуры с помощью LINQ?

28. Для чего предназначена технология WindowsCommunicationFoundation? Как создать службу WCF?

29. Как создать приложение для WindowsPhone для использования методов службы WCF?

30. Для чего предназначена технология XNA? В чём отличие решений XNA от Silverlight?

31. Какие действия выполняются при загрузке игр XNA? Как в программе используется контент-менеджер?

32. Как в программе создать и вывести на экран графический объект?

33. Как можно использовать сенсорный экран для управления игровыми объектами? Какие действия необходимо выполнить для вывода текста на экран приложения? Каково назначение и принцип работы акселерометра?

34. Как использовать акселерометр в приложениях для WindowsPhone? Какие классы XNA используются для воспроизведения звуков в игре? Как управлять воспроизведением звука в игре?

35. Как воспроизводить звук в программе Silverlight?

36. Как в приложении можно управлять настройками экрана?

37. Каким образом можно совместно использовать технологии Silverlight и XNA в одном приложении?

38. Какие изображения должно содержать приложение для WindowsPhone, и как эти изображения используются?

39. Как можно создать и использовать в приложении экран-заставку?

40. Что означает "быстрое переключение приложений"?

41. Как происходит переключение приложений?

42. Как можно сохранить состояние приложения при его деактивации и загрузить при

возобновлении работы?

43. Как в приложении вызвать системные функции телефона?
44. Чем отличаются задачи запуска и задачи выбора?
45. Как приложение может использовать фоновые задачи?
46. В чём разница между периодическими и ресурсоёмкими задачами? Как создать и использовать в приложении агента фоновой задачи?
47. Какие возможности есть у инструмента для анализа производительности приложений для WindowsPhone?
48. Для чего нужны файлы манифеста?
49. Какие значки и изображения необходимо подготовить перед распространением приложения через WindowsPhoneMarketplace?
50. Для чего нужна разблокировка телефона?
51. Как проходит процесс одобрения приложения?
52. Для чего предназначен тестовый режим приложения?
53. Как организовать закрытое бета-тестирование приложения?

Ситуационные задачи для промежуточной аттестации

С помощью Быстрого поиска в «Разработка приложений для WindowsPhone 7 [Электронный ресурс]. -http://msdn.microsoft.com/ru-ru/windowsphone/default.aspx» найдите следующие документы:

1. Мобильное программирование, платформы для разработки.
2. Система Windows Phone 7. Microsoft Visual Studio Express for Windows Phone.
3. Аппаратные средства устройств, поддерживающих WindowsPhone 7.
4. Проектирование программы Silverlight. Язык XAML.
5. Пример создания приложения Silverlight для WindowsPhone.
6. Управление решениями в VisualStudio. Проекты и решения в VisualStudio. Отладка программ.
7. Улучшение приложения. Изменение и отображение данных.
8. Управление ориентацией страницы приложения. Отображение списков данных.
9. Навигация по страницам приложения. Использование классов ViewModel.
10. Хранилище данных WindowsPhone. Базы данных в WindowsPhone.
11. Создание связей данных в LINQ.
12. Основные сведения о сетях
13. . Создание подключения по протоколу UDP.
14. Создание подключения по протоколу TCP.
15. Подключение к сетевому ресурсу.
16. Чтение данных из XML-потока с помощью LINQ.
17. Взаимодействие приложений с сетевыми службами.
18. Основные сведения о технологии XNA.
19. Создание игрового мира. Использование средств WindowsPhone в играх.
20. Совместное использование XNA и Silverlight.
21. Значки и экраны-заставки приложений для WindowsPhone. Быстрое переключение приложений.
22. Задачи запуска и задачи выбора. Фоновые задачи.
23. Подготовка приложения для продажи.
24. Распространение приложений и игр для WindowsPhone.

