

Белорусский государственный университет



УТВЕРЖДАЮ

Проректор по учебной работе и
образовательным инновациям

О. Г. Прохоренко
_____ О. Г. Прохоренко

1 декабря 2023 г.

Регистрационный № УД -727/6.

МАШИННО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

**Учебная программа учреждения высшего образования
по учебной дисциплине для специальностей:**

6-05-0533-11 Прикладная информатика

6-05-0533-12 Кибербезопасность

2023 г.

Учебная программа составлена на основе ОСВО 6-05-0533-11-2023 специальности 6-05-0533-11 «Прикладная информатика», примерного учебного плана, регистрационный № 6-05-05-029/пр. от 30.01.2023, учебных планов БГУ: №6-5.3-59/03 от 15.05. 2023 г., №6-5.3-59/04 от 15.05. 2023 г., №6-5.3-59/05 от 15.05. 2023 г., №6-5.3-59/01ин. от 31.05. 2023 г., №6-5.3-59/02ин. от 31.05. 2023 г., №6-5.3-59/03ин. от 31.05. 2023 г.;

на основе ОСВО 6-05-0533-12-2023 специальности 6-05-0533-12 «Кибербезопасность», примерного учебного плана, регистрационный № 6-05-05-029/пр. от 30.01.2023, учебного плана БГУ № 6-5.3-60/02 г.

СОСТАВИТЕЛИ:

Н.К. Рубашко – старший преподаватель кафедры информационных систем управления Белорусского государственного университета

В.В. Конах – старший преподаватель кафедры информационных систем управления Белорусского государственного университета

РЕЦЕНЗЕНТЫ:

А.А. Воронов – ведущий научный сотрудник ОИПИ НАН Беларуси, кандидат технических наук, доцент

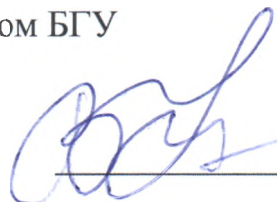
С.В. Гафуров – заместитель директора ИПУП «Байлекс Малтимедиа»

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой информационных систем управления Белорусского государственного университета (протокол № 3 от 19.10.2023 г.).

Научно-методическим Советом БГУ (протокол №3 30.11.2023 г.)

Заведующий кафедрой



В.В. Краснопрошин

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная дисциплина «Машинно-ориентированное программирование» ориентирована на обучение студентов программированию на языке ассемблера для процессоров серии x86 — самых распространенных в современных персональных компьютерах. Изучаемые темы базируются на использовании современных информационных технологий, новейшего программного и технического обеспечения компьютеров.

Основой для обучения являются компетенции, сформированные при изучении дисциплины «Основы и методология программирования».

Цели и задачи учебной дисциплины

Цель преподавания учебной дисциплины «Машинно-ориентированное программирование» – дать студенту понимание того, как язык ассемблера связан с физической архитектурой компьютера, как организовать работу с регистрами процессора, реализовать основные программные конструкции, скомпилировать и запустить законченную программу на языках любого уровня в той части задач, где существенно проявляется зависимость параметров качества программы от учета особенностей архитектуры компьютера.

При изложении курса важно показать возможности использования инструментария программирования на машинно-ориентированных языках при решении прикладных задач, возникающих в различных областях науки, техники, экономики и производства.

Задачи учебной дисциплины

Основной задачей, решаемой при изучении учебной дисциплины «Машинно-ориентированное программирование», является подготовка специалиста, умеющего писать на языке программирования низкого уровня компактные, быстрые и эффективные программы.

Место учебной дисциплины в системе подготовки специалиста с высшим образованием

Учебная дисциплина относится к модулю «Программирование» государственного компонента.

Программа составлена с учётом межпредметных связей с учебными дисциплинами.

