

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

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

УТВЕРЖДАЮ

И.о. декана факультета информационных
технологий и компьютерной безопасности

/А.В. Бредихин/

2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Системы разработки программного обеспечения»

Направление подготовки 09.04.01 Информатика и вычислительная техника

**Профиль Жизненный цикл изделий в едином информационном
пространстве цифрового производства**

Квалификация выпускника магистр

Нормативный период обучения 2 года / 2 года и 4 м.

Форма обучения очная / заочная

Год начала подготовки 2024

Автор программы

А.С. Троценко

И.о. заведующего кафедрой
Компьютерных
интеллектуальных
технологий проектирования

М.И. Чижов

Руководитель ОПОП

М.И. Чижов

Воронеж 2024

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

Целью дисциплины является предоставление обучаемым знаний и умений в области проектирования, тестирования, отладки, внедрения и сопровождения программного обеспечения (ПО) вычислительной техники с использованием современных CALS-технологий и CASE-средств.

1.2. Задачи освоения дисциплины

- современные тенденции развития информатики и вычислительной техники, компьютерных технологий;
- основы создания информационных систем и использование новых информационных технологий обработки информации;
- жизненный цикл программного обеспечения;
- стандарты кодирования;
- международные стандарты и знания программной инженерии

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Системы разработки программного обеспечения» относится к дисциплинам части, формируемой участниками образовательных отношений блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Системы разработки программного обеспечения» направлен на формирование следующих компетенций:

ПК-1 - Способен формализовать задачи по разработке модулей компонентов программных средств поддержки жизненного цикла изделия

ПК-4 - Способен проводить проверку работоспособности программных продуктов и цифровых моделей

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-1	знать подходы к формализации задач по разработке модулей компонентов программных средств поддержки жизненного цикла изделия
	уметь применять знания при создании и тестировании ПО и модулей в рамках жизненного цикла программного обеспечения
	владеть инструментальными средствами по разработке модулей ЖЦ ПО
ПК-4	знать алгоритмы и методы тестирования ПО, анализировать цифровые модели
	уметь применять алгоритмы на практике
	владеть инструментами для подготовки отчетов по отладочным процессам и тестированию в процессе

	разработки
--	------------

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Системы разработки программного обеспечения» составляет 3 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры
		2
Аудиторные занятия (всего)	54	54
В том числе:		
Лекции	18	18
Лабораторные работы (ЛР)	36	36
Самостоятельная работа	54	54
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость:		
академические часы	108	108
зач.ед.	3	3

заочная форма обучения

Виды учебной работы	Всего часов	Семестры
		2
Аудиторные занятия (всего)	12	12
В том числе:		
Лекции	4	4
Лабораторные работы (ЛР)	8	8
Самостоятельная работа	92	92
Часы на контроль	4	4
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость:		
академические часы	108	108
зач.ед.	3	3

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Введение в дисциплину	Почему программному обеспечению присуща сложность. Сложность реальной предметной области, сложность описания поведения больших дискретных систем, сложность управления коллективом разработчиков. Проблемы, возникающие при общении с заказчиками программных систем. Сложность оценки качества программного обеспечения.	4	6	8	18
2	Жизненный цикл	Жизненный цикл программного обеспечения.	4	6	8	18

	программного обеспечения	Распределение финансовых и временных затрат на реализацию каждого из этапов разработки программного обеспечения.				
3	Требования к ПО	Обследование системы, планирование разработки, составление технического задания. Детальный анализ предметной области, принятие окончательного решения о необходимости создания информационной системы, проектирование общей архитектуры системы, выбор метода проектирования.	4	6	8	18
4	Обзор методологий проектирования программных продуктов	Каскадные и итеративные технологии. Критичность и масштабность программных проектов.	2	6	10	18
5	Тестирование и отладка программных систем	Стратегии и методы тестирования. Прямое и обратное тестирование. Программные средства автоматизации тестирования.	2	6	10	18
6	Оценка качества ПО. Сопровождение ПО.	Методики оценки качества ПО. Процессный подход к оценке качества ПО. Основные задачи, решаемые на этапе внедрения. Процесс устранения ошибок на этапе внедрения. Техническая поддержка пользователей на этапе сопровождения.	2	6	10	18
Итого			18	36	54	108

