МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ)

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

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Технологии программирования

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Направленность (профиль): Технологии разработки программного обеспечения

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 ФГАОУ ВО РУТ (МИИТ)
ID подписи: 5665
Подписал: заведующий кафедрой Нутович Вероника Евгеньевна
Дата: 01.09.2026

1. Общие сведения о дисциплине (модуле).

Цель дисциплины «Технологии программирования» ориентирована на формирование у студентов знаний о современных технологиях разработки алгоритмов и программ, о современных методах отладки программ, объектно-ориентированному программированию, объектно-ориентированному анализу и проектированию.

Задачами данной дисциплины является формирование у студента базовых знаний в областях об этапах создания программного продукта в рамках жизненного цикла, о современном состоянии технологий разработки программного продукта; познакомить обучающихся с существующими подходами к оценка качества процессов создания программного обеспечения; дать обучающемуся практические навыки проектирования программного обеспечения и расчета его надежности.

2. Планируемые результаты обучения по дисциплине (модулю).

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

ОПК-4 - Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью;.

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

Знать:

- принципы проектирования программных систем;
- организацию процесса проектирования программного обеспечения;
- методологию структурного проектирования программного обеспечения;
- технологические средства разработки программного обеспечения;
- методы отладки и тестирования программного обеспечения.

Уметь:

- использовать методы декомпозиции и абстракции при проектировании программного обеспечения;
- применять средства разработки программного обеспечения: инструментальные среды разработки, средства поддержки проекта, отладчики;
- документировать и оценивать качество программных продуктов.

Владеть:

- методами и средствами разработки и оформления технической документации;
- методами проектирования программного обеспечения при структурном и объектно-ориентированном подходе;
- методами структурного и функционального тестирования;
- методами совместной разработки приложений.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 4 з.е. (144 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 №3
Контактная работа при проведении учебных занятий (всего):	64	64
В том числе:		
Занятия лекционного типа	32	32
Занятия семинарского типа	32	32

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 80 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	Введение в технологии программирования. Рассматриваемые вопросы: - этапы развития технологий программирования; - современные языки программирования; - интегрированные среды разработки.
2	Качество программного обеспечения. Рассматриваемые вопросы: - критерии оценки качества программы; - отладка, тестирование и верификация программы; - виды ошибок.
3	Основы проектирования программного обеспечения. Рассматриваемые вопросы: - различные подходы к разработке программного обеспечения; - жизненный цикл программного продукта; - разработка технического задания; - предметная область и её описание; - проектирование программного обеспечения при объектно-ориентированном подходе к программированию.
4	UML. Рассматриваемые вопросы: - история развития; - эволюция версий; - структурные диаграммы; - диаграмма классов; - диаграмма компонентов; - диаграмма объектов; - диаграмма пакетов; - диаграмма композитной/сотавной структуры; - диаграмма кооперации; - диаграмма профилей; - диаграмма развертывания; - диаграммы поведения; - диаграмма деятельности; - диаграмма состояния; - диаграмма вариантов использования; - диаграммы взаимодействия; - диаграмма обзоров взаимодействия; - диаграмма последовательности; - диаграмма коммуникации; - диаграмма синхронизации.
5	Принципы S.O.L.I.D. Рассматриваемые вопросы: - принцип единственной ответственности; - принцип открытости/закрытости; - прицип подстановки Барбары Лисков; - принцип разделения интерфейсов; - принцип инверсии зависимостей.

№ п/п	Тематика лекционных занятий / краткое содержание
6	Паттерны проектирования. Рассматриваемые вопросы: - введение в паттерны программирования; - основные принципы; - основные паттерны; - порождающие паттерны; - структурные паттерны; - поведенческие паттерны; - паттерны и антипаттерны; - паттер Строитель; - паттерны Фабричный метод и Фабрика; - паттерн Репозиторий; - паттерн Адаптер.
7	Обеспечение качества программного обеспечения. Рассматриваемые вопросы: - тестирование и отладка программных продуктов; - оценка качества программного обеспечения.

4.2. Занятия семинарского типа.

