

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**  
**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**  
**(РУТ (МИИТ))**



Рабочая программа дисциплины (модуля),  
как компонент образовательной программы  
специализированного высшего образования  
по направлению подготовки  
09.04.01 Информатика и вычислительная техника,  
утвержденной первым проректором РУТ (МИИТ)  
Тимониным В.С.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

**Технология разработки программного обеспечения**

Направление подготовки: 09.04.01 Информатика и вычислительная техника

Направленность (профиль): Информационные технологии в строительстве

Форма обучения: Очная

Рабочая программа дисциплины (модуля) в виде электронного документа выгружена из единой корпоративной информационной системы управления университетом и соответствует оригиналу

Простая электронная подпись, выданная РУТ (МИИТ)  
ID подписи: 2899  
Подписал: заведующий кафедрой Нестеров Иван Владимирович  
Дата: 29.06.2026

## 1. Общие сведения о дисциплине (модуле).

Целями освоения учебной дисциплины являются усвоение знаний, развитие профессиональных умений и навыков, необходимых в области разработки программного обеспечения (ПО), знакомство с современными направлениями программирования.

В задачи освоения дисциплины входят:

1. знакомство с современными языками и системами программирования;
2. изучение процессов, моделей и стадий жизненного цикла ПО, соответствующих принятым международным стандартам;
3. овладение структурным и объектно-ориентированным подходами к проектированию ПО;
4. знакомство с CASE-средствами, поддерживающими как структурный, так и объектно-ориентированный подходы к проектированию ПО, а также промышленными технологиями проектирования ПО;
5. выработка научного подхода к практике применения теоретических знаний в области технологий разработки ПО;
6. повышение мотивации к процессу изучения учебной дисциплины и научной деятельности.

## 2. Планируемые результаты обучения по дисциплине (модулю).

Перечень формируемых результатов освоения образовательной программы (компетенций) в результате обучения по дисциплине (модулю):

**ПК-6** - Применение перспективных методов исследования и решения профессиональных задач на основе знания мировых тенденций развития вычислительной техники и информационных технологий.

Обучение по дисциплине (модулю) предполагает, что по его результатам обучающийся будет:

### **Знать:**

- математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте;
- программное и аппаратное обеспечение информационных и автоматизированных систем;
- способы управления разработкой программных средств и проектов;
- приоритеты собственной деятельности и способы ее совершенствования на основе самооценки.

**Уметь:**

- применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте;
- устанавливать программное и аппаратное обеспечение информационных и автоматизированных систем;
- использовать способы управления разработкой программных средств и проектов;
- определять приоритеты собственной деятельности и способы ее совершенствования на основе самооценки.

**Владеть:**

- математическими, естественнонаучными, социально-экономическими и профессиональными знаниями для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте;
- способностью разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем;
- способами управления разработкой программных средств и проектов;
- реализацией приоритетами собственной деятельности и способами ее совершенствования на основе самооценки.

**3. Объем дисциплины (модуля).****3.1. Общая трудоемкость дисциплины (модуля).**

Общая трудоемкость дисциплины (модуля) составляет 9 з.е. (324 академических часа(ов)).

3.2. Объем дисциплины (модуля) в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Тип учебных занятий	Количество часов		
	Всего	Семестр	
		№1	№2
Контактная работа при проведении учебных занятий (всего):	64	32	32
В том числе:			
Занятия лекционного типа	32	16	16
Занятия семинарского типа	32	16	16

3.3. Объем дисциплины (модуля) в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации составляет 260 академических часа (ов).

3.4. При обучении по индивидуальному учебному плану, в том числе при ускоренном обучении, объем дисциплины (модуля) может быть реализован полностью в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации.

