

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
Димитровградский инженерно-технологический институт –
филиал федерального государственного автономного образовательного учреждения высшего
образования «Национальный исследовательский ядерный университет «МИФИ»
(ДИТИ НИЯУ МИФИ)

УТВЕРЖДАЮ:

Заместитель руководителя

_____ Т.И. Романовская

« ____ » _____ 2018г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Основы алгоритмических языков программирования

Специализация _____ *14.05.01 Ядерные реакторы и материалы*

Квалификация выпускника _____ *инженер-физик*

Профиль _____ *Ядерные реакторы*

Форма обучения _____ *Очная*

Выпускающая кафедра _____ *Ядерных реакторов и материалов*

Кафедра-разработчик рабочей программы _____ *Информационных технологий*

Семестр	Трудоемкость час. (ЗЕТ)	Лекций, час.	Практич. занятий, час.	Лаборат. работ, час.	СРС, час.	Форма промежуточного контроля (экз., час./зачет)
6	108(3)	34	34	-	40	зачёт
Итого	108(3)	34	34	-	40	зачёт

Димитровград
2018 г.

СОДЕРЖАНИЕ

1 ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	3
2 МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО.....	3
3 ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	4
4 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ.....	5
5 ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ.....	10
6 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ВХОДНОГО И ТЕКУЩЕГО КОНТРОЛЯ, ПРОМЕЖУТОЧНОЙ И ИТОГОВОЙ АТТЕСТАЦИИ (АННОТАЦИЯ).....	10
7 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	10
8 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	11
9 ТЕХНОЛОГИЧЕСКАЯ КАРТА ДИСЦИПЛИНЫ.....	11

1 ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Учебная дисциплина «Основы алгоритмических языков программирования» обеспечивает приобретение знаний и умений в соответствии с федеральным государственным Образовательным стандартом, содействует формированию мировоззрения и системного мышления.

Основной целью дисциплины «Основы алгоритмических языков программирования» - ввести студентов в проблематику, связанную с языками программирования, методами разработки алгоритмов и программ, методами реализации языков программирования.

Задачи дисциплины «Основы алгоритмических языков программирования»:

- формирование базовых понятий о языках и методах программирования;
- приобретение теоретических знаний и практических навыков по разработке, тестированию, отладке программных продуктов;
- приобретение практических навыков применения современных методов программирования.

2 МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО

Дисциплина Основы алгоритмических языков программирования относится к *вариативной* части блока блока 1 общепрофессионального модуля учебного плана.

Необходимыми условиями для освоения дисциплины являются: *знание* базовых понятий информатики и вычислительной техники, роли и значения информатики в современном обществе, форм представления и преобразования информации в компьютере; *умения* применять вычислительную технику для решения практических задач, оперировать элементами алгебры логики;

владение навыками работы на персональном компьютере.

Таблица 2.1 - Перечень предшествующих и последующих дисциплин, формирующих общекультурные и профессиональные компетенции

Код	Наименование компетенции	Предшествующие дисциплины	Последующие дисциплины (группы дисциплин)
Профессиональные компетенции			
ПК-4	способностью применять экспериментальные, теоретические и компьютерные методы исследований в профессиональной области	Физика Физический практикум Атомная физика Химия и химический практикум Экология Ядерная физика Электротехника и электроника Основы электротехники Сопротивление материалов	Квантовая механика и статистическая физика Численные методы Материаловедение: материалы ядерных установок Исследовательские реакторы Методы и приборы физических измерений Курсовой проект: проектирование и выбор оборудования ядерных энергетических установок, безопасность и экономичность ядерных энергетических установок Асимптотические методы в физике Надежность и безопасность ядерных энергетических установок Учебная практика Преддипломная практика Итоговая государственная ат-

			тестация
ПК-6	способностью самостоятельно выполнять экспериментальные или теоретические исследования для решения научных и производственных задач с использованием современной техники и методов расчета и исследования	Физика Физический практикум Атомная физика Химия и химический практикум Теоретическая механика Экология Сопротивление материалов Основы электротехники Электротехника и электроника	Квантовая механика и статистическая физика Численные методы Ядерная физика Материаловедение: материалы ядерных установок Исследовательские реакторы Методы и приборы физических измерений Курсовой проект: проектирование и выбор оборудования ядерных энергетических установок, безопасность и экономичность ядерных энергетических установок Асимптотические методы в физике Динамика и безопасность ядерных энергетических установок Надежность и безопасность ядерных энергетических установок Методы проектирования Преддипломная практика Итоговая государственная аттестация

3 ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс изучения дисциплины направлен на формирование элементов компетенций в соответствии с ОС НИЯУ МИФИ и ОП ВО по данному направлению подготовки (специальности).