Дисциплина «Машинно-ориентированное программирование» непосредственно связана с параллельно изучаемыми дисциплинами:

- «Линейная алгебра», «Математический анализ» модуля «Высшая математика» для специальности 6-05-0533-12 «Кибербезопасность»;
- «Алгебра и теория чисел», «Математический анализ» модуля «Высшая математика» для специальности 6-05-0533-11 «Прикладная информатика»;
- «Разработка кросс-платформенных приложений» модуля «Программирование», «Дискретная математика и математическая логика»

модуля «Дискретная математика и алгоритмы» для специальностей 6-05-0533-12 «Кибербезопасность» и 6-05-0533-11 «Прикладная информатика».

Методы, излагаемые в указанных дисциплинах, используются для проектирования алгоритмов и программных приложений.

Сформированные при изучении дисциплины «Машинно-ориентированное программирование» компетенции являются основой для дальнейшего изучения дисциплин:

- «Промышленное программирование» модуля «Программирование»;
- «Алгоритмы и структуры данных» модуля «Дискретная математика и алгоритмы»;
- «Операционные системы» модуля «Информатика и компьютерные системы».

Знания, полученные в учебной дисциплине, используются при изучении всех дисциплин специализации, при выполнении курсовых и дипломных работ, а также используются как инструментарий для моделирования и компьютерного решения задач ряда математических дисциплин, изучаемых на старших курсах.

Требования к компетенциям

Освоение учебной дисциплины «Основы и методологии программирования» должно обеспечить формирование следующих **универсальных компетенций**:

для специальности 6-05-0533-11 «Прикладная информатика»:

УК-5. Быть способным к саморазвитию и совершенствованию в профессиональной деятельности.

УК-6. Проявлять инициативу и адаптироваться к изменениям в профессиональной деятельности.

для специальности 6-05-0533-12 «Кибербезопасность»:

УК-1. Владеть основами исследовательской деятельности, осуществлять поиск, анализ и синтез информации.

УК-2. Решать стандартные задачи профессиональной деятельности на основе применения информационно-коммуникационных технологий.

УК-4. Работать в команде, толерантно воспринимать социальные, этнические, конфессиональные, культурные и иные различия.

УК-5. Быть способным к саморазвитию и совершенствованию в профессиональной деятельности.

УК-6. Проявлять инициативу и адаптироваться к изменениям в профессиональной деятельности.

Освоение учебной дисциплины «Основы и методологии программирования» должно обеспечить формирование следующей **базовой профессиональной компетенции** для обеих специальностей:

БПК-2. Строить, анализировать и тестировать алгоритмы и программы решения типовых задач обработки информации с использованием структурного, объектно-ориентированного и иных парадигм программирования.

В результате освоения учебной дисциплины студент должен:

знать:

- механизмы ассемблирования и организацию ассемблерных программ;
- методы представления данных в вычислительных машинах и преобразования между внешним и внутренним представлением данных;
- методы эффективного использования знания архитектуры компьютера для оптимизации программ, средства управления компьютером через программно доступные компоненты;
- базовые программно-технические решения при создании программных моделей средств вычислительной техники;

уметь:

- программировать задачи средней сложности на ассемблере;
- оценивать эффективность различных машинно-ориентированных программно-технических решений в программах на языках высокого уровня;
- разрабатывать программные приложения с заданной функциональностью и операционным окружением;

владеть:

- навыками управления вычислительными процессами при программировании на ассемблере;
- навыками организации взаимодействия программных модулей, разрабатываемых на ассемблере, с модулями, разрабатываемыми на языках высокого уровня;
- навыками программирования обработки числовой, логической и текстовой информации.

Структура учебной дисциплины

Дисциплина изучается во 2-м семестре. Всего на изучение учебной дисциплины «Машинно-ориентированное программирование» отведено:

- для очной формы получения высшего образования – 90 часов, в том числе 34 аудиторных часа, из них: лекций – 16 часов, лабораторных занятий – 16 часов, управляемая самостоятельная работа – 2 часа.

Трудоемкость учебной дисциплины составляет 3 зачетные единицы.

Форма промежуточной аттестации – зачет.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. Введение в машинно-ориентированное программирование

Тема 1.1. Системы счисления

Понятие машинно-ориентированного программирования. Системы счисления. Двоичная и шестнадцатеричная системы счисления как основа функционирования компьютеров. Бит, байт. Перевод чисел из одной системы счисления в другую. Выполнение арифметических операций. Логические данные.