Типовые тестовые задания

1) Набор средств программирования, который содержит инструменты, необходимые для создания, компиляции и сборки мобильного приложения называется:
а) Android SDK
б) JDK
в) плагин ADT
г) Android NDK
2) С какой целью был создан Open Handset Alliance? а) писать историю развития ОС Android
б) продавать смартфоны под управлением Android
в) рекламировать смартфоны под управлением Android
г) разрабатывать открытые стандарты для мобильных устройств
3) С какой целью инструмент Intel* Graphics Performance Analyzers (Intel* GPA) System Analyzer используется в среде разработки Intel* Beacon Mountain?
а) позволить разработчикам оптимизировать загруженность системы при использовании процеду р OpenGL
б) для ускорения работы эмулятора в среде разработки
в) для оптимизированной обработки данных и изображений
г) позволить разработчикам эффективно распараллелить C++ мобильные приложения
3) Библиотеки, реализованные на базе PacketVideo OpenCORE:
а) Media Framework
б) SQLite
в) FreeType
г) 3D библиотеки
4) Какой движок баз данных используется в ОС Android?
а) InnoDB
б) DBM
в) MyISAM
г) SQLite
5) С какой целью инструмент Intel* Integrated Performance Primitives (Intel* IPP) используется в среде разработки Intel* Beacon Mountain?
а) для оптимизированной обработки данных и изображений
б) позволить разработчикам оптимизировать загруженность системы при использовании процеду р OpenGL
в) для ускорения работы эмулятора в среде разработки
г) позволить разработчикам эффективно распараллелить C++ мобильные приложения

Критерии оценки на этапе зачета по дисциплине

зачет по дисциплине проводятся в форме устного опроса. Зачет проводится по расписанию в компьютерном классе.

Критерии и шкала оценки зачета по дисциплине.

Оценка	Характеристики ответа студента
Зачтено	- студент глубоко и всесторонне усвоил программный материал; - уверенно, логично, последовательно и грамотно его излагает; - опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью IT-специалиста; - умело обосновывает и аргументирует выдвигаемые им идеи; - делает выводы и обобщения; - аргументирует научные положения; - допускает несущественные ошибки и неточности;
Не зачтено	- студент не усвоил значительной части программного материала; - допускает существенные ошибки и неточности при рассмотрении задач в сфере деятельности IT-специалиста;

	- испытывает трудности в практическом применении знаний; - не может аргументировать научные положения; - не формулирует выводов и обобщений

7.2. Методические материалы, определяющие процедуры оценивания

Методические материалы, определяющие процедуры оценивания в рамках текущего контроля успеваемости

С целью определения уровня овладения компетенциями, закрепленными за дисциплиной, в заданные преподавателем сроки проводится текущий и промежуточный контроль знаний, умений и навыков каждого обучающегося.

Краткая характеристика процедуры реализации текущего и промежуточного контроля для оценки компетенций обучающихся представлена в таблице.

Процедура оценивания	Организация деятельности обучающегося
Выполнение практических заданий/творческих заданий	При выполнении практических заданий/творческих заданий обучающимся необходимо выполнить всю работу согласно тексту задания. Результаты работы сохранить в файлах. После выполнения задания необходимо преподавателю продемонстрировать результаты работы и быть готовым ответить на вопросы и продемонстрировать выполнение отдельных пунктов задания. Защита практических работ осуществляется на практических занятиях.
Устный опрос	Средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п. Развернутый ответ обучающегося должен представлять собой связное, логически последовательное сообщение на заданную тему, показывать его умение применять определения, правила в конкретных случаях. Показатели для оценки устного ответа: 1) знание материала; 2) последовательность изложения; 3) владение речью и профессиональной терминологией; 4) применение конкретных примеров; 5) знание ранее изученного материала; 6) уровень теоретического анализа; 7) степень самостоятельности; 8) степень активности в процессе; 9) выполнение регламента. Уровень знаний обучающегося определяется оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Критерии и шкала оценки приведены в п. 3. Фонда оценочных средств.
Зачет	**Зачет** проводится в устной форме по расписанию экзаменационной сессии.

