

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
27.03.05 Инноватика,
утвержденной первым проректором РУТ (МИИТ)
Тимониным В.С.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Основы программирования

Направление подготовки: 27.03.05 Инноватика

Направленность (профиль): Управление инновациями (по отраслям и сферам экономики)

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: 2221
Подписал: заведующий кафедрой Тарасова Валентина
Николаевна
Дата: 06.04.2022

1. Общие сведения о дисциплине (модуле).

Целями освоения дисциплины (модуля) являются:

- изучение студентами основ программирования / кодирования / составления листинга учебного и прикладного программного обеспечения;
- изучение студентами основ алгоритмизации, декомпозиции задач на атомарные составляющие с последующим упорядочиванием их в корректной логической последовательности.

Задачами дисциплины (модуля) являются:

- развитие навыков разработки специфических прикладных программных продуктов для узкоспециальных нужд (задач профессиональной деятельности);
- формирование навыков и умений подготовки технической, проектной и предпроектной документации, относящейся к учебным и прикладным программным продуктам.

2. Планируемые результаты обучения по дисциплине (модулю).

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

ОПК-7 - Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности;

ОПК-10 - Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.

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

Знать:

- назначение процесса отладки программного обеспечения,
- правила вежливости программиста,
- суть линейных сдвигов кода и кодовых конструкций,
- синтаксис операторов,
- работу основных операторов и функций языка,
- структурное программирование,
- элементы объектно-ориентированного программирования,
- суть процесса подбора тестовых примеров к разработанному программному обеспечению.

Уметь:

- планировать вычислительный процесс с учётом неопытности и неграмотности пользователей авторского программного обеспечения (контроль ввода и вывода информации и побуждающих сообщений),
- читать программный код, составленный опытными программистами,
- грамотно комментировать программный код,
- вычищать программный код от избыточности,
- подготавливать тестовые примеры к разработанному программному обеспечению,
- программировать и отлаживать функциональные зависимости.

Владеть:

- навыками использования подключаемого модуля математики,
- методами конвертации типов данных,
- навыками косольного ввода/вывода,
- навыками ввода/вывода через интерфейсные элементы управления,
- навыками документирования и описания программного кода в различных графических нотациях (блок-схемы алгоритмов),
- навыками программирования и отладки программного обеспечения,
- навыками использования циклических конструкций для планирования и запуска имитационных экспериментов сферы профессиональной деятельности

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	Сем. №3
Контактная работа при проведении учебных занятий (всего):	96	96
В том числе:		
Занятия лекционного типа	32	32
Занятия семинарского типа	64	64

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 48 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	Составление отчётной документации по решённым задачам программирования Рассматриваемые вопросы: – структура типового отчёта; – связь структуры типового отчёта с жизненным циклом программного обеспечения; – этапы жизненного цикла программного обеспечения.
2	Основы алгоритмизации Рассматриваемые вопросы: – множество целей и задач окружающей действительности; – простые и сложные задачи; – декомпозиция задач; – понятие алгоритма.
3	Моделирование вычислительных процессов Рассматриваемые вопросы: – нотация структурных блок-схем алгоритмов (ГОСТ); – типовые блоки схем алгоритмов; – запрещённые связи на схемах алгоритмов; – оптимизация представления типовых моделей вычислительных процессов; – прочие модели вычислительных процессов; – векторный редактор Microsoft Office Visio.
4	Введение в интегрированную среду разработки (IDE) Рассматриваемые вопросы: – начало работы с интегрированной средой разработки Microsoft Visual Studio; – режимы работы интегрированной среды разработки; – понятие о целостности программного проекта / решения; – проверка языковых шаблонов и путей размещения проектов; – связь интегрированной среды разработки с системами контроля версий.

