

Балаковский инженерно-технологический институт – филиал федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский ядерный университет «МИФИ»

Факультет атомной энергетики и технологий
Кафедра «Информационные системы и технологии»

РАБОЧАЯ ПРОГРАММА

по дисциплине «Технология и языки программирования»

Специальность

«14.05.02 Атомные станции: проектирование, эксплуатация и инжиниринг»

Основная профессиональная образовательная программа

«Системы контроля и управления атомных станций»

Квалификация выпускника

Инженер-физик

Форма обучения

Очная

Цель освоения дисциплины

Цель освоения дисциплины: формирование, развитие и совершенствование компетенций в вопросах разработки программного обеспечения.

Задачи освоения дисциплины: изучение парадигм программирования, технологий разработки программ на языках высокого и низкого уровня, получение навыков программирования в интегрированных средах разработки приложений.

Место дисциплины в структуре ООП ВО

Требуемым условием для освоения дисциплины «Технология и языки программирования» являются знания, умения и практические навыки, полученные при изучении следующих дисциплин: Информатика, Введение в специальность. Знания, умения и практические навыки, полученные при освоении дисциплины «Технология и языки программирования», необходимы при изучении следующих дисциплин и прохождения практик: Автоматизированное проектирование электронных элементов и систем, АСУ технологическими процессами АЭС, Стандартные программные пакеты и средства для моделирования технологических объектов/Информационное обеспечение проектирования техники, Производственная практика (эксплуатационная, по получению профессиональных умений и опыта профессиональной деятельности), Производственная практика (эксплуатационная), Производственная практика (научно-исследовательская работа), Производственная практика (преддипломная), Государственная итоговая аттестация

Компетенции обучающегося, формируемые в результате освоения дисциплины

В процессе освоения данной дисциплины у студента формируются следующие компетенции:

общепрофессиональные

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции
ОПК-3	Способен понимать принципы работы информационных технологий; осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий, соблюдать основные требования информационной безопасности, в том числе защиты государственной тайны	З-ОПК-3 Знать средства и методы поиска, анализа, обработки и хранения информации, в том числе виды источников информации, поисковые системы и системы хранения информации, требования информационной безопасности, включая защиту государственной тайны У-ОПК-3 Уметь осуществлять поиск, хранение, анализ и обработку информации, представлять ее в требуемом формате; применять компьютерные и сетевые технологии, выполнять требования информационной безопасности и защиты государственной тайны В-ОПК-3 Владеть навыком поиска, хранения, обработки и анализа информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий, соблюдать основные требования информационной безопасности, в том числе защиты государственной тайны
ОПК-4	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	З-ОПК-4 Знать основные принципы и требования построения алгоритмов, синтаксис языка программирования У-ОПК-4 Уметь разрабатывать алгоритмы для решения практических задач согласно предъявляемым требованиям В-ОПК-4 Владеть средой программирования и отладки для разработки программ для практического применения

универсальные

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции
УКЦ-2	Способен искать нужные источники информации и данные, воспринимать, анализировать, запоминать и передавать информацию с использованием цифровых средств, а также с помощью алгоритмов при работе с полученными из различных источников данными с целью эффективного использования полученной информации для решения задач	<p>З-1 Знать: методики сбора и обработки информации с использованием цифровых средств, а также актуальные российские и зарубежные источники информации в сфере профессиональной деятельности, принципы, методы и средства решения стандартных задач профессиональной деятельности с использованием цифровых средств и с учетом основных требований информационной безопасности</p> <p>У-1 Уметь: применять методики поиска, сбора и обработки информации; с использованием цифровых средств, осуществлять критический анализ и синтез информации, полученной из разных источников, и решать стандартные задачи профессиональной деятельности с использованием цифровых средств и с учетом основных требований информационной безопасности</p> <p>В-1 Владеть: методами поиска, сбора и обработки, критического анализа и синтеза информации с использованием цифровых средств для решения поставленных задач, навыками подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с использованием цифровых средств и с учетом требований информационной безопасности</p>