#### 4. Содержание дисциплины (модуля).

##### 4.1. Занятия лекционного типа.

№ п/п	Тематика лекционных занятий / краткое содержание
1	<p>Языки и системы программирования. Основные понятия и определения. Языки программирования.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- базовые понятия, определяющие область программирования (языки программирования, их роль и значение в создании программного обеспечения);</li> <li>- определение категорий языков программирования;</li> <li>- критерии оценки и выбор наиболее подходящего инструмента для реализации проекта.</li> </ul>
2	<p>Системы программирования. Распределенное программирование.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- основы современных систем программирования, включая интегрированные среды разработки (IDE), компиляторы, интерпретаторы и другие инструменты.</li> <li>- особенности распределённого программирования и взаимодействие модулей программного обеспечения.</li> </ul>
3	<p>Методологии и технологии разработки программного обеспечения. Жизненный цикл программного обеспечения. Методы и средства разработки программного обеспечения.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- структура жизненного цикла разработки программного обеспечения;</li> <li>- основные стадии ПО (анализ, проектирование, разработка, тестирование, эксплуатация и сопровождение);</li> <li>- методологии разработки (водопадная, итерационная, гибкая (Agile)).</li> </ul>
4	<p>Технологии коллективной разработки программного обеспечения. Оценка качества программного обеспечения.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- современные практики и инструменты для эффективной работы команды разработчиков;</li> <li>- системы контроля версий, интеграция CI/CD, методы улучшения совместимости труда группы</li> </ul>

№ п/п	Тематика лекционных занятий / краткое содержание
	инженеров; - метрики и показатели качества программного обеспечения для оценки продуктивности и надежности ПО.
5	<p>Проектирование программного обеспечения. Анализ требований и определение спецификаций программного обеспечения. Проектирование программного обеспечения при структурном подходе.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- методики анализа требований заказчиков и пользователей;</li> <li>- процесс формализации требований, создание функциональных и нефункциональных спецификаций;</li> <li>- описание традиционного структурного подхода к проектированию программного обеспечения.</li> </ul>
6	<p>Проектирование программного обеспечения при объектном подходе. Средства проектирования программного обеспечения. Использование CASE-средств.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- принципы объектно-ориентированного проектирования;</li> <li>- основные конструкции объектно-ориентированных языков (класс, объект, интерфейс, наследование, полиморфизм);</li> <li>- использование автоматизированных CASE-средств для повышения эффективности разработки.</li> </ul>
7	<p>Тестирование и отладка программного обеспечения. Подходы к проектированию тестов.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- стратегии и методы тестирования программного обеспечения;</li> <li>- модульное тестирование;</li> <li>- интеграционное и функциональное тестирование;</li> <li>- рекомендации по созданию эффективного набора тестов.</li> </ul>
8	<p>Тестирование программного обеспечения. Отладка программного обеспечения.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- приёмы отладки программных комплексов;</li> <li>- инструменты и методы локализации ошибок;</li> <li>- алгоритмы обнаружения и устранения проблем в программе для повышения стабильности и работоспособности приложений.</li> </ul>
9	<p>Сопровождение программного обеспечения. Документирование программного обеспечения. Организация и технология сопровождения программного обеспечения. Защита программных продуктов.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- техническая поддержка готового программного продукта;</li> <li>- подходы к управлению обновлениями и устранением неисправностей;</li> <li>- правовые аспекты охраны авторских прав и лицензирования программного обеспечения.</li> </ul>
10	<p>Жизненный цикл программного обеспечения. Выявление требований к программной системе.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- жизненные циклы программного обеспечения;</li> <li>- важность точного формулирования требований и дальнейшего этапа анализа;</li> <li>- сравнение подходов Waterfall, V-модели, Spiral и Scrum.</li> </ul>
11	<p>Работа с заказчиком. Обзор методологий проектирования программных продуктов.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- коммуникация с заказчиком;</li> <li>- оформление запросов на разработку;</li> <li>- классические и альтернативные методологии проектирования программного обеспечения.</li> </ul>

№ п/п	Тематика лекционных занятий / краткое содержание
12	Технологии быстрой разработки программного обеспечения. Объектно-ориентированное проектирование программной системы. Рассматриваемые вопросы: - техники быстрого прототипирования и разработке минимально жизнеспособного продукта (MVP); - применение объектно-ориентированной парадигмы при проектировании сложных программных систем.
13	Средства информационной поддержки программных проектов и изделий (CALS)-технологий. Рассматриваемые вопросы: - возможности современных информационно-коммуникационных технологий для поддержки полного жизненного цикла программных продуктов; - преимущества использования CALS-технологий для интеграции бизнес-процессов предприятия.
14	Тестирование и отладка программных систем. Рассматриваемые вопросы: - тестирование и отладка крупных программных систем; - передовые подходы к автоматическому тестированию и контролю качества программного обеспечения.
15	Оценка качества программного обеспечения. Рассматриваемые вопросы: - объективная оценка качества программного обеспечения; - влияние различных критериев (надежность, эффективность, поддерживаемость, переносимость) на оценку конечного продукта; - систематизированный подход к оценке качества с точки зрения ISO стандартов.
16	Внедрение и сопровождение программных продуктов. Рассматриваемые вопросы: - переход от фазы разработки к фазе эксплуатации; - мероприятия, необходимые для успешной инсталляции, настройки и адаптации программного обеспечения на стороне клиента; - аспекты постпродажного обслуживания; - оптимизация программных продуктов в процессе их жизненного цикла.