	Зачет по дисциплине включает в себя: собеседование преподавателя со студентами по контрольным вопросам и ситуационным задачам. Контрольный вопрос — это средство контроля усвоения учебного материала дисциплины. Процедура проведения данного оценочного мероприятия включает в себя: беседу преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме дисциплины. Ситуационная задача — это оценочное средство, включающее совокупность условий, направленных на решение практически значимой ситуации с целью формирования компетенций, соответствующих основным типам профессиональной деятельности. Процедура проведения данного оценочного мероприятия включает в себя: оценку правильности решения задачи. В случае вариативности решения задачи следует обосновать все возможные варианты решения. Контрольные вопросы и ситуационные задачи к экзамену доводятся до сведения студентов заранее. Билет к экзамену содержит один контрольный вопрос и одну ситуационную задачу. При подготовке к ответу пользование учебниками, учебно-методическими пособиями, средствами связи и электронными ресурсами на любых носителях запрещено. Время на подготовку ответа – от 30 до 45 минут. По истечении времени подготовки ответа, студент отвечает на вопросы экзаменационного билета. На ответ обучающегося по каждому вопросу билета отводится, как правило, 3-5 минут. После ответа обучающегося преподаватель может задать дополнительные (уточняющие) вопросы в пределах предметной области экзаменационного задания. После окончания ответа преподаватель объявляет обучающемуся оценку по результатам экзамена, а также вносит эту оценку в экзаменационную ведомость, зачетную книжку. Уровень знаний, умений и навыков обучающегося определяется оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Перечень вопросов к экзамену, а также критерии и шкала оценки приведены в разделе 3. Фонда оценочных средств.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1. Основная литература

1. Соколова, В. В. Разработка мобильных приложений : учебник для среднего профессионального образования / В. В. Соколова. — Москва : Издательство Юрайт, 2025. — 160 с. — (Профессиональное образование). — ISBN 978-5-534-16868-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/566082

2. Соколова, В. В. Вычислительная техника и информационные технологии. Разработка мобильных приложений : учебник для вузов / В. В. Соколова. — Москва : Издательство Юрайт, 2025. — 160 с. — (Высшее образование). — ISBN 978-5-534-16302-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/561336

3. Тузовский, А. Ф. Объектно-ориентированное программирование : учебник для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2025. — 213 с. — (Высшее образование). — ISBN 978-5-534-16316-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/561394

8.2. Дополнительная литература

1. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 285 с. — (Высшее образование). — ISBN 978-5-534-16031-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/560815

Периодические издания:

–Прикладная информатика : научно-информационный журнал / Издательство университет «Синергия». – 2006. – Москва, 2006-2025. – ISSN 1993-8314. - Текст : электронный. - URL: http://www.iprbookshop.ru/11770.html

–IT Expert : журнал «Экспресс Электроника» / Издательство ИТ Медиа. - 1993. - Санкт-Петербург, 2009-2022. - Текст электронный. URL: https://www.iprbookshop.ru/38869.html

8.3.**Программноеобеспечение**

MicrosoftWindows, Яндекс 360, MicrosoftOfficeProfessionalPlus 2019, GoogleChrome, Яндекс.Браузер

8.4. Профессиональные базы данных

1. База данных IT специалиста» –http://info-comp.ru/
3. База данных «Стратегическое управление и планирование» – http://www.stplan.ru/

8.5. Информационные справочные системы

Справочно-правовая система «КонсультантПлюс» - http://www.consultant.ru/

Поисковые системы

Поисковая система Yandex- https://www.yandex.ru/

Поисковая система Rambler – https://www.rambler.ru/

8.6. Интернет-ресурсы

1. Национальный открытый университет Интуит – интернет университет информационных технологий – http://www.intuit.ru
2. Информационный ресурс «Projectimo.ru» – http://projectimo.ru
3. Академия ORACLE – https://academy.oracle.com/ru
4. Виртуальная академия Microsoft – https://www.bing.com
5. Все о компьютере и программировании для начинающих – https://info-comp.ru/
6. Маркетинговые исследования в области IT – http://www.marketing.spb.ru/mr/it/index.htm

8.7. Методические указания по освоению дисциплины

Методические указания для подготовки к лекции

Аудиторные занятия планируются в рамках такой образовательной технологии, как проблемно-ориентированный подход с учетом профессиональных и личностных особенностей обучающихся. Это позволяет учитывать исходный уровень знаний обучающихся, а также существующие технические возможности обучения.