Задачи воспитания, реализуемые в рамках освоения дисциплины

Направление/ цели	Создание условий, обеспечивающих	Использование воспитательного потенциала учебных дисциплин	Вовлечение в разноплановую внеучебную деятельность
Профессиональное и трудовое воспитание	- формирование культуры исследовательской и инженерной деятельности (В16)	Использование воспитательного потенциала дисциплин профессионального модуля для формирования инженерного мышления и инженерной культуры за счёт практических студенческих исследований современных производственных систем; проектной деятельности студентов по разработке и оптимизации технологических систем, связанной с решением реальных производственных задач; прохождения через	<p>1. Организация научно-практических конференций и встреч с ведущими специалистами предприятий города и ветеранами атомной отрасли.</p> <p>2. Организация и проведение предметных олимпиад и участие в конкурсах профессионального мастерства.</p> <p>3. Участие в ежегодных акциях студенческих строительных отрядов</p>

		разнообразные игровые, творческие, фестивальные формы, требующие анализа сложного объекта, постановки относительно него преобразовательных задач для их оптимального решения.	
--	--	---	--

Структура и содержание учебной дисциплины

Дисциплина преподается студентам в 3-ем семестре. Общая трудоемкость дисциплины составляет 3 зачетных единицы, 108 ак. часов.

Календарный план

№ Р а з д е л а	№ Т е м ы	Наименование раздела (темы) дисциплины	Виды учебной деятельности (в часах)					Аттеста ция раздела (форма*)	Макси маль ный балл за раздел*
			Всего	Лекции	Лабораторные	Практические	СРС/КРС		
Раздел 1									
1	1	Понятия технологии и языка программирования. Виды языков программирования, жизненный цикл программы.	9/2	2		2/2	5	Тест 1	30
	2	Программирование алгоритмов различной структуры	16/4	2		4/4	10		
	3	Массивы, символьные массивы.	18/4	2		6/4	10		
	4	Указатели.	11/4	2		4/4	5		
Раздел 2									
2	5	Функции	11/2	2		4/2	5	Тест 2	30
	6	Статические структуры и файлы	18/4	2		6/4	10		
	7	Динамические структуры данных	18/4	2		6/4	10		
	8	Директивы препроцессора	7	2			5		
Вид промежуточной аттестации			108/ 24	16		32/24	60	Зачет	40

Содержание лекционного курса

Темы лекции. Вопросы, обрабатываемые на лекции	Всего часов	Учебно- методическое обеспечение
1	2	3
Лекция 1. Понятия технологии и языка программирования. Виды языков программирования, жизненный цикл программы. 1.Этапы разработки программного обеспечения, жизненный цикл ПО. Архитектура современных программных приложений. 2. Современные технологии разработки ПО. Виды технологий	2	2,3,4,5,7