Таблица 3.1 - Перечень планируемых результатов обучения по дисциплине

Планируемые результаты освоения ОП (компетенции), достижение которых обеспечивает дисциплина		Перечень планируемых результатов обучения по дисциплине
Код компетенции	Содержание компетенции	
ПК-4	способностью применять экспериментальные, теоретические и компьютерные методы исследований в профессиональной области	Знать основные принципы использования в профессиональной деятельности базовых знаний в области естествознания, информатики и современных информационных технологий, использования ресурсов Интернет; Уметь: применять на практике методы разработки алгоритмов и программ; Владеть: навыками применения языков программирования
ПК-6	способностью самостоятельно выполнять экспериментальные или теоретические исследования для решения научных и производственных задач с использованием современной техники и методов расчета и исследования	Знать современные тенденции развития информатики и вычислительной техники, компьютерных технологий; Уметь выбирать и применять конкретные информационные технологии; Владеть навыками самоорганизации и саморазвития.

4 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Структура дисциплины

Общая трудоемкость (объем) дисциплины составляет 3 зачетных единиц (ЗЕТ), 108 академических часов.

Таблица 4.1

Объём дисциплины по видам учебных занятий (в соответствии с учебным планом)

Вид учебной работы	Всего, зачетных единиц (акад. часов)	Семестр
Общая трудоемкость дисциплины	108	6
Контактная работа с преподавателем:		
занятия лекционного типа	32	
в том числе: практические занятия	32	
групповые консультации		
индивидуальные консультации		
Самостоятельная работа обучающихся:	44	
изучение теоретического курса	25	
задачи	19	
Вид промежуточной аттестации	зачёт	

Распределение учебной нагрузки по разделам дисциплины

Таблица 4.2

№ раздела	Наименование раздела дисциплины	Виды учебной нагрузки и их трудоемкость, акад. часы				Формируемые компетенции
		Лекции	Практические занятия	Самостоятельная работа	Всего часов	
1	Введение. Основы алгоритмики	2	2	10	14	ПК-4,6
2	Основы программирования на языке Си	20	20	15	55	ПК-4,6
3	Модульное программирование	4	4	9	17	ПК-4,6
4	Графическая библиотека Borland C	6	6	10	22	ПК-4,6
Итого		32	32	44	108	

4.2 Содержание дисциплины

Удельный вес проводимых в активных и интерактивных формах проведения аудиторных занятий по дисциплине составляет _____ %.

Лекционный курс

Таблица 4.3

№ лекции	Номер раздела	Тема лекции и перечень дидактических единиц	Трудоемкость, акад. часов	
			всего	в том числе с использованием интерактивных образовательных технологий
1	1	Введение. Основы алгоритмики. Обзор языков высокого уровня. Место языка Си/Си++ в вычислительной технике. Обзор литературы по языку Си. Повторение основных понятий информатики. Системы счисления и переходы между ними. Шестнадцатиричная арифметика. Представление отрицательных чисел. Системы кодирования символов. Способы представления целых и вещественных чисел. Понятие алгоритма. Основная задача алгоритмизации и порядок составления алгоритмов. Понятие структурного программирования и схемы алгоритма программы. Способы записи алгоритма.	2	
2-11	2	Основы программирования на языке Си	20	
2	2.1	Исторический обзор. История создания и развития языка Си. Основные преимущества Си перед другими языками программирования высокого уровня. Основные особенности работы в Си. Обзор компиляторов языка. Стандарты языка Си. Основные понятия теории программирования. Основные принципы написания программ. Структура программы на языке Си. Понятие выражения. Основные этапы решения задач на ЭВМ. Критерии качества программы. Жизненный цикл программы. Постановка задачи и спецификация программы.	2	
3	2.2	Стандартные типы данных. Имена переменных. Типы и размеры данных. Описания переменных. Символьные и строчные константы, константные выражения. Понятие операции. Пример простейшей программы на Си. Таблица приоритетов операций. Арифметические операции. Основные понятия математической логики. Операции отношения и логические операции. Преобразование типов. Операции уменьшения и увеличения. Побитовые логические операции. Операции и выражения присваивания. Условные выражения. Старшинство и порядок присваивания.	2	
4-5	2.3	Операторы управления. Представление основных управляющих структур программирования. Операторы и блоки. Операторы if-else и else-if. Переключатель. Цикл for. Оператор цикла с предусловием while-do. Оператор цикла с постусловием do-while. Операторы безусловного перехода и метки. Пустой оператор. Операторы break и continue. Среда программирования Borland C. Общие сведения. Назначение и состав среды программирования Borland C. Правила работы с файлами. Работа с меню и настройка среды. Компиляция и запуск программ. Основы работы системы отладки программ.	4	
6-7	2.4	Массивы. Определение массива. Одномерные и двумерные массивы. Размещение массива в памяти. Особенности индексации массивов в Си. Многомерные массивы. Утверждения о массивах.	4	
8-9	2.5	Функции. Основные сведения. Аргументы функции: передача по адресу, передача по значению. Синтаксис функций.	4	