Методологической основой преподавания дисциплины являются научность и объективность.

Лекция является первым шагом подготовки обучающихся к практическим занятиям. Проблемы, поставленные в ней, на практическом занятии приобретают конкретное выражение и решение.

Преподаватель на вводной лекции определяет структуру дисциплины, поясняет цели и задачи изучения дисциплины, формулирует основные вопросы и требования к результатам освоения. При проведении лекций, как правило, выделяются основные понятия и определения. При описании закономерностей обращается особое внимание на сравнительный анализ конкретных примеров.

На первом занятии преподаватель доводит до обучающихся требования к текущей и промежуточной аттестации, порядок работы в аудитории и нацеливает их на проведение самостоятельной работы с учетом количества часов, отведенных на нее учебным планом по направлению подготовки 40.03.01 Юриспруденция и рабочей программой по дисциплине (п. 5.5).

Рекомендуя литературу для самостоятельного изучения, преподаватель поясняет, каким образом максимально использовать возможности, предлагаемые библиотекой АНО ВО СКСИ, в том числе ее электронными ресурсами, а также сделает акцент на привлечение ресурсов сети Интернет и профессиональных баз данных для изучения практики.

Выбор методов и форм обучения по дисциплине определяется:

– общими целями образования, воспитания, развития и психологической подготовки обучающихся;

– особенностями учебной дисциплины и спецификой ее требований к отбору дидактических методов;

– целями, задачами и содержанием материала конкретного занятия;

– временем, отведенным на изучение того или иного материала;

– уровнем подготовленности обучающихся;

– уровнем материальной оснащенности, наличием оборудования, наглядных пособий, технических средств.

Лекции дают обучающимся систематизированные знания по дисциплине, концентрируют их внимание на наиболее сложных и важных вопросах.

Лекции обычно излагаются в традиционном или в проблемном стиле (интерактивном). Интерактивный стиль позволяет стимулировать активную познавательную деятельность обучающихся и их интерес к дисциплине, формировать творческое мышление, прибегать к противопоставлениям и сравнениям, делать обобщения, активизировать внимание обучающихся путем постановки проблемных вопросов, поощрять дискуссию. Во время лекционных занятий

рекомендуется вести конспектирование учебного материала, обращать внимание на формулировки и категории, раскрывающие суть того или иного явления или процессов, выводы и практические рекомендации.

В конце лекции делаются выводы и определяются задачи на самостоятельную работу. Во время лекционных занятий рекомендуется вести конспектирование учебного материала, обращать внимание на формулировки и категории, раскрывающие суть того или иного явления или процессов, научные выводы и практические рекомендации. В случае недопонимания какой-либо части предмета следует задать вопрос в установленном порядке преподавателю.

Конспект – это систематизированное, логичное изложение материала источника. Различаются четыре типа конспектов:

План-конспект – это развернутый детализированный план, в котором достаточно подробные записи приводятся по тем пунктам плана, которые нуждаются в пояснении.

Текстуальный конспект – это воспроизведение наиболее важных положений и фактов источника.

Свободный конспект – это четко и кратко сформулированные (изложенные) основные положения в результате глубокого осмысливания материала. В нем могут присутствовать выписки, цитаты, тезисы; часть материала может быть представлена планом.

Тематический конспект – составляется на основе изучения ряда источников и дает более или менее исчерпывающий ответ по какой-то схеме (вопросу).

Подготовленный конспект и рекомендуемая литература используются при подготовке к и практическим занятиям. Подготовка сводится к внимательному прочтению учебного материала, к выводу с карандашом в руках всех утверждений, к решению примеров, задач, к ответам на вопросы. Примеры, задачи, вопросы по теме являются средством самоконтроля.

Методические указания по подготовке к практическим работам

Целью практических работ является углубление и закрепление теоретических знаний, полученных обучающимися на лекциях и в процессе самостоятельного изучения учебного материала, а, следовательно, формирование у них определенных умений и навыков.