Лабораторные работы

№ п/п	Наименование лабораторных работ / краткое содержание
1	Определение предметной области. В результате выполнения лабораторных работ студент получает навыки написания технического задания и описания предметной области.
2	Принципы SOLID. Принцип единственной ответственности (single responsibility principle). В результате выполнения лабораторных работ студент получает навыки разработки программного обеспечения с применением объектно-ориентированного подхода на основе принципа единственной ответственности (single responsibility principle).
3	Принципы SOLID. Принцип открытости/закрытости (open-closed principle). В результате выполнения лабораторных работ студент получает навыки разработки программного обеспечения с применением объектно-ориентированного подхода на основе принципа открытости/закрытости (open-closed principle).
4	Принципы SOLID. Принцип подстановки Лисков (Liskov substitution principle). В результате выполнения лабораторных работ студент получает навыки разработки программного обеспечения с применением объектно-ориентированного подхода на основе принципа подстановки Лисков (Liskov substitution principle).
5	Принципы SOLID. Принцип разделения интерфейса (interface segregation principle). В результате выполнения лабораторных работ студент получает навыки разработки программного обеспечения с применением объектно-ориентированного подхода на основе принципа разделения интерфейса (interface segregation principle).
6	Принципы SOLID. Принцип инверсии зависимостей (dependency inversion principle). В результате выполнения лабораторных работ студент получает навыки разработки программного обеспечения с применением объектно-ориентированного подхода на основе принципа инверсии зависимостей (dependency inversion principle).

№ п/п	Наименование лабораторных работ / краткое содержание
7	Проектирование программного обеспечения с помощью структурных диаграмм UML. В результате выполнения лабораторных работ по созданию структурных диаграмм UML (пакетов, классов, компонентов) студент получает навыки создания программных моделей со статическими параметрами при проектировании программного обеспечения при использовании объектно-ориентированного подхода с помощью UML-дизайнера.
8	Проектирование программного обеспечения с помощью диаграмм поведения UML. В результате выполнения лабораторных работ по созданию диаграмм поведения UML (вариантов использования, деятельности, состояний, последовательност) студент получает навыки создания программных моделей с динамическими параметрами при проектирования программного обеспечения при использовании объектно-ориентированного подхода с помощью UML-дизайнера.
9	Паттерны проектирования. Фабрика. В результате выполнения лабораторных работ студент получает навыки разработки программного обеспечения с применением объектно-ориентированного подхода используя паттерн проектирования Фабрика.
10	Паттерны проектирования. Фабричный метод. В результате выполнения лабораторных работ студент получает навыки разработки программного обеспечения с применением объектно-ориентированного подхода используя паттерн проектирования Фабричный метод.
11	Паттерны проектирования. Строитель. В результате выполнения лабораторных работ студент получает навыки разработки программного обеспечения с применением объектно-ориентированного подхода используя паттерн проектирования Строитель.
12	Паттерны проектирования. Адаптер. В результате выполнения лабораторных работ студент получает навыки разработки программного обеспечения с применением объектно-ориентированного подхода используя паттерн проектирования Адаптер.
13	Паттерны проектирования. Репозиторий. В результате выполнения лабораторных работ студент получает навыки разработки программного обеспечения с применением объектно-ориентированного подхода используя паттерн проектирования Репозиторий.
14	Оценка качества программного обеспечения. В результате выполнения лабораторных работ студент получает навыки структурного и функциональнго тестирования разработанного программного обеспечения.
15	Использование UML для проектирования параллельных приложений. В результате выполнения лабораторных работ студент получает навыки проектирования программных приложений с параллельными действиями и событиями на диаграмме деятельности.
16	Параллельные подавтоматы на диаграмме состояний UML. В результате выполнения лабораторных работ студент получает навыки описания составных состояний, входа и выхода из ортогональных состояний, взаимодействия подавтоматов.
17	Жизненный цикл программного обеспечения. В результате выполнения лабораторных работ студент получает навыки использования ГОСТ34 и других стандартов для составления программной документации на разных стадиях разработки.
18	Модели жизненного цикла. В результате выполнения лабораторных работ студент получает навыки выявления достоинств и недостатков различных видов моделей (каскадная, спиральная и комбинированные).