№ п/п	Тематика лекционных занятий / краткое содержание
5	Введение в контроль версий и системы контроля версий (VCS) Рассматриваемые вопросы: - понимание целей и задач системы контроля версий; - установка и настройка системы контроля версий GIT; - создание рабочих удалённых репозиториях для размещения выполненных работ; - установка и настройка графической оболочки для работы с удалёнными репозиториями системы контроля версий.
6	Консольный режим разработки. Базовые конструкции языка Visual C# Рассматриваемые вопросы: – понимание классово-объектной нотации при восприятии консоли интегрированной среды разработки; – шаблон консольного приложения и его составляющие; – понятие подключаемых к проекту / решению модулей; – понятие пространства имён; – понятие класса «Программа»; – понятие главного метода программы; – типовая задача консольного ввода; – типовая задача консольного вывода; – константы и переменные; – настройка консоли для удобства размещения результатов работы программных приложений в отчётной документации; – приостановка работы консоли и режимы работы, облегчающие отладку и просмотр результатов работы программ; – значимые типы данных; – ссылочные типы данных; – понятие строковой константы.
7	Типы вычислительных процессов. Последовательный вычислительный процесс. Присвоение. Рассматриваемые вопросы: – оператор присвоения как базовая единица, обуславливающая последовательный вычислительный процесс; – иллюстрирование последовательного вычислительного процесса на блок-схемах алгоритмов; – решение арифметических выражений; – использование стандартных тригонометрических и логарифмических функций и констант; – модуль «Математика».
8	Типы вычислительных процессов. Разветвляющийся вычислительный процесс. Условный оператор Рассматриваемые вопросы: – условный оператор как основной оператор разветвления вычислительного процесса; – понятие логического выражения; – структуры «если-то»; – структуры «если-то-иначе-то»; – структуры «если-то-иначе-если-то»; – структуры «если-если-то-иначе-то-иначе-то»; – построение условного оператора на блок-схемах алгоритмов.
9	Типы вычислительных процессов. Разветвляющийся вычислительный процесс. Оператор переключения. Рассматриваемые вопросы: – оператор переключения как средство оптимизированной и читаемой древовидности кода; – метки оператора переключения;

№ п/п	Тематика лекционных занятий / краткое содержание
	– переменная выбора оператора переключения; – оператор прерывания переключения; – метка обработки по умолчанию в операторе переключения; – построение оператора переключения на блок-схемах алгоритмов.
10	Типы вычислительных процессов. Циклический вычислительный процесс. Пред- и постусловие. Исследование функций. Рассматриваемые вопросы: – предусловие как основной инструмент управления необходимостью корректировки параметров; – предусловие как ключ необходимости выполнения однотипных действий при одновременном контроле ограничений; – постусловие как возможность корректировки параметров в условиях неопределённости; – построение цикла с предусловием на блок-схемах алгоритмов; – построение цикла с постусловием на блок-схемах алгоритмов; – понятие о параметрах циклов; – понятие о параметрах диапазона для построения графиков функциональных зависимостей (начало диапазона, конец диапазона, шаг).
11	Типы вычислительных процессов. Циклический вычислительный процесс. Диапазоны значений. Ряды данных и алгебраические ряды. Рассматриваемые вопросы: – понятие о параметрах диапазона, определяющих количество обрабатываемых шагов; – сокращение количества просматриваемых точек функциональных зависимостей за счёт возможности ввода не единичного шага циклической конструкции; – понятие о рядах данных; – понятие о разложении функции в ряд; – понятие рекуррентного члена ряда; – планирование знакопеременных рядов; – планирование знакопостоянных рядов; – использование параметров циклов для зависимых членов рядов.
12	Типы вычислительных процессов. Циклический вычислительный процесс. Множества значений. Списки Рассматриваемые вопросы: – работа со множествами; – перебор множеств; – организация списков; – списки как динамически наращиваемая структура однотипных данных; – обработка списков.
13	Структуры данных. Одномерные массивы Рассматриваемые вопросы: – аналогии массивов в реальной действительности; – способы ввода одномерных массивов; – способы корректного вывода одномерных массивов; – векторные операции; – параметры векторов; – связь с линейной алгеброй.
14	Структуры данных. Многомерные массивы Рассматриваемые вопросы: – аналогии массивов в реальной действительности; – способы ввода многомерных массивов; – способы корректного вывода многомерных массивов; – матричные операции;