<p>программирования.</p> <p>3. Языки программирования, эволюция языков программирования. Компилируемые и интерпретируемые языки программирования. Стандарты языков программирования.</p> <p>4. Среда разработки программных продуктов. Системы компиляции, этапы сборки исполняемого файла, трансляция программы.</p> <p>5. Основы алгоритмизации, виды и свойства алгоритмов, способы записи алгоритмов.</p> <p>6. Алгоритмические языки, характеристики алгоритмического языка. Алфавит языка программирования, ключевые слова и идентификаторы, структура программы, типы данных, выражения, инструкции, разделители, унарные и бинарные операторы. Организация ввода и вывода данных.</p>		
<p>Лекция 2. <i>Программирование алгоритмов различной структуры</i></p> <p>1. Программирование алгоритмов линейной структуры на языке высокого уровня, форматный ввод и вывод данных.</p> <p>2. Программирование разветвляющихся алгоритмов, условные операторы, полные и неполные условные операторы, вложенные условные операторы, тернарный оператор, оператор выбора.</p> <p>3. Программирование циклических алгоритмов, операторы повтора с предусловием, с постусловием и заданным числом итераций, бесконечные циклы, вложенные циклы.</p>	2	1, 3, 4,5,8
<p>Лекция 3. <i>Массивы, символьные массивы.</i></p> <p>1. Массивы, инициализация массивов, способы заполнения массивов, обход массива, массивы переменной длины, размерность массива, обработка массивов: вычисление суммы элементов массива, поиск элемента в массиве, сортировка массивов.</p> <p>2. Символьные массивы - строки. Создание строк, строка как набор символов, обращение к символу строки. Массивы строк, ввод и вывод строк, преобразование строк к числовому типу. Работа со строками: копирование строк, конкатенация строк, сравнение строк, разбиение строк на лексемы, поиск подстроки в строке.</p>	2	1,3,4,6,5,7,8
<p>Лекция 4. <i>Указатели.</i></p> <p>1. Понятие и назначение указателя, операции получения адреса объекта и разыменованного указателя, операции над указателями., адресация и индексация.</p> <p>2. Алгоритмы обработки числовых массивов с использованием указателей, массивы указателей, указатели на массивы, массивы с разной длиной строк, динамическое изменение размера массива.</p>	2	3,4,5,7
<p>Лекция 5. <i>Функции</i></p> <p>1. Функции, библиотечные функции и функции, созданные пользователем, синтаксическая конструкция пользовательской функции, вызов функции, выход из функции, возврат значения функции.</p> <p>2. Параметры функции, формальные и фактические параметры, функции с неопределенным числом параметров, передача параметров в функцию, указатели на функцию, рекурсия .</p>	2	1,3,4,5,6,7,8
<p>Лекция 6. <i>Статические структуры и файлы</i></p> <p>1. Структуры, инициализация структур, доступ к членам структуры, массивы структур, вложенные структуры, передача структуры в качестве параметра.</p> <p>2. Объединения, перечисления. Примеры использования объединений и перечислений в программах.</p> <p>3. Файлы в программировании, открытие и закрытие потоков, текстовые и бинарные файлы, режимы открытия и закрытия текстовых и бинарных файлов.</p>	2	4,5,7

4. Чтение и з фала и запись в файл, форматируемый ввод-вывод, позиционирование в потоке.		
Лекция 7. <i>Динамические структуры данных.</i> 1. Линейные динамические структуры: односвязные и двусвязные списки, стек, очередь. Добавление и удаление элементов в линейную динамическую структуру. Поиск элемента в линейной динамической структуре. 2. Нелинейные динамические структуры: деревья, бинарные деревья, добавление элементов в дерево, поиск элемента в дереве, удаление элемента из дерева, способы обхода деревьев. Графы, способы представления графов.	2	1,4,5
Лекция 8. <i>Директивы препроцессора.</i> 1. Препроцессор языка С, директивы препроцессора: создание макроопределений, директива включения, директивы условной компиляции, директива определяемая реализацией.	2	4,5

Перечень практических занятий

Тема практического занятия. Вопросы, обрабатываемые на практическом занятии	Всего часов	Учебно-методическое обеспечение
1	2	3
Изучение интегрированной среды разработки ПО 1.Окна и панели инструментов интегрированной среды разработки ПО 2.Технология написания и отладки программного кода 3.Разработка алгоритма вычисления математического выражения. Использование в программе математических операторов и методов. Организация ввода/вывода в программе	2	3,4
Программирование разветвляющихся алгоритмов 1. Разработка алгоритма и программы для решения задачи с использованием условного оператора 2. Использование оператора выбора в программировании	2	1,3,4,5
Программирование циклических алгоритмов 1. Использование операторов повтора с предусловием при программировании циклических алгоритмов 2. Использование операторов повтора с постусловием при программировании циклических алгоритмов 3. Оператор повтора с заданным числом итераций	2	1,3,4,5,8
Программирование с использованием массивов 1. Объявление и инициализация элементов одномерных и двумерных массивов 2. Вычисление суммы и произведения элементов массива 3. Поиск элемента в массиве по заданному условию 4. Инверсия и сортировка элементов массива	4	1,3,4,5,6,7,8
Обработка строк символов 1. Создание строк, строка как символьный массив, обращение к символу строки. 2. Работа со строками: копирование строк, конкатенация строк, сравнение строк, разбиение строк на лексемы, поиск подстроки в строке..	2	1,3,4,5,6,7,8
Работа с указателями 1. Создание одномерных массивов переменной длины с помощью указателей. 2. Создание двумерных массивов переменной длины с помощью указателей.	4	3,4,5,7