заочная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Введение в дисциплину	Почему программному обеспечению присуща сложность. Сложность реальной предметной области, сложность описания поведения больших дискретных систем, сложность управления коллективом разработчиков. Проблемы, возникающие при общении с заказчиками программных систем. Сложность оценки качества программного обеспечения.	2	2	14	18
2	Жизненный цикл программного обеспечения	Жизненный цикл программного обеспечения. Распределение финансовых и временных затрат на реализацию каждого из этапов разработки программного обеспечения.	2	2	14	18
3	Требования к ПО	Обследование системы, планирование разработки, составление технического задания. Детальный анализ предметной области, принятие окончательного решения о необходимости создания информационной системы, проектирование общей архитектуры системы, выбор метода проектирования.	-	2	16	18
4	Обзор методологий проектирования программных продуктов	Каскадные и итеративные технологии. Критичность и масштабность программных проектов.	-	2	16	18
5	Тестирование и отладка программных систем	Стратегии и методы тестирования. Прямое и обратное тестирование. Программные средства автоматизации тестирования.	-	-	16	16
6	Оценка качества ПО. Сопровождение ПО.	Методики оценки качества ПО. Процессный подход к оценке качества ПО. Основные задачи, решаемые на этапе внедрения. Процесс устранения ошибок на этапе внедрения. Техническая поддержка пользователей на этапе сопровождения.	-	-	16	16
Итого			4	8	92	104

5.2 Перечень лабораторных работ

1. Создание и настройка среды разработки программных проектов.
2. Выбор и реализация шаблона некоторой модели приложения.
3. Тестирование и отладочные процессы в проекте.
4. Анализ полученных результатов, поиск зависимостей.

5. Профилирование программной разработки.
6. Создание установочного пакета.
7. Запуск и эксплуатация проектных решений в ОС.

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

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

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

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

7.1.1 Этап текущего контроля

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

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ПК-1	знать подходы к формализации задач по разработке модулей компонентов программных средств поддержки жизненного цикла изделия	Выполнение лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь применять знания при создании и тестировании ПО и модулей в рамках жизненного цикла программного обеспечения	Выполнение лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть инструментальными средствами по разработке модулей ЖЦ ПО	Выполнение лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ПК-4	знать алгоритмы и методы тестирования ПО, анализировать цифровые модели	Выполнение лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

уметь применять алгоритмы на практике	Выполнение лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
владеть инструментами для подготовки отчетов по отладочным процессам и тестированию в процессе разработки	Выполнение лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 2 семестре для очной формы обучения, 2 семестре для заочной формы обучения по двухбалльной системе:

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Не зачтено
ПК-1	знать подходы к формализации задач по разработке модулей компонентов программных средств поддержки жизненного цикла изделия	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь применять знания при создании и тестировании ПО и модулей в рамках жизненного цикла программного обеспечения	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть инструментальными средствами по разработке модулей ЖЦ ПО	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ПК-4	знать алгоритмы и методы тестирования ПО, анализировать цифровые модели	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь применять алгоритмы на практике	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть инструментами	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в	Задачи не решены

	для подготовки отчетов по отладочным процессам и тестированию в процессе разработки	области	большинстве задач	
--	---	---------	-------------------	--