#### 4.2. Занятия семинарского типа.

##### Лабораторные работы

№ п/п	Наименование лабораторных работ / краткое содержание
1	Разработка приложений, основанных на диалоге, в среде Visual C++ с использованием библиотеки MFC. - работа с библиотекой Microsoft Foundation Classes (MFC) для создания оконных приложений Windows; - создание диалоговых окон, обработка событий и элементов интерфейса (кнопки, поля ввода текста и меню).
2	Разработка SDI- приложений в среде Visual C++ с использованием библиотеки MFC. Часть 1. - начало изучения Single Document Interface (SDI) приложений; - знакомство с созданием простого окна приложения; - обработка сообщений; - управление окном главного документа.

№ п/п	Наименование лабораторных работ / краткое содержание
3	<p>Разработка SDI- приложений в среде Visual C++ с использованием библиотеки MFC. Часть 2.</p> <ul style="list-style-type: none"> <li>- создание графического редактора с возможностью сохранения и открытия документов;</li> <li>- работа с файлами и представление данных внутри SDI-документа.</li> </ul>
4	<p>Разработка SDI- приложений в среде Visual C++ с использованием библиотеки MFC. Часть 3</p> <ul style="list-style-type: none"> <li>- завершение освоения SDI-приложений;</li> <li>- добавление функционала печати;</li> <li>- добавление предварительного просмотра;</li> <li>- добавление масштабирования отображаемых документов.</li> </ul>
5	<p>Разработка MDI- приложений в среде Visual C++ с использованием библиотеки MFC. Часть 1.</p> <ul style="list-style-type: none"> <li>- изучение основ Multiple Document Interface (MDI) приложений;</li> <li>- начальное знакомство с архитектурой MDI;</li> <li>- открытие и закрытие дочерних окон.</li> </ul>
6	<p>Разработка MDI- приложений в среде Visual C++ с использованием библиотеки MFC. Часть 2</p> <ul style="list-style-type: none"> <li>- работа с несколькими окнами одновременно;</li> <li>- добавление панели инструментов и меню для переключения между окнами;</li> <li>- реализация механизма перетаскивания и группировки окон.</li> </ul>
7	<p>Разработка MDI- приложений в среде Visual C++ с использованием библиотеки MFC. Часть 3</p> <ul style="list-style-type: none"> <li>- создание полноценного MDI-приложения с поддержкой табуляции окон,</li> <li>- панелью инструментов и</li> <li>- средствами навигации по открытым документам.</li> </ul>
8	<p>Обследование системы, общение с заказчиком, планирование разработки, составление технического задания. Детальный анализ предметной области, принятие окончательного решения о необходимости создания информационной системы, проектирование общей архитектуры системы, выбор метода проектирования.</p> <ul style="list-style-type: none"> <li>- подготовка к процессу общения с клиентом;</li> <li>- определение потребностей бизнеса;</li> <li>- выявление технических ограничений;</li> <li>- постановка целей проекта;</li> <li>- составление технического задания.</li> </ul>
9	<p>Каскадные и итеративные технологии. Критичность и масштабность программных проектов.</p> <ul style="list-style-type: none"> <li>- исследование преимуществ и недостатков каскадных подходов к разработке программного обеспечения;</li> <li>- исследование преимуществ и недостатков итеративных подходов к разработке программного обеспечения;</li> <li>- решение задач по выбору подходящей методологии в зависимости от масштаба и критичности проекта.</li> </ul>
10	<p>Технология экстремального программирования. SCRUM технология. Преимущества и недостатки технологий быстрой разработки программного обеспечения. Организация коллективной работы над проектом при использовании технологий быстрой разработки.</p> <ul style="list-style-type: none"> <li>- изучение технологий Extreme Programming (XP) и SCRUM;</li> </ul>