		Описание функций, вызов функций, прототип функций. Внешние переменные. Область действия. Статические и регистровые переменные. Инициализация переменных.		
10-11	2.6	Указатели. Указатели и адреса. Указатели и аргументы функций. Указатели и массивы. Адресная арифметика. Многомерные массивы. Массивы указателей. Инициализация массивов указателей. Командная строка аргументов. Указатели на функции.	4	
12-13	3	Модульное программирование Понятия библиотечных и объектных файлов. Правила создания библиотеки пользователя. Понятие оверлея. Правила создания и использования оверлейных блоков.	4	ДК-3 ОПК-2
14-16	4	Графическая библиотека Основные библиотечные функции. Инициализация графического режима. Функции установки, рисования, заливки.	6	ДК-3 ОПК-2
Итого:			18	

Практические занятия

Таблица 4.3

№ занятия	Номер раздела	Наименование практического занятия и перечень дидактических единиц	Трудоёмкость, акад. часов	
			всего	в том числе с использованием интерактивных образовательных технологий
1	1	Ознакомление со средой Borland C++. Цель практического занятия – ознакомление со средой Borland C++.	2	
2		Общие сведения о языке C++. Цель практического занятия : Необходимо составить блок-схему линейного алгоритма и написать программу в среде Borland C++ 3.1 по заданному варианту.	2	
3		Основные конструкции в языке C++. Программирование условных операторов и переключателей Цель практического занятия : Необходимо составить блок-схему разветвляющего алгоритма и написать программу в среде Borland C++ 3.1 по заданному варианту.	2	
4-5		Основные конструкции в языке C++. Программирование циклов. Цель практического занятия : Необходимо составить блок-схему циклического алгоритма и написать программу в среде Borland C++ 3.1 по заданному варианту	4	
6-7		Создание и использование функций. Цель практического занятия : Необходимо составить блок-схему смешанного алгоритма и написать программу в среде Borland C++ 3.1 по заданному варианту	4	
8		Обработка массивов в C++. Цель практического занятия : Необходимо составить блок-схему смешанного алгоритма и написать программу в среде Borland C++ 3.1 по заданному варианту с обязательным применением массивов.	2	
9-10		Указатели и динамические массивы. Цель практического занятия : Необходимо составить блок-схему смешанного алгоритма и написать программу в среде Borland C++ 3.1 по заданному варианту	4	

11		Обработка матриц в C++. Цель практического занятия : Необходимо составить блок-схему смешанного алгоритма и написать программу в среде Borland C++ 3.1 по заданному варианту с обязательным применением динамических матриц.	2	
12		Организация ввода-вывода в C++. Цель практического занятия : Необходимо составить блок-схему смешанного алгоритма и написать программу в среде Borland C++ 3.1 по заданному варианту. Первая программа должна сформировать двоичный файл. Вторая – считать данные из этого файла, выполнить соответствующие вычисления (при этом используемые массивы должны быть динамическими) и записать их результаты в текстовый файл.	2	
13		Обработка строк. Цель практического занятия : Необходимо составить блок-схему смешанного алгоритма и написать программу в среде Borland C++ 3.1 по заданному варианту.	2	
14		Структуры. Задание: Цель практического занятия : Необходимо составить блок-схему смешанного алгоритма и написать программу в среде Borland C++ 3.1 по заданному варианту с обязательным использованием структур. В программе должны быть также реализованы функции для создания двоичного файла с информацией, добавления туда новой информации, обработки информации и просмотра информации с выводом на экран.	2	
15-16		Объектно-ориентированное программирование. Цель практического занятия : Необходимо составить блок-схему смешанного алгоритма и написать программу в среде Borland C++ 3.1 по заданному варианту с обязательным использованием классов. Программа должна состоять из двух файлов: один с описанием классов и методов (с обязательными конструкторами и деструкторами), а второй – с демонстрацией работы всех возможностей класса на примере конкретных объектов. Для всех вариантов предусмотреть разработку конструктора и деструктора.	4	
Итого:			32	

Самостоятельная работа студента

Самостоятельная работа позволяет закрепить теоретические знания и практические навыки, полученные студентом на лекциях и практических занятиях.