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

7.2.1 Примерный перечень заданий для подготовки к тестированию

1. Что такое жизненный цикл программы?
 - а) Поиск способов повышения эффективности работы программы.
 - б) Период времени между принятием решения о создании программного продукта и его изъятием из эксплуатации.
 - в) Экспериментальный анализ сложности алгоритма.
 - г) Поэтапная модель с промежуточным контролем.
2. Что такое программа?
 - а) Точное описание порядка действий для решения задачи.
 - б) Комплекс взаимосвязанных программ для поставки и продажи пользователю.
 - в) Последовательность инструкций для вычислительной машины.
 - г) Модель, характеризующаяся возможностью перехода с одного этапа на другой после завершения текущего этапа.
3. Какие существуют модели жизненного цикла программного обеспечения?
 - а) Каскадная модель.
 - б) Инкрементная модель.
 - в) Спиральная модель.
 - г) Поэтапная модель с промежуточным контролем.
4. В каком порядке обычно располагаются этапы жизненного цикла программы?
 - а) Анализ требований, определение спецификаций, проектирование, кодирование, тестирование и отладка, эксплуатация и сопровождение.
 - б) Определение спецификаций, анализ требований, проектирование, кодирование, тестирование и отладка, эксплуатация и сопровождение.
 - в) Анализ требований, проектирование, определение спецификаций, кодирование, тестирование и отладка, эксплуатация и сопровождение.
 - г) Определение спецификаций, анализ требований, проектирование, кодирование, тестирование и отладка, эксплуатация и сопровождение.
5. Что такое анализ требований?
 - а) Перевод на язык программирования конструкций, записанных на языке проектирования.
 - б) Создание общей структуры программы, удовлетворяющей спецификациям.

- в) Ответ на вопрос: «Что должна делать программа?»
 - г) Экспериментальный анализ сложности алгоритма.
6. Что такое спецификация?
- а) Описание функций и поведения программы.
 - б) Набор требований к программе.
 - в) Процесс перевода на язык программирования конструкций, записанных на языке проектирования.
 - г) Создание общей структуры программы, удовлетворяющей спецификациям.
7. Что такое проектирование?
- а) Создание общей структуры программы, удовлетворяющей спецификациям.
 - б) Перевод на язык программирования конструкций, записанных на языке проектирования.
 - в) Процесс определения требований к программе.
 - г) Этап разработки компьютерной программы, в процессе которого проверяется работоспособность программы без явных ошибок.
8. Что такое кодирование?
- а) Перевод на язык программирования конструкций, записанных на языке проектирования.
 - б) Создание общей структуры программы, удовлетворяющей спецификациям.
 - в) Процесс определения требований к программе.
 - г) Этап разработки компьютерной программы, в процессе которого происходит обнаружение, локализация и устранение явных ошибок в программе.
9. Что такое тестирование программ?
- а) Этап разработки компьютерной программы, в процессе которого проверяется работоспособность программы без явных ошибок.
 - б) Этап разработки компьютерной программы, в процессе которого происходят обнаружение, локализация и устранение явных ошибок в программе.
 - в) Проверка соответствия функциональности программного обеспечения решаемым задачам.
 - г) Экспериментальный анализ сложности алгоритма или экспериментальное сравнение нескольких алгоритмов, решающих одну и ту же задачу.
10. Что такое отладка программы?
- а) Этап разработки компьютерной программы, в процессе которого проверяется работоспособность программы без явных ошибок.
 - б) Этап разработки компьютерной программы, в процессе которого происходят обнаружение, локализация и устранение явных ошибок в программе.
 - в) Экспериментальный анализ сложности алгоритма или экспериментальное сравнение нескольких алгоритмов, решающих одну и ту же задачу.

же задачу.

7.2.2 Примерный перечень заданий для решения стандартных задач

1. Что такое исследовательское тестирование?

а) Тестирование, основанное на анализе требований.
б) Тестирование, основанное на анализе пользовательского интерфейса.

в) Тестирование, основанное на анализе кода.

г) Тестирование, основанное на анализе бизнес-процессов.

2. Что такое тестирование вариантов использования?

а) Тестирование, основанное на анализе требований.
б) Тестирование, основанное на анализе пользовательского интерфейса.

в) Тестирование, основанное на анализе кода.

г) Тестирование, основанное на анализе бизнес-процессов.

3. В чём разница между STLC (жизненным циклом тестирования программного обеспечения) и SDLC (жизненным циклом разработки программного обеспечения)?