Тема 1.2. Принципы функционирования микропроцессоров

Архитектура компьютера. Принципы фон Неймана. Принципиальная схема компьютера на базе процессора x86. Режимы работы компьютера. Механизм исполнения машинных команд. Регистры процессора x86. Флаги. Организация памяти. Сегменты. Модели памяти. Представление целочисленных данных и адресов в памяти компьютера. Принципы работы со стеком.

Раздел 2. Основы языка Ассемблера

Тема 2.1. Синтаксис языка Ассемблера

Машинные команды и команды Ассемблера. Механизм исполнения машинных команд. Задание операндов в машинных командах. Форматы данных. Представление целочисленных данных и адресов в памяти компьютера. Непосредственная и регистровая адресация. Адресация в памяти.

Тема 2.2. Команды целочисленной арифметики в Ассемблере

Команды пересылки данных, сложения, вычитания, умножения, деления. Ограничения на операнды. Команды преобразования чисел со знаком. Команды изменения знака, увеличения размеров. Команды условной пересылки.

Побитовые команды. Команды сдвига. Команды работы со стеком.

Тема 2.3. Типовые управляющие структуры

Безусловный переход. Условные переходы. Сравнение данных. Ограничения на операнды.

Организация вычислительных циклов с помощью команды `loop` и с помощью команд условного и безусловного перехода.

Тема 2.4. Сложные структуры данных

Типы косвенной адресации. Работа с массивами.

Строковые команды. Префиксы повторения в строковых командах.

Раздел 3. Модульное программирование в Ассемблере

Тема 3.1. Процедуры в языке Ассемблера

Процедуры и организация связей между процедурами на языке Ассемблера. Механизм вызова процедур и возврата из процедур. Передача параметров через регистры и общие области. Передача параметров через стек. Использование регистра ЕВР для доступа к параметрам. Соглашения о передаче параметров. Описание локальных переменных в процедурах.

Тема 3.2. Связь Ассемблера с языками высокого уровня

Встраиваемый ассемблерный код. Внешний ассемблерный код. Ассемблер и С++. Понятие соглашения о вызовах. Наиболее распространенные соглашения о вызовах cdecl, stdcall, fastcall. Декорирование имен (namedecoration). Упрощенное namedecoration. Требования к оформлению процедур на Ассемблере, вызываемых из программ на С++.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Очная форма получения высшего образования с применением
дистанционных образовательных технологий (ДОТ)

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСР	Форма контроля знаний
		Лекции	Практические занятия	Семинарские	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
	Машинно-ориентированное программирование	16			16		2	
1.	Введение в машинно-ориентированное программирование	2			2			
1.1.	Системы счисления				2			Экспресс-опрос
1.2.	Принципы функционирования микропроцессоров	2						Экспресс-опрос
2.	Основы языка Ассемблера	10			8		2	
2.1.	Синтаксис языка Ассемблера	2			2			Собеседование
2.2.	Команды целочисленной арифметики в Ассемблере	2			2			Экспресс-опрос
2.3.	Типовые управляющие структуры	2			2			Расчетно-графическая работа №1
2.4.	Сложные структуры данных	4			2		2	Собеседование Расчетно-графическая работа №2 Коллоквиум по материалам раздела 2
3.	Модульное программирование в Ассемблере	4			6			
3.1.	Процедуры в языке Ассемблера	2			2			Электронный тест
3.2.	Связь Ассемблера с языками высокого уровня	2			4			Расчетно-графическая работа №3 Контрольная работа по темам 3.1 - 3.2

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Перечень основной литературы

1. Бунаков, П Ю. Машинно-ориентированные языки программирования. Введение в ассемблер: учебное пособие / П. Ю. Бунаков. - Санкт-Петербург; Москва; Краснодар: Лань, 2023. - 143 с. - URL: <https://e.lanbook.com/book/302627>.
2. Йо Ван Гуй. Программирование на ассемблере x64: от начального уровня до профессионального использования AVX / Йо Ван Гуй ; [пер. с англ. А. В. Снастина]. - Москва: ДМК Пресс, 2021. - 331 с.
3. Таненбаум, Э. Архитектура компьютера = Structured Computer Organization / Э. Таненбаум, Т. Остин; [пер. с англ. Е. Матвеев]. - 6-е изд. - Санкт-Петербург [и др.]: Питер, 2020. - 811 с. - URL: <https://ibooks.ru/bookshelf/361850/reading>.

