

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 - программы бакалавриата
по направлению подготовки
01.03.02 Прикладная математика и информатика,
утвержденной первым проректором РУТ (МИИТ)
Тимониным В.С.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Технологии программирования

Направление подготовки: 01.03.02 Прикладная математика и информатика

Направленность (профиль): Математическое моделирование и системный анализ

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: 5665
Подписал: заведующий кафедрой Нутович Вероника Евгеньевна
Дата: 24.05.2022

1. Общие сведения о дисциплине (модуле).

Дисциплина «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.

Целями освоения учебной дисциплины "Технологии программирования" являются:

- сформировать у студента базовые знания в областях об этапах создания программного продукта в рамках жизненного цикла, о современном состоянии технологий разработки программного продукта;
- познакомить обучающихся с существующими подходами к оценкам 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-2 - Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач;

ПК-3 - Уметь разрабатывать методики выполнения аналитических работ; планировать, организовывать и контролировать аналитические работы в информационно-технологическом проекте.

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

Знать:

Принципы проектирования программных систем; организацию процесса проектирования программного обеспечения; методологию структурного проектирования ПО; технологические средства разработки программного обеспечения; методы отладки и тестирования программ.

Уметь:

Использовать методы декомпозиции и абстракции при проектировании ПО; 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 3 з.е. (108 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	Сем. №3
Контактная работа при проведении учебных занятий (всего):	66	66
В том числе:		
Занятия лекционного типа	32	32
Занятия семинарского типа	34	34

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 42 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>Этапы развития технологий программирования.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none">- цели, задачи и содержание курса. Основные понятия и определения. Место и роль технологии программирования в программировании. Основные этапы развития программирования как науки;- проблемы разработки сложных информационных систем;- блочно-иерархический подход к созданию сложных информационных систем;- введение в программное обеспечение информационных систем.
2	<p>Обзор современных языков программирования и сред разработки.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none">- обзор современных языков программирования и сред разработки;- C++,- Java,- Python
3	<p>Критерии оценки качества программы.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none">- критерии оценки качества программы;- отладка, тестирование и верификация программы;- виды ошибок.
4	<p>Проектирование программного обеспечения при объектно-ориентированном подходе к программированию.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none">- 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.
5	<p>UML.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none">- история развития;- эволюция версий;- структурные диаграммы;- диаграмма классов;- диаграмма компонентов;- диаграмма объектов;- диаграмма пакетов;- диаграмма композитной/составной структуры;- диаграмма кооперации;- диаграмма профилей;- диаграмма развертывания;- диаграммы поведения;- диаграмма деятельности;- диаграмма состояния;

№ п/п	Тематика лекционных занятий / краткое содержание
	<ul style="list-style-type: none"> - диаграмма вариантов использования; - диаграммы взаимодействия; - диаграмма обзоров взаимодействия; - диаграмма последовательности; - диаграмма коммуникации; - диаграмма синхронизации.
6	<p>Принципы SOLID</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - принцип единственной ответственности; - принцип открытости/закрытости; - принцип подстановки Барбары Лисков; - принцип разделения интерфейсов; - принцип инверсии зависимостей.
7	<p>Введение в паттерны программирования. Основные принципы.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - 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.
8	<p>Тестирование и отладка программных продуктов.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - критерии оценки качества программы; - отладка, тестирование и верификация программы; - виды ошибок.

4.2. Занятия семинарского типа.

Лабораторные работы

№ п/п	Наименование лабораторных работ / краткое содержание
1	<p>Введение в технологию программирования.</p> <p>В результате выполнения лабораторных работ студент получает навыки написания технического задания и описания предметной области.</p> <p>В результате выполнения лабораторных работ студент получает навыки написания технического задания и описания предметной области.</p>
2	<p>Принципы SOLID.</p> <p>В результате выполнения лабораторных работ студент получает навыки разработки программного обеспечения с применением объектно-ориентированного подхода на основе принципа единственной ответственности (single responsibility principle).</p>
3	<p>Проектирование программного обеспечения.</p> <p>В результате выполнения лабораторных работ по созданию структурных диаграмм UML (пакетов,</p>