№ п/п	Тематика лекционных занятий / краткое содержание
	– параметры матриц; – информационные гиперкубы и их проекции; – связь с линейной алгеброй.
15	Структуры данных. Строки Рассматриваемые вопросы: – понятие символа в программировании; – таблицы символов и кодировка символов; – строка как массив символов; – строка как универсальный контейнер; – строковые методы.
16	Подходы к оптимизации кода. Типизированные и нетипизированные методы Рассматриваемые вопросы: – назначение методов в программировании; – переход от привычных понятий функций и процедур к типизированным и нетипизированным методам; – модификаторы доступа методов; – оператор возврата значений; – рекурсивный вызов.

4.2. Занятия семинарского типа.

Лабораторные работы

№ п/п	Наименование лабораторных работ / краткое содержание
1	«Исследование фрагментов кода и составление блок-схем алгоритмов к ним» В результате выполнения лабораторной работы обучающиеся проводят исследование применимости типовых блоков схем алгоритмов к конкретным операторам языка программирования Visual C#.
2	«Исследование консольных методов и свойств» В результате выполнения лабораторной работы обучающиеся получают представление о консоли интегрированной среды разработки и о её возможностях.
3	«Исследование базовых значимых типов данных» В результате выполнения лабораторной работы, обучающиеся самостоятельно строят таблицы границ использования базовых значимых типов данных в обход использованию справочников. Убеждаются в том, что вся необходимая справочная информация о базовых конструкциях языка содержится внутри интегрированной среды разработки.
4	«Исследование работы арифметических операторов» В результате выполнения лабораторной работы обучающиеся исследуют поведение стандартный операторов языка и получают представление о соответствиях типов данных операторам.
5	«Исследование безопасного и небезопасного приведения типов данных» В результате выполнения лабораторной работы, обучающиеся получают представление о возможных ситуациях, связанных со вводом данных. Рассматривают поведение среды разработки при различных вводах данных и делают выбор безопасного ввода для большинства известных случаев ввода данных.
6	«Исследование составляющих модуля "Математика"» В результате выполнения лабораторной работы обучающиеся рассматривают полное множество методов, входящих в модуль «Математика», и получают представление о вычислительных возможностях языка программирования для решения чисто математических задач.
7	«Выполнение вычислений с наперёд заданной точностью» В результате выполнения лабораторной работы обучающиеся получают более чёткое представление об ограничениях разрядной сетки вычислителя, усваивают термин эпсилон-окрестности некоторой