Перечень дополнительной литературы

1. Бочкарева, В. В. Языки программирования низкого уровня: Практикум: учебное пособие / В. В. Бочкарева. – Москва: РТУ МИРЭА, 2021. – 83 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/176501>.
2. Дубков, Виталий Павлович. Программирование на Ассемблере: учеб.-метод. пособие для студ. фак. прикладной математики и информатики: в 2 ч. Ч. 1 / В.П. Дубков, О.М. Кондратьева, В.Ю. Сакович; БГУ, Фак. прикладной математики и информатики, Каф. математического обеспечения ЭВМ. – Минск: БГУ, 2010.
3. Ассемблер – это просто. Учимся программировать. – 2-е изд., перераб. и доп. / О.А.Калашников–СПб.: БХВ-Петербург, 2011. – 336 с.
4. Assembler. Учебник для вузов. 2-е изд. / В.И. Юров – СПб.: Питер, 2010. – 637 с.
5. Ассемблер для процессоров IntelPentium. / Ю.С.Магда – СПб.: Питер, 2006. – 410 с.
6. Ассемблер на примерах. Базовый курс / Рудольф Марек – СПб: Наука и Техника, 2005. – 240 с.
7. Искусство программирования на Ассемблере. Лекции и упражнения / Н.Г.Голубь – СПб.: ООО “ДиаСофтЮП”, 2002. – 656 с.
8. Программирование на языке ассемблера IBM PC. / В.Н.Пильщиков. – М.: “ДИАЛОГ-МИФИ”, 1996. – 288 с.

Электронные ресурсы

1. Куляс, О. Л. Курс программирования на Assembler : учебное пособие / О. Л. Куляс, К. А. Никитин. –Москва: СОЛОН-ПРЕСС, 2017. – 220 с. – ISBN 978-5-91359-245-3. – Текст: электронный // Электронно-

- библиотечная система IPR BOOKS: [сайт]. –URL:
<http://www.iprbookshop.ru/80561.html>.
2. Образовательный портал БГУ [Электронный ресурс]. – Режим доступа:
<https://edufpmi.bsu.by/course/view.php?id=839>.

Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки

Для диагностики компетенций в рамках учебной дисциплины рекомендуется использовать следующие формы:

1. Устная форма: собеседование, экспресс-опрос.
2. Письменная форма: контрольные работы, коллоквиум.
3. Устно-письменная форма: отчеты по расчетно-графическим работам с их устной защитой.
4. Техническая форма: электронный тест.

В качестве рекомендуемых технических средств диагностики используется обучение, организованное на платформе Moodle (<https://edufpmi.bsu.by>).

Формой промежуточной аттестации по дисциплине учебным планом предусмотрен зачет.

Точки контроля по текущей успеваемости формируются из расчета общего количества часов (зачетных единиц), выделенных на изучение дисциплины.

Критерием оценивания является выполнение заданий для управляемой самостоятельной работы и расчетно-графических работ. Задания и лабораторные работы по результатам выполнения и защиты оцениваются с учетом следующих основных параметров:

- своевременное выполнение работы;
- полнота и правильность ответов на вопросы, заданные в ходе защиты работы.

В случае выполнения данных условий студент имеет возможность сдавать зачет по теоретическим вопросам с решением задач.

Отметка «зачтено» выставляется студенту, который твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Отметка «не зачтено» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, допускающему в ответе или в решении задач грубые ошибки.

Примерный перечень заданий для управляемой самостоятельной работы студентов

В качестве заданий для управляемой самостоятельной работы могут быть выданы задания для самостоятельного решения задач по следующим темам:

Тема 2.4. Сложные структуры данных. Строковые команды. Префиксы повторения в строковых командах. (2 ч.).