а) STLC предшествует SDLC.

б) STLC и SDLC являются разными названиями одного и того же процесса.

в) SDLC предшествует STLC.

г) STLC и SDLC не связаны друг с другом.

4. Что такое матрица прослеживаемости?

а) Таблица, связывающая требования с тестовыми сценариями.

б) График, показывающий зависимости между модулями кода.

в) Диаграмма, иллюстрирующая процесс тестирования.

г) Документ, описывающий стратегию тестирования.

5. Что такое тестирование эквивалентного разделения?

а) Метод тестирования, основанный на разделении требований на эквивалентные группы.

б) Метод тестирования, основанный на разделении пользователей на группы.

в) Метод тестирования, основанный на разделении кода на модули.

г) Метод тестирования, основанный на разделении тестовых сценариев на группы.

6. Что такое white box тестирование и перечислить виды white box тестирования?

а) Тестирование, основанное на анализе требований.

б) Тестирование, основанное на анализе пользовательского интерфейса.

в) Тестирование, основанное на анализе кода.

г) Тестирование, основанное на анализе бизнес-процессов.

7. В white box тестировании, что вы проверяете?

а) Соответствие требованиям.

б) Удобство использования.

в) Ошибки в коде.

г) Безопасность данных.

8. Что такое black box тестирование?

а) Тестирование, основанное на анализе требований.

б) Тестирование, основанное на анализе пользовательского интерфейса.

в) Тестирование, основанное на анализе кода.

г) Тестирование, основанное на анализе бизнес-процессов.

9. В чём разница между статическим и динамическим тестированием?

а) Статическое тестирование проводится до начала разработки, а динамическое тестирование — после.

б) Статическое тестирование проверяет требования, а динамическое тестирование — пользовательский интерфейс.

в) Статическое тестирование проверяет код, а динамическое тестирование — работу системы.

г) Статическое тестирование проверяет безопасность данных, а динамическое тестирование — производительность системы.

10. Что такое верификация и валидация?

а) Верификация — это проверка требований, а валидация — проверка пользовательского интерфейса.

б) Верификация — это проверка кода, а валидация — проверка работы системы.

в) Верификация — это проверка безопасности данных, а валидация — проверка производительности системы.

г) Верификация и валидация — это разные названия одного и того же процесса.

7.2.3 Примерный перечень заданий для решения прикладных задач

1. Приложение с использованием БД.

2. Приложение с распараллеливанием вычислительных действий.

3. Приложение с настраиваемым параметрическим интерфейсом.

4. Приложение с визуализацией твердотельных моделей.

5. Приложение по обработке различных форматов данных.

6. Приложение с по поиску данных в некоторой цифровой модели (моделях).

7. Встраиваемое приложение (модуль) в одну из систем САПР по моделированию.

8. Модуль для систем управления цифровыми данными на производстве.

9. Создание приложения с использованием геометрических ядер.

10. Создание модуля по работе с программируемым устройством.

7.2.4 Примерный перечень вопросов для подготовки к зачету

1. Что такое промышленный программный продукт. Дать определения пакета прикладных программ, программной системы.