№ п/п	Наименование лабораторных работ / краткое содержание
	точки.
8	«Переключение, управляемое данными различных типов» В результате выполнения лабораторной работы, обучающиеся проверяют применимость данных различного типа к формату оператора переключения. Анализируют возможности оператора переключения и возможные границы его применения.
9	«Безусловный переход и его аналоги» В результате выполнения лабораторной работы, обучающиеся получают возможность отыграть работу с оператором безусловного перехода и твёрдо усвоить, что хорошее программное обеспечение с прозрачно читаемым по коду алгоритмом строится без использования оператора безусловного перехода. Получают представление о том, что это не более чем временный, отладочный инструмент.
10	«Итерационный процесс и способы его прерывания» В результате выполнения лабораторной работы, обучающиеся осваивают циклический вычислительный процесс применительно ко вводу исходных данных. Получают представление о корректном вводе данных и способах добиваться от пользователя ввода корректных значений в разумных пределах с возможностью отказа от дальнейшего исполнения вычислительного процесса согласно ранее разработанному алгоритму.
11	«Анализ функциональной зависимости средствами программирования» В результате выполнения лабораторной работы обучающиеся закрепляют навыки математического анализа функциональных зависимостей и получают более чёткое представление о неразрывной связи программирования с математическим анализом.
12	«Циклические конструкции и их место в инженерных задачах» В результате выполнения лабораторной работы, обучающиеся принимают решения о рациональном выборе и использовании конкретной циклической конструкции к конкретной решаемой задаче. Получают представление об избыточности кода и устранении избыточности в случае ошибочно принятых решений по выбору циклической конструкции.
13	«Одномерные массивы и стандартные методы их обработки» В результате выполнения лабораторной работы обучающиеся осваивают работу с векторами и методами, входящими в состав типа данных «массив», что способствует закреплению и более чёткому пониманию разделов линейной алгебры.
14	«Способы корректного ввода / вывода одномерных и многомерных массивов» В результате выполнения лабораторной работы обучающиеся привыкают к определённой нотации ввода и вывода массивов, что в дальнейшем делает разрабатываемое ими программное обеспечение более аккуратным с точки зрения форматирования побуждающих сообщений, вывода промежуточных результатов для отладки и предоставления конечных результатов в удобном для восприятия виде.
15	«Исследование основных строковых методов» В результате выполнения лабораторной работы обучающиеся изучают формат методов, входящих в состав типа данных string (строка), начинают уверенно чувствовать себя с синтаксисом этих методов и понимать необходимые наборы входной информации, которые они должны подготавливать, если планируют использование этих методов в разрабатываемом программном обеспечении.
16	«Составление интерактивного текстового меню» В результате выполнения лабораторной работы обучающиеся приобретают навыки проектирования в первом приближении простейшего графического пользовательского интерфейса.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	«Настройка персонального рабочего пространства в системе контроля версий» В результате выполнения работы, обучающиеся создают собственное рабочее пространство для размещения выполненных работ в системе контроля версий, что приближает освоение курса к

№ п/п	Тематика практических занятий/краткое содержание
	реальному взаимодействию разработчиков и заказчиков программного обеспечения.
2	«Составление схем семантических алгоритмов» В результате работы, обучающиеся приобретают навыки декомпозиции задач и представления её решения в виде упорядоченной логической последовательности шагов.
3	«Организация консольного ввода / вывода» В результате выполнения работы, обучающиеся приобретают навыки модификации базового шаблона консольного режима разработки программного обеспечения для нужд решения большинства типовых вычислительных задач курса.
4	«Подбор тестовых примеров в задачах программирования» В результате выполнения работы, обучающиеся приобретают навыки подбора тестовых примеров, необходимых для доказательства работоспособности программного обеспечения. Получают представление о необходимом и достаточном количестве тестовых примеров для конкретной задачи предметной области.
5	«Решение сложных арифметических выражений в среде Microsoft Visual Studio на языке Visual C#» В результате выполнения работы обучающиеся приобретают навыки планирования и ввода переменных различного типа, начинают различать операции и методы по типам данных.
6	«Условный оператор. Учёт наложенных ограничений» В результате работы обучающиеся приобретают навыки выявления прямых и косвенных ограничений, наложенных на функциональные зависимости, и преобразования их в код.
7	«Условный оператор. Решение задач, имеющих геометрическую интерпретацию» В результате работы обучающиеся приобретают навыки трансформирования геометрических задач в программный код, собственно, решающий подобные задачи для различных наборов исходных данных.
8	«Разветвление вычислительного процесса переключением» В результате работы обучающиеся приобретают навыки использования оператора переключения для ёмкого разветвления вычислительного процесса.
9	«Циклический вычислительный процесс. Построение графиков функциональных зависимостей» В результате работы обучающиеся приобретают навыки табличного представления графиков функциональных зависимостей.
10	«Циклический вычислительный процесс. Работа с рядами данных» В результате работы обучающиеся приобретают навыки планирования изменяющихся по определённому правилу значений переменных внутри циклических конструкций.
11	«Одномерные массивы» В результате работы обучающиеся приобретают навыки использования одномерных массивов для удобного хранения данных.
12	«Многомерные массивы» В результате работы обучающиеся приобретают навыки использования многомерных массивов для удобного хранения данных.
13	«Строковые методы» В результате работы обучающиеся приобретают навыки использования стандартных строковых методов.
14	«Решение типовых задач строковой контейнеризации» В результате работы обучающиеся приобретают навыки декомпозиции, модификации и агрегации строк.
15	«Типизированные методы» В результате работы обучающиеся приобретают навыки проектирования типизированных методов.