Форма контроля – собеседование.

Примерная тематика лабораторных занятий

Занятие 1. Системы счисления: десятичная, двоичная и шестнадцатеричная. Арифметические действия.

Занятие 2. Машинные команды и команды Ассемблера. Основы языка Ассемблера. Команды Ассемблера: пересылки данных, сложения, вычитания, умножения, деления.

Занятие 3. Безусловный переход. Условные переходы. Сравнение данных. Решение задач целочисленной арифметики.

Занятие 4. Типы косвенной адресации. Обработка массивов.

Занятие 5. Побитовые команды. Команды сдвига. Работа со стеком.

Занятие 6. Механизм вызова подпрограмм и возврата из подпрограмм. Передача параметров. Соглашения о передаче параметров. Описание локальных переменных в подпрограммах.

Занятие 7. Связь ассемблера и языков программирования высокого уровня. Взаимодействие ассемблерных процедур с языком программирования C++.

Занятие 8. Контрольная работа.

Рекомендуемая тематика контрольных работ и коллоквиума

- 1) Коллоквиум «Основы языка Ассемблера».
- 2) Контрольная работа «Связь ассемблера и языка программирования высокого уровня C++».

Описание инновационных подходов и методов к преподаванию учебной дисциплины

При организации образовательного процесса используются следующие методы:

– **метод группового обучения**, который представляет собой форму организации учебно-познавательной деятельности обучающихся, предполагающую функционирование разных типов малых групп, работающих как над общими, так и специфическими учебными заданиями.

В качестве технических средств для организации работы в рамках учебной дисциплины рекомендуется использовать Образовательный портал БГУ (<https://edufpmi.bsu.by>) – инструмент с эффективной функциональностью контроля, тренинга и самостоятельной работы.

– **практико-ориентированный подход**, который предполагает освоение содержания образования через решения практических задач; приобретение

навыков эффективного выполнения разных видов профессиональной деятельности; ориентацию на генерирование идей, реализацию групповых студенческих проектов; использование процедур, способов оценивания, фиксирующих профессиональные компетенции.

Методические рекомендации по организации самостоятельной работы обучающихся

Для организации самостоятельной работы студентов по учебной дисциплине следует использовать современные информационные ресурсы: разместить на образовательном портале комплекс учебных и учебно-методических материалов (учебно-программные материалы, учебное издание для теоретического изучения дисциплины, методические указания к лабораторным занятиям, материалы текущего контроля и текущей аттестации, позволяющие определить соответствие учебной деятельности обучающихся требованиям образовательных стандартов высшего образования и учебно-программной документации, в том числе вопросы для подготовки к зачету и экзамену, задания, тесты, вопросы для самоконтроля и др., список рекомендуемой литературы, информационных ресурсов и др.).

Задания УСР по учебной дисциплине рекомендуется делить на три модуля:

1. Задания, формирующие достаточные знания по изученному учебному материалу **на уровне узнавания**:
 - a) Как символы строк размещаются в оперативной памяти?
 - b) Как определить, что строка обрабатывается по 1/2/4 байта?
 - c) Как определить направление обработки строк (слева направо или наоборот)?
 - d) Где находятся операнды команд обработки строк?
 - e) Для чего используются префиксы гер/гере/герне?
 - f) Как определяется длина обрабатываемой строки?
 - g) Как после команды сканирования определить, что искомое найдено?
2. Задания, формирующие компетенции **на уровне воспроизведения**:
 - a) Определить, является ли строка палиндромом.
 - b) Удалить из строки все символы, равные заданному.
 - c) Строка представляет собой слова, разделенные пробелами. Вычислить количество слов.
 - d) Строка представляет собой слова, разделенные пробелами. Удалить лишние пробелы.
3. Задания, формирующие компетенции **на уровне применения полученных знаний**:
 - a) В строке найти все слова, начинающиеся с прописной буквы латинского алфавита.
 - b) Подсчитать частоту встречаемости символов в строке.
 - c) Отсортировать слова в строке по алфавиту.