№ п/п	Наименование лабораторных работ / краткое содержание
	- организация групповой работы по принципу Scrum; - демонстрация спринтов и проведение ретроспектив.
11	Построение объектно-ориентированной архитектуры системы. Методы объектно-ориентированного анализа для выявления классов и объектов. CASE-средства объектно-ориентированного проектирования. - применение объектно-ориентированного анализа для выделения классов и объектов; - составление диаграмм классов и последовательности действий; - разработка объектно-ориентированной архитектуры небольшого приложения.
12	Средства управления проектами. Применение данных средств при разработке и сопровождении программных продуктов. Использование средств коллективного владения кодом при создании корпоративных информационных систем. - изучение инструментов управления проектами; - настройка проекта, распределение задач и контроль сроков.
13	Стратегии и методы тестирования. Прямое и обратное тестирование. Программные средства автоматизации тестирования. - изучение прямого и обратного тестирования; - написание юнит-тестов и приемочных тестов; - подготовка теста для конкретного модуля программы.
14	Использование средств коллективного владения кодом при создании корпоративных информационных систем. - освоение системы контроля версий Git и сервиса GitHub; - совместное редактирование кода в группе.
15	Прямое и обратное тестирование. Программные средства автоматизации тестирования. - использование инструментов автоматического тестирования; - подготовка автотестов для веб-интерфейса.
16	Методики оценки качества ПО. Процессный подход к оценке качества ПО. - определение показателей качества программного обеспечения согласно стандартам ISO; - расчет индекса качества реального программного продукта; - формирование рекомендаций по улучшению качества.

#### 4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	1. Проработка конспекта лекций. 2. Подготовка к лабораторным работам. 3. Изучение учебной литературы из приведенных источников 4. Изучение методических рекомендаций, проработка соответствующих разделов учебника.
2	Выполнение курсовой работы.
3	Подготовка к промежуточной аттестации.
4	Подготовка к текущему контролю.

#### 4.4. Примерный перечень тем курсовых работ

Все перечисленные задания выполняются по персональным вариантам.

Задание состоит в разработке информационно-аналитической системы с использованием технологии объектно-ориентированного программирования.

Рекомендуемые темы:

1. «Библиотека»: В системе должны поддерживаться режимы поиска книги по заданному критерию (автор, название), заказа книги, учета клиентов и книг в книгохранилище, выдачи отчетов по запросам (местонахождение книги в архиве или ее отсутствие), выдачи документов о должниках.

2. «Магазин»: В системе должны поддерживаться режимы заказа товара (продовольственных товаров), покупки и учета товаров, анализа продажи товара, анализа продажи продуктов в зависимости времени дня и дня недели.

3. «Дом»: В системе должны поддерживаться режимы учета жильцов и учета доходов и расходов, связанных с проживанием жильцов в доме (аренда, электроэнергия, коммунальные услуги и т. д.), выдачи аналитической информации, выдачи списка жильцов.

4. «Успеваемость студентов на факультете»: В системе должны поддерживаться режимы учета учащихся и результатов сдачи экзаменов, анализа сессии по семестрам, по факультетам, специальностям, генерации отчетов отличников и двоечников.

5. «Конференция»: В системе должны поддерживаться режимы учета выступающих студентов, аспирантов и преподавателей по тематикам, по специальностям, регистрации участников и гостей на конференции, учета длительности и новизны тем, анализа конференции.

6. «Центр занятости»: В системе должны поддерживаться режимы учета безработных, их стажа, квалификации, желаний работать по определенной специальности, места расположения и заработной платы, учета уже стоящих на учете в центре занятости, анализа занятости от времени, специальности и т. д.

7. «Автобусный парк»: В системе должны поддерживаться режимы учета транспортных средств в автопарке, маршрутов, водителей, учета доходов и расходов (оплата за проезд, ремонт и т. п.), выдачи отчетов по запросам.

8. «Расписание занятий в университете»: В системе должны поддерживаться режимы поиска занятия по заданному критерию (время, преподаватель), регистрации занятий, учета занятий по типу, генерации расписаний.