Таблица 4.4

Раздел дисциплины	№ п/п	Вид самостоятельной работы студента (СРС) и перечень дидактических единиц	Трудоемкость, часов
1	1.1	Понятие алгоритма. Основная задача алгоритмизации и порядок составления алгоритмов. Понятие структурного программирования и схемы алгоритма программы. Способы записи алгоритма. Начало и конец программного блока. Понятие действия. Условное	4

		обозначение действия на схемах алгоритма. Условие как основной тип ветвления алгоритмов и программ. Блок выбора как расширенный вариант блока условия. Понятие цикла. Индекс цикла, переменная цикла, диапазон, шаг. Представление цикла с помощью оператора условия. Цикл с индексом, цикл с условием в начале, цикл с условием в конце. Области применимости и основные различия между тремя конструкциями циклов. Правила работы с вложенными циклами. Служебные блоки комментариев.	
2	2.1.	Теорема структуры и структурное программирование. Утверждения о программах. Корректность программ. Правила вывода для основных структур программирования. Инвариантные утверждения. Этапы преобразования программы. Создание ассемблерного кода программы с помощью компилятора командной строки.	3
	2.2.	Классы памяти переменных. Области действия, видимости и продолжительности жизни переменных и функций. Особенности статических и глобальных переменных. Выход переменных из области видимости	4
	2.3.	Области, выделяемые программе при ее выполнении. Область кода. Область данных, глобальных и статических переменных. Стек. Динамически распределяемая память. Определение физических адресов и размеров областей памяти с помощью отладчика.	3
	2.4.	Рекурсия. Рекурсивные определения и алгоритмы. Понятие стека вызовов функций. Ширина и глубина стека вызовов. Особенности использования стека программы в случае применения рекурсивных функций. Индуктивные функции на последовательностях. Программирование рекурсивных алгоритмов.	5
	2.5.	Препроцессор языка Си. Включение файлов и директива include. Макроопределение и макроподстановка. Прагмы препроцессора. Параметризованные макросы. Условная прекомпиляция.	3
	2.6.	Структуры. Основные сведения. Структуры и записи. Массивы структур. Указатели на структуры. Структуры, ссылающиеся сами на себя. Поиск в таблице. Битовые поля. Объединения. Передача структур в функции. Динамические структуры данных. Линейные списки, основные виды и способы реализации. Стеки, деревья, односвязные и двусвязные списки, очереди, деки. Линейный список как абстрактный тип данных. Структуры и функции времени. Определение и изменение системного времени и даты. Определение дня недели.	4
3	3.1.	Модульное программирование Использование проектов. Преимущества проектной организации программы.	4
4	4.1.	Инициализация графического режима. Работа с пикселями. Копирование и перемещение образов. Графические окна. Работа с текстом и шрифтами. Рисование графиков функций и соответствие между математической и графической системами координат.	8
	4.2.	Организация графического ввода и вывода текстовой и числовой информации.	6
		Итого	444

5 ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Рекомендуемые образовательные технологии: лабораторные занятия, самостоятельная работа студентов.

Для обучения применяются активные и интерактивные формы обучения (разбор конкретных ситуаций, обсуждение отдельных разделов дисциплины). В сочетании с внеаудиторной работой это способствует формированию и развитию профессиональных навыков обучающихся.

Для закрепления знаний студентов проводятся лабораторные занятия, целью которых является формирование навыков самостоятельной работы по решению некоторых задач. Также для овладения и повторения материалов курса студенты выполняют тестовые задания по пройденному материалу.

В рамках учебного курса предусмотрена встреча с представителями ведущих IT-компаний и организация с их стороны для студентов мастер-классов.

6 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ВХОДНОГО И ТЕКУЩЕГО КОНТРОЛЯ, ПРОМЕЖУТОЧНОЙ И ИТОГОВОЙ АТТЕСТАЦИИ (АННОТАЦИЯ)

Входной контроль осуществляется в форме тестирования с целью определения базовых знаний студента и выявления разделов дисциплины, вызывающих наибольшие затруднения у студентов.

Контроль освоения дисциплины производится в соответствии с Положением о рейтинговой системе оценки знаний студентов ДИТИ НИЯУ МИФИ.