В ходе подготовки к практическим работам необходимо прочитать конспект лекции, изучить основную литературу, ознакомиться с дополнительной литературой, выполнить выданные преподавателем задания. При этом учесть рекомендации преподавателя и требования программы. Дорабатывать свой конспект лекции, делая в нем соответствующие записи из литературы. Желательно при подготовке к практическим работам по дисциплине одновременно использовать несколько источников, раскрывающих заданные вопросы.

Методические указания для выполнения самостоятельной работы

Самостоятельная работа обучающихся заключается:

В целях наиболее эффективного изучения дисциплины подготовлены различные задания, различающиеся по преследуемым целям.

Задания представлены – 1) контрольными вопросами, предназначенными для самопроверки; 2) письменными заданиями, включающими задачи и задание.

Задачи самостоятельной внеаудиторной работы обучающихся заключаются в продолжении изучения теоретического материала дисциплины и в развитии навыков самостоятельного анализа литературы.

I. Самостоятельное теоретическое обучение предполагает освоение студентом во внеаудиторное время рекомендуемой преподавателем основной и дополнительной литературы. С этой целью обучающимся рекомендуется постоянно знакомиться с классическими теоретическими источниками по темам дисциплины, а также с новинками литературы, статьями в периодических изданиях, справочных правовых системах.

Для лучшего понимания материала целесообразно осуществлять его конспектирование с возможным последующим его обсуждением на практических занятиях, на научных семинарах и в индивидуальных консультациях с преподавателем. Формы конспектирования материала могут быть различными:

1) обобщение – при подготовке такого конспекта студентом осуществляется анализ и обобщение всех существующих в доктрине подходов по выбранному дискуссионному вопросу раздела, в том числе, дореволюционных ученых, ученых советского и современного периода развития. Основная задача обучающегося заключается не только в изложении точек зрения по исследуемому вопросу, но и в выражении собственной позиции с соответствующим развернутым теоретическим обоснованием.

2) рецензия – при подготовке такого конспекта студентом осуществляется рецензирование выбранного источника по изучаемому дискуссионному вопросу, чаще всего, статьи и периодическом издании, тезисов выступления на конференции либо главы из монографии. Для этого студентом дается оценка содержанию соответствующего источника по следующим параметрам: актуальность выбранной темы, в том числе убедительность обоснования актуальности исследования автором; соответствие содержания работы ее названию; логичность, системность и аргументированность (убедительность) выводов автора; научная добросовестность (наличие ссылок на использованные источники, самостоятельность исследования, отсутствие фактов недобросовестных заимствований текстов, идей и т.п.); научная новизна и др.

Формами контроля за самостоятельным теоретическим обучением являются теоретические опросы, которые осуществляются преподавателем на практических занятиях в устной форме, преследующие цель проверки знаний обучающихся по основным понятиям и терминам по теме дисциплины. В случае представления студентом выполненного им в письменном виде конспекта по предложенным вопросам темы, возможна его защита на практическом занятии или в индивидуальном порядке.

II. Ключевую роль в планировании индивидуальной траектории обучения по дисциплине играет *опережающая самостоятельная работа* (ОПС). Такой тип обучения предлагается в замену традиционной репродуктивной самостоятельной работе (самостоятельное повторение учебного материала и рассмотренных на занятиях алгоритмов действий, выполнение по ним аналогичных заданий). ОПС предполагает следующие виды самостоятельных работ:

познавательно-поисковая самостоятельная работа, предполагающая подготовку докладов, выступлений на практических занятиях, подбор литературы по конкретной проблеме, написание рефератов и др.;

творческая самостоятельная работа, к которой можно отнести выполнение специальных творческих и нестандартных заданий. Задача преподавателя на этапе планирования самостоятельной работы – организовать ее таким образом, чтобы максимально учесть индивидуальные способности каждого обучающегося, развить в нем познавательную потребность и готовность к выполнению самостоятельных работ все более высокого уровня. Студенты, приступая к изучению тем, должны применить свои навыки работы с библиографическими источниками и рекомендуемой литературой, умение четко формулировать свою собственную точку зрения и навыки ведения научных дискуссий. Все подготовленные и представленные тексты должны являться результатом самостоятельной информационно-аналитической работы обучающихся. На их основе студенты готовят материалы для выступлений в ходе практических занятий.