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение документации по языку UML и системе UML Designer
2	Анализ и дополнительная проработка лекционного материала
3	Подготовка к лабораторным работам
4	Изучение учебной литературы из приведенных источниковаммного обеспечения.
5	Выполнение курсовой работы
6	Подготовка к промежуточной аттестации.
7	Подготовка к текущему контролю.
8	Выполнение курсовой работы.
9	Подготовка к промежуточной аттестации.
10	Подготовка к текущему контролю.

4.4. Примерный перечень тем курсовых работ

1. Проектирование приложения с предметной областью «Блог».

2. Проектирование приложения с предметной областью «Университет».

3. Проектирование приложения с предметной областью «Автодиллер».

4. Проектирование приложения с предметной областью «Интернет-магазин».

5. Проектирование приложения с предметной областью «Каршеринг».

6. Проектирование приложения с предметной областью «Доставка еды».

7. Проектирование приложения с предметной областью «Доставка посылок».

8. Проектирование приложения с предметной областью «Заказ такси».

9. Проектирование приложения с предметной областью «Бронирование жилья»

10. Проектирование приложения с предметной областью «Оформление туристических поездок».

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

№ п/п	Библиографическое описание	Место доступа
1	Технология разработки программного обеспечения: учебное пособие Зубкова, Т. М. Санкт-Петербург : Лань , 2019	https://e.lanbook.com/book/122176

2	Технологии командной разработки программного обеспечения информационных систем Долженко, А. И. Москва : ИНТУИТ , 2016	https://e.lanbook.com/book/100515
3	Технология разработки программного обеспечения: учебное пособие Машкин А.В. Вологда : ВоГУ , 2014	https://e.lanbook.com/book/93087
4	Гибкая методология разработки программного обеспечения Москва : ИНТУИТ , 2016	https://e.lanbook.com/book/100590
5	Разработка моделей предметной области автоматизации : учебник для вузов С. В. Котлинский Учебник Санкт-Петербург : Лань , 2021	https://e.lanbook.com/book/183204 (дата обращения: 04.02.2022)
6	Методы верификации и оценки качества программного обеспечения : учебное пособие А. М. Семахин Учебное пособие Курган : КГУ , 2018	https://e.lanbook.com/book/177908 (дата обращения: 04.02.2022)
7	Проектирование программной системы в UML Designer М. А. Давыдовский, М. Н. Никольская. Учебное пособие М. : РУТ(МИИТ) , 2019	https://www.elibrary.ru/item.asp?id=41662371 (дата обращения: 03.10.2022)

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

- Научная электронная библиотека (http://elibrary.ru)
- Электронно-библиотечная система Научно-технической библиотеки МИИТ (http://library.miit.ru)
- Википедия (https://ru.wikipedia.org)
- Материалы по информационным технологиям (www.citforum.ru)
- Сайт системы автоматизированного проектирования UML Designer (http://www.umldesigner.org)
- Стандарты языка UML (http://www.omg.org/spec/UML)

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

- Язык программирования Java
- Язык программирования C++
- UML Designer (лицензия – свободно распространяемое ПО)

- Eclipse (лицензия – свободно распространяемое ПО)

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

Учебные аудитории для проведения учебных занятий, оснащенные компьютерной техникой и наборами демонстрационного оборудования.

Для практических занятий – наличие персональных компьютеров вычислительного класса.

9. Форма промежуточной аттестации:

Зачет в 3 семестре.

Курсовая работа в 3 семестре.

10. Оценочные материалы.

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

Авторы:

старший преподаватель кафедры
«Цифровые технологии управления
транспортными процессами» И.С. Разживайкин

старший преподаватель кафедры
«Цифровые технологии управления
транспортными процессами» Е.А. Заманов

Согласовано:

Заведующий кафедрой ЦТУТП В.Е. Нутович

Председатель учебно-методической
комиссии Н.А.Клычева