Текущий контроль студентов по дисциплине производится в дискретные временные интервалы в следующих формах:

- выполнение лабораторных работ;
- защита лабораторных работ;
- устные опросы;

Дополнительно оцениваются личностные качества студента: умение работать в коллективе, своевременная сдача тестов, выполнение и защита лабораторных работ, выполнение самостоятельной работы.

Промежуточный контроль студентов производится в форме тестирования:

Итоговый контроль по дисциплине проходит в форме устного опроса по экзаменационным билетам, включающим в себя ответ на теоретический вопрос, решение задач, выполнение практического задания на ПК.

Фонды оценочных средств, включают в себя типовые задания, тесты и методы контроля, позволяющие оценить результаты обучения по данной дисциплине, перечислены в Приложении.

7 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1 Перечень основной и дополнительной учебной литературы

Таблица 7.1 - Обеспечение дисциплины основной и дополнительной литературой по дисциплине

N п/п	Автор	Название	Место издания	Наименование издательства	Год издания	Количество экземпляров
Основная литература						
1	Фридман А. Л.	Язык программирования Си++. Курс лекций : учебное пособие / под ред. А Л. Фридман .	Москва	Интернет-Университет Информационных Технологий	2010	2

2	Орлов С А	Теория и практика языков программирования Учебник для вузов	Санкт-Петербург	Питер	2012	3
Дополнительная литература						
1	Костюкова Н.И. / Н. И. Костюкова, Л. А. Калинина	Язык Си и особенности работы с ним: учебное пособие	Москва	Интернет-Университет Информационных Технологий	2009	2
2	Подбельский, В. В.	Программирование на языке Си : Учебное пособие	Москва	Финансы и статистика	2005	5
3	Керниган, Брайан У.	Язык программирования С : пер. с англ. / Керниган, Брайан У., Ритчи, Деннис	Москва	Вильямс	2009	10

7.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Электронная библиотечная система (ЭБС) Книгафонд <http://www.knigafund.ru/>
2. Центр информационно-библиотечного обеспечения учебно-научной деятельности НИЯУ МИФИ <http://www.library.mephi.ru/>
3. Научная электронная библиотека <http://elibrary.ru/defaultx.asp>
4. Электронно-библиотечная система «Лань» <http://elibrary.ru/defaultx.asp>

8 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

1. Для проведения лекционных занятий используется:
 - комплект электронных презентаций/слайдов;
 - компьютерный класс, оснащенный презентационной техникой (проектор, интерактивная доска, компьютер).
2. Для проведения лабораторных работ используется:
 - компьютерный класс, оснащенный презентационной техникой (проектор, интерактивная доска, компьютер);
 - пакеты прикладного программного обеспечения MS Office/

9 ТЕХНОЛОГИЧЕСКАЯ КАРТА ДИСЦИПЛИНЫ

Технологическая карта рейтинговых баллов по дисциплине « Основы алгоритмических языков программирования »

Специальность 14.05.01 «Ядерные реакторы»

3 курс дневное обучение

Максимальное количество баллов за работу в течение семестра: 60 баллов.

Итоговый контроль: 40 баллов

Семестр 6

Всего часов _____

в том числе:

1 лекции - 32 часов;

2 лабораторные работы - часов;

- 3 семинарские / практические занятия - 32 часов;
 4 подготовка к лекциям - _____ часов;
 5 подготовка к семинарским / практическим занятиям - _____ часов;
 6 подготовка к лабораторным работам - _____ часов;
 7 подготовка к экзамену / зачету - _____ часов;
 8 творческая самостоятельная работа (за исключением пп. 4 – 7) - _____ часов

Структура текущего и промежуточного контроля.

Информация о контр. точках	Текущий контроль(<=25) (ТК)									Промежуточный контроль (<=30) (ПК)		Форма итогового контроля
	ТК ₁	ТК ₂	ТК ₃	ТК ₄	ТК ₅	ТК ₆	ТК ₇	ТК ₈	ТК ₈	ПК ₁	ПК ₂	
форма контроля	Л/ПР ₁	Л/ПР ₂	Л/ПР ₃	Л/ПР ₄	Л/ПР ₅	Л/ПР ₆	Л/ПР ₇	Л/ПР ₈	Л/ПР ₉	КР	КР	Э
неделя сдачи	2	4	6	7	10	12	13	15	18	8	14	
макс. балл	3	3	3	2,5	2,5	2,5	2,5	2,5	3,5	15	15	