- d) Распечатать слова столбиком в порядке возрастания их длин.

Примерный перечень вопросов к зачету

1. Регистры процессора. Типы регистров.
2. Регистры общего назначения.
3. Регистр флагов. Команды изменения флагов.
4. Работа со стеком. Команды работы со стеком.
5. Представление целых чисел в двоичной системе счисления.
6. Адресация данных в машинных командах. Типы адресации.
7. Команды сложения и вычитания.
8. Операторы преобразования длины.
9. Команды пересылки данных.
10. Классификация команд управления программным потоком.
11. Условные переходы.
12. Команды условной пересылки.
13. Организация циклов с помощью переходов.
14. Организация циклов с помощью команды LOOP и ее модификаций.
15. Команды умножения целых чисел.
16. Команды деления целых чисел.
17. Команды для работы с битами.
18. Команды линейного сдвига.
19. Команды циклического сдвига.
20. Доступ к элементам одномерных массивов.
21. Доступ к элементам двумерных массивов.
22. Строковые команды. Префиксы повторения в строковых командах.
23. Механизм вызова процедур. Команды для работы с процедурами.
24. Способы передачи параметров в процедуры.
25. Проблемы передачи параметров через стек и способы их решения.
26. Использование регистра EBP для доступа к параметрам.
27. Организация работы с локальными переменными.
28. Вызов ассемблерных процедур из программ на C++.
29. Соглашения о связях.

Примерный перечень заданий к зачету

Разработать многомодульное приложение для решения следующей задачи:

1. Ввести целое число. Ввод числа организовать из головного модуля на C++. Определить, является ли модуль этого числа степенью 2. Если да, то вывести эту степень (Ассемблер). Вывести результат (C++).
2. Ввести целое число. Ввод числа организовать из головного модуля на C++. Вычислить сумму цифр модуля этого числа (Ассемблер). Вывести результат (C++).
3. Ввести целое число. Ввод числа организовать из головного модуля на C++. Из записи числа выбросить все четные цифры, оставив прежним

порядок остальных цифр (Ассемблер). Вывести результат (C++).

4. Ввести целое число. Ввод числа организовать из головного модуля на C++. Определить количество четных цифр числа (Ассемблер). Вывести результат (C++).

5. Ввести целое число. Ввод числа организовать из головного модуля на C++. Определить целую часть среднего арифметического цифр в записи числа (Ассемблер). Вывести результат (C++).

6. Ввести два целых числа. Ввод числа организовать из головного модуля на C++. Определить, является ли модуль одного из них степенью модуля второго. Если да, то вывести эту степень (Ассемблер). Вывести результат (C++).

7. Ввести массив целых чисел. Ввод организовать из головного модуля на C++. Найти сумму положительных чисел массива (Ассемблер). Вывести результат (C++).

8. Ввести массив целых чисел и число X . Ввод организовать из головного модуля на C++. Определить, есть ли число X в массиве (Ассемблер). Вывести результат (C++).

9. Ввести массив целых чисел. Ввод организовать из головного модуля на C++. Отсортировать массив по возрастанию (Ассемблер). Вывести результат (C++).

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Разработка кросс-платформенных приложений	Информационных систем управления	Нет	Изменений не требуется (протокол № 3 от 19.10.2023 г.).
Операционные системы	Информационных систем управления	Нет	Изменений не требуется (протокол № 3 от 19.10.2023 г.).
Промышленное программирование	Информационных систем управления	Нет	Изменений не требуется (протокол № 3 от 19.10.2023 г.).
Технологии программирования	Информационных систем управления	Нет	Изменений не требуется (протокол № 3 от 19.10.2023 г.).

**ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ
ПО ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ**

на ____ / ____ учебный год

№№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры информационных систем управления (протокол № ____ от _____ 202_ г.)

Заведующий кафедрой

_____ (степень, звание) _____ (подпись) _____ (И.О.Фамилия)

УТВЕРЖДАЮ
Декан факультета

_____ (степень, звание) _____ (подпись) _____ (И.О.Фамилия)