№ п/п	Тематика практических занятий/краткое содержание
16	«Нетипизированные (пустые) методы» В результате работы обучающиеся приобретают навыки проектирования нетипизированных методов.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение дополнительной литературы.
2	Подготовка к практическим занятиям.
3	Подготовка к лабораторным работам.
4	Подготовка к промежуточной аттестации.
5	Подготовка к текущему контролю.
6	Подготовка к промежуточной аттестации.
7	Подготовка к текущему контролю.

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

№ п/п	Библиографическое описание	Место доступа
1	Построение графиков и диаграмм в Excel Антон Игоревич Сафронов, Н.Н. Зольникова, Людмила Николаевна Логинова Книга 2017	http://elibrary.ru
2	Составление отчётной документации по решённым задачам алгоритмизации и программирования Антон Игоревич Сафронов, Н.Н. Зольникова, Вячеслав Геннадьевич Новиков Книга 2018	http://elibrary.ru
3	Получение первичных профессиональных умений и навыков научно-исследовательской деятельности Антон Игоревич Сафронов, Н.Н. Зольникова, Вячеслав Геннадьевич Новиков Книга 2019	http://elibrary.ru
4	«CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C#» Джеффри Рихтер	https://vk.com/wall-54530371_183
5	Основы программирования на языке C# : учебное пособие для бакалавров, обучающихся по направлению "220400 - Управление в технических системах" профиль "Управление и информатика в технических системах" / В. Г. Сидоренко, Д. И. Харчилин ; В. Г. Сидоренко, Д. И. Харчилин ; ФГБ ОУ ВПО "Московский гос. ун-т путей сообщ.", Каф. "Упр. и информатика в технических системах". Сидоренко, В. Г. Москва.: МИИТ , 2011	http://elibrary.ru

6	Применение системы контроля версий GIT для организации учебного процесса в вузе / М. А. Васильева // Современные технологии в науке и образовании - СТНО-2021 : Сборник трудов IV Международного научно-технического форума: в 10 т., Рязань, 03–05 марта 2021 года Васильева, М. А. Рязань: Рязанский государственный радиотехнический университет имени В.Ф.Уткина , 2021	http://elibrary.ru
7	Искусство программирования [4 тома] Дональд Кнут	https://vk.com/wall-51126445_68593

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

- Официальный сайт РУТ (МИИТ) (<https://www.miit.ru/>).
- Научно-техническая библиотека РУТ (МИИТ) (<http://library.miit.ru>).
- Национальная электронная библиотека (<http://elibrary.ru>).
- Электронно-библиотечная система издательства «Лань» (<http://e.lanbook.com/>).
- Электронно-библиотечная система ibooks.ru (<http://ibooks.ru/>).

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

- Microsoft Internet Explorer (или любой другой браузер).
- Операционная система Microsoft Windows.
- Пакет офисных приложений Microsoft Office.
- Двухпанельный файловый менеджер Total Commander (Double Commander).
- Менеджер командной строки системы контроля версий GIT.
- Графическая оболочка системы контроля версий GIT Extensions.
- Интегрированная среда разработки (IDE) прикладного программного обеспечения Microsoft Visual Studio.

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

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

9. Форма промежуточной аттестации:

Экзамен в 3 семестре.

10. Оценочные материалы.

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

Авторы

Доцент, к.н. кафедры «Управление и
защита информации»

Сафронов Антон
Игоревич

Лист согласования

Заведующий кафедрой УиЗИ

Л.А. Баранов

Заведующий кафедрой УИТ

В.Н. Тарасова

Председатель учебно-методической
комиссии

С.В. Володин