Структура баллов, начисляемых студентам по результатам текущего контроля (промежуточного контроля)

№ п/п	Наименование видов учебной работы	Начисляемое количество баллов (долей баллов)	Максимальное количество баллов по данному виду учебной работы
1.	Посещение практических занятий и выполнение заданий на практических занятиях	9 работ по 1 баллу	9
2.	Оформление конспектов по практическим занятиям	3 работы по 2 балла 5 работ по 1,5 баллов 1 работа по 2,5 балла	16
Максимальная сумма баллов по результатам текущего контроля			25

ПЕРЕЧЕНЬ домашних заданий и видов самостоятельной работы студентов

№ п/п	Темы домашних заданий и самостоятельной работы	Недели семестра, в которых будет выдаваться задание	Недели семестров, в которых будут приниматься отчеты по домашним заданиям и работам
1.	Построение схем ветвящихся и циклических алгоритмов. (ПР№1-2)	2	2
2.	Программирование условных операторов и переключателей. (ПР№3)	4	5
3.	Программирование циклов. (ПР№4)	6	6
4.	Создание и использование функций. (ПР№5)	8	7
5.	Работа с одномерными и двумерными массивами. (ПР№6)	7	8

6.	Работа со структурами. Создание простой базы данных. (ПР№7)	9	10
7.	Работа с файлами, директориями и дисковым пространством. (ПР№8)	12	13
8.	Использование динамических типов данных в форме двусвязных списков. (ПР№9)	15	17

**Дополнения и изменения в рабочей программе
дисциплины на 20__/20__ уч.г.**

Внесенные изменения на 20__/20__ учебный год

УТВЕРЖДАЮ

Декан _____ факультета

(в состав которого входит кафедра-составитель)

« ____ » _____ 20 __ г.

В рабочую программу вносятся следующие изменения:

- 1)
- 2)

или делается отметка о нецелесообразности внесения каких-либо изменений на данный учебный год

Рабочая программа пересмотрена на заседании кафедры

(дата, номер протокола заседания кафедры, подпись зав. кафедрой).

СОГЛАСОВАНО:

Заведующий выпускающей кафедрой

наименование кафедры личная подпись расшифровка подписи дата

Декан

наименование факультета, где производится обучение, личная подпись расшифровка подписи дата

Начальник УМУ

личная подпись расшифровка подписи дата

Аннотация рабочей программы

Дисциплина Б1.В.ДВ.5.2 «Основы алгоритмических языков программирования» является базовой частью общепрофессионального модуля дисциплин подготовки студентов по направлению подготовки бакалавров 14.05.01 «Ядерные реакторы». Дисциплина реализуется на информационно-технологическом факультете ДИТИ НИЯУ МИФИ кафедрой информационных технологий.

Цели и задачи дисциплины: изучение студентами различных парадигм программирования, современных языков и методов программирования. Основными изучаемыми парадигмами для данного курса являются структурное (модульное), объектно-ориентированное и визуальное программирование; выработка у студентов умения самостоятельно разрабатывать алгоритмы и составлять программы для решения прикладных задач, используя при этом наиболее подходящие для данной задачи языки и методы программирования.

Требования к уровню освоения содержания дисциплины. В результате изучения курса студенты должны: знать различные парадигмы программирования; знать принципы организации данных, способы и алгоритмы управления данными; знать возможности конкретных языков программирования и их библиотек; владеть современными технологиями разработки и программирования алгоритмов; владеть практическими навыками разработки и реализации алгоритмов с использованием наиболее подходящих структур данных и языка программирования; уметь работать в современных интегрированных средах; уметь документировать все этапы разработки программного продукта.

Дисциплина нацелена на формирование следующих компетенций выпускника:

профессиональных компетенций ПК-4 (способность применять экспериментальные, теоретические и компьютерные методы исследований в профессиональной области); **ПК-6** (способность самостоятельно выполнять экспериментальные или теоретические исследования для решения научных и производственных задач с использованием современной техники и методов расчета и исследования).

Содержание дисциплины охватывает круг вопросов, связанных с разработкой и реализацией алгоритмов решения прикладных задач математики и программирования, и технологией создания программных продуктов.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, практические занятия, самостоятельная работа студента.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости в форме выполнения, защиты лабораторных работ и устного опроса по изученной теме, промежуточный контроль в форме тестирования и итоговый контроль в форме зачёта.

Общая трудоемкость освоения дисциплины составляет 3 зачетных единицы, 108 часов. Программой дисциплины предусмотрены практические (32 часа) занятия, лекции(32 часа) и 44 часа самостоятельной работы студента.