2. Жизненный цикл программного обеспечения. Дать краткую характеристику каждого этапа.
3. Почему программные системы сложны. Привести пять признаков сложной системы.
4. Техническое задание. Перечислить и охарактеризовать разделы, входящие в техническое задание.
5. Унифицированный процесс разработки программного обеспечения. Жизненный цикл унифицированного процесса.
6. Работа с кадрами. Перечислить роли разработчиков и дать характеристику каждой из них.
7. Дать определения проекта, процесса, продукта с точки зрения унифицированного процесса разработки программного обеспечения.
8. Что такое артефакт. В чем преимущества организованного процесса разработки программного обеспечения.
9. Использование языка UML при проектировании сложных программных систем. Какие диаграммы используются в UML для создания моделей программной системы.
10. Диаграмма вариантов использования, ее назначение. Рассказать о варианте использования и действующем лице. Правила построения диаграммы вариантов использования.
11. Понятие класса и объекта. Что может быть объектом. Что такое атрибут и операция.
12. Пять критериев проверки правильности построения класса.
13. Что такое классификация с точки зрения объектно-ориентированного проектирования программных систем. Теории классификации.
14. Методы классификации.
15. Микропроцесс проектирования. Перечислить этапы и основные виды деятельности выполняемые на каждом из них.
16. Микропроцесс проектирования – первый этап.
17. Микропроцесс проектирования – второй этап.
18. Микропроцесс проектирования – третий этап.
19. Микропроцесс проектирования – четвертый этап.
20. Диаграммы взаимодействия. Основное назначение.
21. Диаграмма классов. Ее назначение. Что она включает. Рассказать об основных видах связей между классами.
22. Дать определение тестированию и отладке. Особенности и объекты тестирования. Автономное и комплексное тестирование.
23. Дать определение тестированию и отладке. Направления тестирования. Стратегия тестирования. Контрольный лист тестирования модуля.
24. Дать определение тестированию и отладке. Локализация ошибок. Классификация ошибок. Безопасное программирование.
25. Оценки ошибок.
26. Документирование. Состав и содержание документов

прилагаемых к программной системе.

27. Внедрение программного комплекса. Планирование испытаний.

28. Внедрение программного комплекса. Подготовка тестовых данных. Анализ результатов испытаний.

29. Что такое качество с точки зрения квалиметрии. Дать определение свойству и показателю качества ПО. Основные задачи решаемые при оценке качества.

30. Оценка качества программного обеспечения. Методы оценки свойств программного обеспечения.

7.2.5 Примерный перечень заданий для подготовки к экзамену

Не предусмотрено учебным планом

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Зачет проводится по тест-билетам, каждый из которых содержит 10 вопросов и задачу. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, задача оценивается в 5 баллов. Максимальное количество набранных баллов – 15.

1. Оценка «не зачтено» ставится в случае, если студент набрал менее 8 баллов.

2. Оценка «зачтено» ставится в случае, если студент набрал от 9 до 15 баллов

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Введение в дисциплину	ПК-1, ПК-4	Тест, защита лабораторных работ
2	Жизненный цикл программного обеспечения	ПК-1, ПК-4	Тест, защита лабораторных работ
3	Требования к ПО	ПК-1, ПК-4	Тест, защита лабораторных работ
4	Обзор методологий проектирования программных продуктов	ПК-1, ПК-4	Тест, защита лабораторных работ
5	Тестирование и отладка программных систем	ПК-1, ПК-4	Тест, защита лабораторных работ
6	Оценка качества ПО. Сопровождение ПО.	ПК-1, ПК-4	Тест, защита лабораторных работ

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

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

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных

задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

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

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1. С. Орлов. Технологии разработки программного обеспечения. Учебное пособие. — СПб.: Изд-во «Питер», 2003. – 480 с.

2. Мирошниченко Е.А. Технология программирования: Учебное пособие. – Томск: Изд. ТПУ, 2011. – 42 с.

3. ИСО 9000-3: ИСО 9001 Общее руководство качеством и стандарты по обеспечению качества, часть 3: Руководящие указания по применению ИСО 9001 при разработке, поставке и обслуживанию программного. Международная организация стандартов, Женева, 1991.

4. ИСО/МЭК 9126 Информационные технологии. Оценка продукции программного обеспечения. Характеристики качества и инструкции по их применению. Международная организация стандартов, Женева, 1991.

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

-Среда разработки Qt Creator +Qt SDK;

-Среда разработки Microsoft Visual Studio с компонентом SQL Server Data Tools.

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

- Проекционная аппаратура;

- Учебная лаборатория с доступом к локальной сети и Интернет (202/2, расположенная по адресу г. Воронеж, ул. Плехановская, д. 11)

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

По дисциплине «Системы разработки программного обеспечения» читаются лекции, проводятся лабораторные работы.

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

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

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

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП
----------	-----------------------------	----------------------------	--