Подготовка к устному опросу

Самостоятельная работа обучающихся включает подготовку к устному опросу на практических занятиях. Для этого студент изучает лекции, основную и дополнительную литературу, публикации, информацию из Интернет-ресурсов.Кроме того, изучению должны быть подвергнуты различные источники права, как регламентирующие правоотношения, возникающие в рамках реализации основ права, так и отношения, что предопределяют реализацию их, либо следуют за ними.

Тема и вопросы к практическим занятиям по дисциплине доводятся до обучающихся заранее. Эффективность подготовки обучающихся к устному опросу зависит от качества ознакомления с рекомендованной литературой. Для подготовки к устному опросу студенту необходимо ознакомиться с материалом, посвященным теме практического занятия, в

рекомендованной литературе, записях с лекционного занятия, обратить внимание на усвоение основных понятий дисциплины, выявить неясные вопросы и подобрать дополнительную литературу для их освещения, составить тезисы выступления по отдельным проблемным аспектам. В среднем, подготовка к устному опросу по одному практическому занятию занимает от 2 до 4 часов в зависимости от сложности темы и особенностей организации студентом своей самостоятельной работы.

Методические указанияк подготовке и проведению лекции с элементами дискуссии, постановкой проблем

Правильно организованная дискуссия проходит три стадии развития: ориентация, оценка и консолидация.

На первой стадии вырабатывается определенная установка на решение поставленной проблемы. При этом перед преподавателем (организатором дискуссии) ставятся следующие задачи:

1. Сформулировать проблему и цели дискуссии. Для этого надо объяснить, что обсуждается, что должно дать обсуждение.
2. Создать необходимую мотивацию, т.е. изложить проблему, показать ее значимость, выявить в ней нерешенные и противоречивые вопросы, определить ожидаемый результат (решение).
3. Установить регламент дискуссии, а точнее, регламент выступлений, так как общий регламент определяется продолжительностью практического занятия.
4. Сформулировать правила ведения дискуссии, основное из которых — выступить должен каждый.
5. Добиться однозначного семантического понимания терминов, понятий и т п.

Вторая стадия — стадия оценки — обычно предполагает ситуацию сопоставления, конфронтации и даже конфликта идей. На этой стадии перед преподавателем ставятся следующие задачи:

1. Начать обмен мнениями, что предполагает предоставление слова конкретным участникам.
2. Собрать максимум мнений, идей, предложений. Для этого необходимо активизировать каждого обучающегося. Выступая со своим мнением, студент может сразу внести свои предложения, а может сначала просто выступить, а позже сформулировать свои предложения.
3. Не уходить от темы, что требует некоторой твердости организатора, а иногда даже авторитарности. Следует тактично останавливать отклоняющихся, направляя их в заданное «русло»,
4. Поддерживать высокий уровень активности всех участников. Не допускать чрезмерной активности одних за счет других, соблюдать регламент, останавливать затянувшиеся монологи, подключать к разговору всех присутствующих обучающихся.
5. Оперативно проводить анализ высказанных идей, мнений, позиций, предложений перед тем, как переходить к следующему витку дискуссии. Такой анализ, предварительные выводы или резюме целесообразно делать через определенные интервалы (каждые 10—15 минут), подводя при этом промежуточные итоги.
6. В конце дискуссии предоставить право обучающимся самим оценить свою работу (рефлексия).

Третья стадия — стадия консолидации — предполагает выработку определенных единых или компромиссных мнений, позиций, решений. На этом этапе осуществляется контролирующая функция. Задачи, которые должен решить преподаватель, можно сформулировать следующим образом:

1. Проанализировать и оценить проведенную дискуссию, подвести итоги, результаты. Для этого надо сопоставить сформулированную в начале дискуссии цель с полученными результатами, сделать выводы, вынести решения, оценить результаты, выявить их положительные и отрицательные стороны.

2. Помочь участникам дискуссии прийти к согласованному мнению, чего можно достичь путем внимательного выслушивания различных толкований, поиска общих тенденций для принятия решений.