№ п/п	Наименование лабораторных работ / краткое содержание
	классов, компонентов) студент получает навыки создания программных моделей со статическими параметрами при проектировании программного обеспечения при использовании объектно-ориентированного подхода с помощью UML-дизайнера.
4	Анализ программного обеспечения. В результате выполнения лабораторных работ студент получает навыки выявления достоинств и недостатков различных видов ПО
5	Отладка, тестирование и верификация программы, виды ошибок. В результате выполнения лабораторных работ студент получает навыки отладки, структурного и функционального тестирования разработанного программного обеспечения.
6	Разработка технического задания. В результате выполнения лабораторных работ студент получает навыки написания технического задания и описания предметной области.
7	Проектирование программного обеспечения при объектно-ориентированном подходе к программированию. В результате выполнения лабораторных работ студент получает навыки разработки программного обеспечения с применением объектно-ориентированного подхода на основе принципа единственной ответственности (single responsibility principle).
8	Проектирование программного обеспечения при объектно-ориентированном подходе к программированию. В результате выполнения лабораторных работ студент получает навыки разработки программного обеспечения с применением объектно-ориентированного подхода на основе принципа открытости/закрытости (open-closed principle).
9	Паттерны Фабричный метод и Фабрика. В результате выполнения лабораторных работ студент получает навыки разработки программного обеспечения с применением объектно-ориентированного подхода используя паттерн проектирования Фабричный метод и Фабрика.
10	Паттерн Репозиторий. В результате выполнения лабораторных работ студент получает навыки разработки программного обеспечения с применением объектно-ориентированного подхода используя паттерн проектирования Репозиторий.
11	Тестирование и отладка программных продуктов. В результате выполнения лабораторных работ студент получает навыки структурного и функционального тестирования и отладки разработанного программного обеспечения.
12	Оценка качества программного обеспечения. В результате выполнения лабораторных работ студент получает навыки работы с критериями оценки качества программы; отладка, тестирование и верификация программы; изучает виды ошибок.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение дополнительной литературы. Введение в технологию программирования.
2	Изучение дополнительной литературы. Принципы SOLID.
3	Подготовка к лабораторным работам. Проектирование программного обеспечения.
4	Подготовка к лабораторным работам. Анализ программного обеспечения.
5	Выполнение курсовой работы.

6	Подготовка к промежуточной аттестации.
7	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».

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

№ п/п	Библиографическое описание	Место доступа
1	Кручинин, В. В. Технологии программирования: Учебное пособие. Москва : ТУСУР, 2013. - 271 с. - ISBN нет.	https://e.lanbook.com/book/110371
2	Зубкова, Т. М. Технология разработки программного обеспечения: учебное пособие. Санкт-Петербург : Лань, 2019. - 252 с. - ISBN 978-5-507-45571-3	https://e.lanbook.com/book/122176
3	Долженко, А. И. Технологии командной разработки программного обеспечения информационных систем. Москва : ИНТУИТ, 2016. - 300 с. - ISBN нет	https://e.lanbook.com/book/100515
1	Машкин А.В. Технология разработки программного обеспечения: учебное пособие. Вологда : ВоГУ, 2014. - 75 с. - ISBN 978-5-87851-526-9	https://e.lanbook.com/book/93087
2	Гибкая методология разработки программного обеспечения. Москва : ИНТУИТ, 2016. - 153 с. - ISBN нет	https://e.lanbook.com/book/100590
3	Шопырин Д.Г. Управление проектами разработки ПО. Дисциплина «Гибкие технологии разработки программного обеспечения». Санкт-Петербург : НИУ ИТМО, 2007. - 131 с. - ISBN нет	https://e.lanbook.com/book/43554

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

<http://library.miit.ru/> - электронно-библиотечная система Научно-

технической библиотеки МИИТ

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

Microsoft Office

Microsoft Visual Studio

JetBrain IntelliJ Idea

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.

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