»

Методические указания для самостоятельной работы обучающихся

Тема	Задание
Понятие алгоритма. Основная задача алгоритмизации и порядок составления алгоритмов.	Изучить теоретический материал и оформить конспект по темам: «Алгоритмы. Условное обозначение действия на схемах алгоритма.»
Понятие структурного программирования и схемы алгоритма программы.	Изучить теоретический материал и оформить конспект по темам: Этапы преобразования программы
Классы памяти переменных. Области действия, видимости и продолжительности жизни переменных и функций. Особенности статических и глобальных переменных. Выход переменных из области видимости	Изучить теоретический материал и оформить конспект по темам: Классы памяти переменных.
Области, выделяемые программе при ее выполнении. Область кода. Область данных, глобальных и статических переменных. Стек. Определение физических адресов и размеров областей памяти с помощью отладчика.	Изучить теоретический материал и оформить конспект по темам: . Динамически распределяемая память
Рекурсия.	Изучить теоретический материал и оформить конспект по темам: Рекурсивные определения и алгоритмы
Препроцессор языка Си.	Изучить теоретический материал и оформить конспект по темам: Препроцессор языка Си.
Структуры.	Изучить теоретический материал и оформить конспект по темам: Структуры и записи. Динамические структуры данных
Модульное программирование.	Изучить теоретический материал и оформить конспект по темам: Модульное программирование.
Инициализация графического режима.	Изучить теоретический материал и оформить конспект по темам: Организация графического ввода и вывода текстовой и числовой информации.
Организация графического ввода и вывода текстовой и числовой информации.	Изучить теоретический материал и оформить конспект по темам: Организация графического ввода и вывода текстовой и числовой информации.

Методические указания для студентов по освоению дисциплины

Трудоемкость освоения дисциплины составляет 108 часов, из них 64 часа аудиторных занятий и 44 часа, отведенных на самостоятельную работу студента.

вид учебных занятий	Организация деятельности студента
Контрольная работа/индивидуальные задания	Знакомство с основной и дополнительной литературой, включая справочные издания, зарубежные источники, конспект основных положений, терминов, сведений, требующих для запоминания и являющихся основополагающими в этой теме. Составление аннотаций к прочитанным литературным источникам и др.
Практикум / практические занятия	Методические указания по выполнению практических работ
Подготовка к экзамену (зачету)	При подготовке к экзамену (зачету) необходимо ориентироваться на конспекты лекций, рекомендуемую литературу и др.

ТЕХНОЛОГИИ И ФОРМЫ ПРЕПОДАВАНИЯ

Рекомендации по организации и технологиям обучения для преподавателя

I. Образовательные технологии

Преподавание дисциплины ведется с применением следующих видов образовательных технологий:

Информационные технологии: использование электронных образовательных ресурсов при подготовке к лекциям, практическим и лабораторным занятиям.

Работа в команде: совместная работа студентов в группе при выполнении лабораторных работ, выполнении групповых домашних заданий по разделу.

II. Виды и содержание учебных занятий

Теоретические занятия (лекции) - 32 часа.

РАЗДЕЛ 1. Введение. Основы алгоритмики

Лекция 1. Обзор языков высокого уровня. Место языка Си/Си++ в вычислительной технике. Обзор литературы по языку Си. Понятие алгоритма. Основная задача алгоритмизации и порядок составления алгоритмов. Понятие структурного программирования и схемы алгоритма программы.

Тип: Информационная лекция

Структура: Повторение основных понятий информатики: Системы счисления и переходы между ними. Шестнадцатиричная арифметика. Представление отрицательных чисел. Системы кодирования символов. Способы представления целых и вещественных чисел. Способы записи алгоритма.

РАЗДЕЛ 2. Основы программирования на языке Си

Лекция 2. Исторический обзор. История создания и развития языка Си. Стандарты языка Си.

Тип: Информационная лекция

Структура: Рассмотрены: Основные преимущества Си перед другими языками программирования высокого уровня. Основные особенности работы в Си. Обзор компиляторов языка. Основные понятия теории программирования. Основные принципы написания программ. Структура программы на языке Си. Понятие выражения. Основные этапы решения задач на ЭВМ. Критерии качества программы. Жизненный цикл программы. Постановка задачи и спецификация программы.

Лекция 3. Стандартные типы данных. Понятие операции.

Тип: Информационная лекция

Структура: Имена переменных. Типы и размеры данных. Описания переменных. Символьные и строчные константы, константные выражения.