3. Принять групповое решение совместно с участниками. При этом следует подчеркнуть важность разнообразных позиций и подходов.

4. В заключительном слове подвести группу к конструктивным выводам, имеющим познавательное и практическое значение.

Составной частью любой дискуссии является процедура *вопросов и ответов*.

С функциональной точки зрения, все вопросы можно разделить на две группы:

- *Уточняющие (закрытые)* вопросы, направленные на выяснение истинности или ложности высказываний, грамматическим признаком которых обычно служит наличие в предложении частицы «ли», например: «Верно ли что?», «Правильно ли я понял, что?». Ответить на такой вопрос можно только «да» или «нет».
- *Восполняющие (открытые)* вопросы, направленные на выяснение новых свойств или качеств интересующих нас явлений, объектов. Их грамматический признак — наличие вопросительных слов: *что, где, когда, как, почему* и т.д.

Методические указания по подготовке к промежуточной аттестации

Промежуточная аттестация по дисциплине проводится в форме экзамена.

Для допуска к экзамену студенту необходимо выполнить и успешно сдать практические работы (практические задания) по каждой теме.

При подготовке к экзамену необходимо повторить конспекты лекций по всем разделам дисциплины. До экзамена обычно проводится консультация, но она не может возместить отсутствия систематической работы в течение триместра и помочь за несколько часов освоить материал, требующийся к экзамену. На консультации студент получает лишь ответы на трудные или оставшиеся неясными вопросы. Польза от консультации будет только в том случае, если студент до нее проработает весь материал.

На экзамене студент должен подтвердить усвоение учебного материала, предусмотренного рабочей программой дисциплины, а также продемонстрировать приобретенные навыки адаптации полученных теоретических знаний к своей профессиональной деятельности. Экзамен проводится в форме устного собеседования по контрольным вопросам, а также обучающемуся необходимо решить ситуационную задачу.

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для реализации дисциплины необходимо следующее материально-техническое обеспечение:

- для проведения занятий лекционного типа - аудитория, оборудованная мультимедийными средствами обучения: проектором, ПК, экраном, доской;

- для проведения лабораторных занятий - компьютерный класс с предустановленным программным обеспечением, указанным в п.8.3.

- для проведения промежуточной аттестации - компьютерный класс с предустановленным программным обеспечением, указанным в п.8.3.

- практическая подготовка - компьютерный класс с предустановленным программным обеспечением, указанным в п.8.3.

- для самостоятельной работы: помещение для самостоятельной работы с возможностью подключения к информационно-коммуникационной сети «Интернет» и обеспечением доступав электронную информационно-образовательную среду.

10.ОСОБЕННОСТИ ОСВОЕНИЯ ДИСЦИПЛИНЫ ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники, учебные пособия и дидактические материалы, специальные технические средства обучения коллективного и индивидуального пользования, услуги ассистента (тьютора), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

Освоение дисциплины обучающимися с ограниченными возможностями здоровья и инвалидами может быть организовано совместно с другими обучающимися, а также в отдельных группах.

Освоение дисциплины обучающимися с ограниченными возможностями здоровья и инвалидами осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья.

В целях доступности получения среднего профессионального образования по образовательной программе лицами с ограниченными возможностями здоровья при освоении дисциплины обеспечивается:

1) для лиц с ограниченными возможностями здоровья по зрению:

– присутствие тьютора, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку),

– письменные задания, а также инструкции о порядке их выполнения оформляются увеличенным шрифтом,

– специальные учебники, учебные пособия и дидактические материалы (имеющие крупный шрифт или аудиофайлы),

– индивидуальное равномерное освещение не менее 300 люкс,

– при необходимости студенту для выполнения задания предоставляется увеличивающее устройство;

2) для лиц с ограниченными возможностями здоровья по слуху:

– присутствие ассистента, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку),

– обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающемуся предоставляется звукоусиливающая аппаратура индивидуального пользования;

– обеспечивается надлежащими звуковыми средствами воспроизведения информации;

3) для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата:

– письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются тьютору;

– по желанию студента задания могут выполняться в устной форме.