9. «Школа»: В системе должны поддерживаться режимы учета классов и учеников в них, регистрации нового ученика, учета посещаемости занятий и оценок учащихся, генерации отчетов по успеваемости учеников.

10. «Библиотека университета»: В системе должны поддерживаться режимы поиска книги по заданному критерию (автор, название), заказа книги, учета клиентов и книг в книгохранилище, выдачи отчетов по запросам (местонахождение книги в архиве или ее отсутствие), выдачи документов о должниках.

11. «Магазин стройматериалов»: В системе должны поддерживаться режимы заказа товара, покупки и учета товаров, анализа продажи товара.

12. «Общежитие»: В системе должны поддерживаться режимы учета жильцов и учета доходов и расходов, связанных с проживанием жильцов (оплата, электроэнергия и т. д.), выдачи аналитической информации, выдачи списка жильцов.

13. «Успеваемость студентов в институте»: В системе должны поддерживаться режимы учета учащихся и результатов сдачи экзаменов, анализа сессии по семестрам, по специальностям, генерации отчетов отличников и двоечников.

14. «Конференция в университете»: В системе должны поддерживаться режимы учета выступающих студентов, аспирантов и преподавателей по тематикам, по специальностям, регистрации участников и гостей на конференции, учета длительности докладов, анализа конференции.

15. «Логистика»: В системе должны поддерживаться режимы учета транспортных средств в автопарке, маршрутов, водителей, учета доходов и расходов (оплата за проезд, ремонт и т. п.), выдачи отчетов по запросам.

16. «Расписание занятий в университете»: В системе должны поддерживаться режимы поиска занятия по заданному критерию (время, преподаватель), регистрации занятий, учета занятий по типу, генерации расписаний.

5. Перечень изданий, которые рекомендуется использовать при освоении дисциплины (модуля).

№ п/п	Библиографическое описание	Место доступа
1	Практикум «Программирование на языке Си» Шишкин А.Д., Чернецова Е.А. Методические указания СПб.: Российский государственный гидрометеорологический университет, 2013	ЭБС «IPRbooks» <a href="http://www.iprbookshop.ru/17958">http://www.iprbookshop.ru/17958</a>
2	Основы тестирования программного обеспечения Котляров В.П. Учебное пособие М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2013	ЭБС «IPRbooks» <a href="http://www.iprbookshop.ru/16095">http://www.iprbookshop.ru/16095</a>

6. Перечень современных профессиональных баз данных и информационных справочных систем, которые могут использоваться при освоении дисциплины (модуля).

1. <http://library.miit.ru/> — электронно-библиотечная система Научно-технической библиотеки МИИТ.

2. <http://rzd.ru/> — сайт ОАО «РЖД».

3. <http://elibrary.ru/> — научно-электронная библиотека.

4. <http://www.e-heritage.ru> — Электронная Библиотека «Научное Наследие России».

5. <http://citforum.ru> — on-line библиотека свободно доступных материалов по информационным технологиям на русском языке.

6. <http://www.rushelp.com> — компьютерная документация от А - Z.

7. <http://www.emanual.ru> — сайт, посвящённый всем значимым событиям в IT-индустрии: новейшие разработки, уникальные методы и горячие новости.

8. Поисковые системы: Nigma, Yandex, Google, Mail.

7. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, необходимого для освоения дисциплины (модуля).

Компьютеры должны быть обеспечены стандартными лицензионными программными продуктами Microsoft Office, Microsoft Visual Studio.

8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю).

Для проведения аудиторных занятий и самостоятельной работы требуется:

1. Рабочее место преподавателя с персональным компьютером, подключённым к сети Интернет.

2. Специализированная лекционная аудитория с мультимедиа аппаратурой и интерактивной доской.

3. Компьютерный класс.

9. Форма промежуточной аттестации:

Экзамен в 1 семестре.

Зачет во 2 семестре.

Курсовая работа во 2 семестре.

## 10. Оценочные материалы.

Оценочные материалы, применяемые при проведении промежуточной аттестации, разрабатываются в соответствии с локальным нормативным актом РУТ (МИИТ).

Авторы:

доцент, доцент, к.н. кафедры  
«Системы автоматизированного  
проектирования в строительстве»

О.В. Смирнова

Согласовано:

Заведующий кафедрой САП  
Председатель учебно-методической  
комиссии

И.В. Нестеров

М.Ф. Гуськова