Пример простейшей программы на Си. Таблица приоритетов операций. Арифметические операции. Основные понятия математической логики. Операции отношения и логические операции.

Преобразование типов. Операции уменьшения и увеличения. Побитовые логические операции. Операции и выражения присваивания. Условные выражения. Старшинство и порядок присваивания.

Лекция 4. Операторы управления.

Тип: Информационная лекция

Структура: Представление основных управляющих структур программирования. Операторы и блоки. Операторы if–else и else–if. Переключатель. Цикл for. Оператор цикла с предусловием while–do. Оператор цикла с постусловием do–while. Операторы безусловного перехода и метки. Пустой оператор. Операторы break и continue.

Лекция 5. Среда программирования Borland C

Тип: Информационная лекция

Структура: Общие сведения. Назначение и состав среды программирования Borland C. Правила работы с файлами. Работа с меню и настройка среды. Компиляция и запуск программ. Основы работы системы отладки программ.

Лекция 6-7. Массивы.

Тип: Информационная лекция

Структура: Определение массива. Одномерные и двумерные массивы. Размещение массива в памяти. Особенности индексации массивов в Си. Многомерные массивы. Утверждения о массивах.

Лекция 8-9. Функции.

Тип: Информационная лекция

Структура: Основные сведения. Аргументы функции: передача по адресу, передача по значению. Синтаксис функций. Описание функций, вызов функций, прототип функций. Внешние переменные. Область действия. Статические и регистровые переменные. Инициализация переменных.

Лекция 10-11. Указатели.

Тип: Информационная лекция

Структура: Указатели и адреса. Указатели и аргументы функций. Указатели и массивы. Адресная арифметика. Многомерные массивы. Массивы указателей. Инициализация массивов указателей. Командная строка аргументов. Указатели на функции.

Лекция 12-13. Модульное программирование/

Тип: Информационная лекция

Структура: Понятия библиотечных и объектных файлов. Правила создания библиотеки пользователя. Понятие оверлея. Правила создания и использования оверлейных блоков.

Лекция 14-16. Графическая библиотека

Тип: Информационная лекция

Структура: Основные библиотечные функции. Инициализация графического режима. Функции установки, рисования, заливки.

Практические занятия - 32 часа.

Занятие №1 Ознакомление со средой Borland C++.

Цель: ознакомление со средой Borland C++.

Занятие №2. Общие сведения о языке C++.

Построение схем ветвящихся и циклических алгоритмов.

Цель: освоить типовые приемы построения схем ветвящихся и циклических алгоритмов.

Занятие №3. Программирование условных операторов и переключателей.

Цель: освоить типовые приемы программирования задач на языке СИ++ с использованием простых типов данных. Составить алгоритм решения задачи. В соответствии с алгоритмом разработать программу на языке C++.

Занятие №4-5. Программирование циклов.

Цель: освоить основы работы с циклами. Составить алгоритм решения задачи. В соответствии с алгоритмом разработать программу на языке C++.

Занятие №6-7. Создание и использование функций

Цель: освоить основы работы с функцией. Составить алгоритм решения задачи. В соответствии с алгоритмом разработать программу на языке C++.

Занятие № 8. Работа с одномерными и двумерными массивами.

Цель: освоить основы работы с массивами. Составить алгоритм решения задачи. В соответствии с алгоритмом разработать программу на языке C++.

Занятие №9-10. Указатели и динамические массивы.

Цель: освоить основы работы с указателями. Составить алгоритм решения задачи. В соответствии с алгоритмом разработать программу на языке C++.

Занятие №11. Обработка матриц в C++.

Цель: освоить основы работы с матрицами. Составить алгоритм решения задачи. В соответствии с алгоритмом разработать программу на языке СИ.

Занятие №12. Организация ввода-вывода в C++.

Цель: освоить основы работы с двоичными и текстовыми файлами. Составить алгоритм решения задачи. В соответствии с алгоритмом разработать программу на языке C++.

Занятие №13-14. Обработка строк.

Цель: освоить основы работы со строками. Составить алгоритм решения задачи. В соответствии с алгоритмом разработать программу на языке C++.

Занятие №15 Структуры.

Цель. освоить основы работы со структурами. Составить алгоритм решения задачи. В соответствии с алгоритмом разработать программу на языке C++.

Занятие №16-17 Объектно-ориентированное программирование.

Цель. освоить основы работы с классами. Составить алгоритм решения задачи. В соответствии с алгоритмом разработать программу на языке C++.

Управление самостоятельной работой студента.
Проверка готовности к практической работе.