Программирование функций 1. Объявление функций в программе, вызов функций. 2. Передача параметров в функцию, передача функции в качестве параметра.	4	1,3,4,5,6,7,8
Программирование с использованием статических структур и бинарных файлов 1. Структуры, инициализация структур, доступ к членам структуры, массивы структур. 2. Создание программ, использующих бинарные файлы. 3. Чтение и запись текстовых файлов в языке программирования C.	6	4,5,7
Программирование линейных динамических структур данных 1. Программирование односвязных списков, добавление элементов в список и удаление элементов из списка. Поиск элемента в списке. 2. Программирование стека и очереди. 3. Программирование двусвязных списков.	4	1,4,5
Программирование нелинейных динамических структур данных 1. Программирование бинарных деревьев, добавление элемента в дерево и удаление элемента из дерева, поиск элемента в бинарном дереве. 2. Программирование графов.	2	1,5
	32	

Задания для самостоятельной работы студентов

Вопросы для самостоятельного изучения (задания)	Всего часов	Учебно-методическое обеспечение
1	2	3
<i>Тема 1. Понятия технологии и языка программирования. Виды языков программирования, жизненный цикл программы.</i> 1. История развития языков и технологий программирования. 2. Области применения языков программирования. 3. Стандарты жизненного цикла программ 4. Программные платформы	5	2,3,4,5,7
<i>Тема 2. Программирование алгоритмов различной структуры</i> 1. Решение задач с использованием сложных условных выражений и оператора выбора 2. Решение задач на использование циклических конструкций. 3. Спецификаторы форматного ввода/вывода	10	1,3,4,5,8
<i>Тема 3. Массивы, символьные массивы.</i> 1. Виды сортировки массивов. 2. Функции поиска элементов в строке.	10	1,3,4,5,6,7,8
<i>Тема 4. Указатели.</i> 1. Арифметика указателей	5	3,4,5,7
<i>Тема 5. Функции</i> 1. Рекурсии, рекурсивные алгоритмы.	5	1,3,4,5,6,7,8
<i>Тема 6. Статические структуры и файлы</i> 1. Использование структур для работы с битовыми полями 2. Форматированный ввод и вывод в строки.	10	4,5,7
<i>Тема 7 Динамические структуры данных</i> 1. Распределение рабочего пространства оперативной памяти. 2. Построение дерева разбора арифметического выражения. 3. Динамический список смежности (хэш-таблица)	10	1,4,5
<i>Тема 8. Директивы препроцессора</i> 1. Препроцессорные операции	5	4,5
	60	

Образовательные технологии

При изучении дисциплины «Технологии и языки программирования» используются интерактивные формы проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

При реализации учебного материала курса используются следующие образовательные технологии: использование электронных образовательных ресурсов, обеспечивающих обучение в информационной образовательной среде; лекционные занятия проводятся с использованием мультимедийного лекционного материала, практические занятия проводятся с применением ПК в интегрированной среде разработки программного обеспечения. Самостоятельная работа студентов проводится под руководством преподавателей, с оказанием консультаций и помощи при подготовке к контрольным работам, выполнении домашних заданий.

Фонд оценочных средств

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий текущего, рубежного и промежуточного контроля по дисциплине.

Связь между формируемыми компетенциями и формами контроля их освоения представлена в следующей таблице:

№ п/п	Наименование контролируемых разделов (темы)	Код и наименование индикатора достижения компетенций	Наименование оценочного средства
Входной контроль			
1	Входной контроль		Вопросы входного контроля (письменно)
Аттестация разделов, текущий контроль успеваемости			
2	Раздел 1	3-УКЦ-2, У-УКЦ-2, В-УКЦ-2, 3-ОПК-3, У-ОПК-3, ОПК-4	Тестирование (письменно) Практические задания (письменно)
3	Раздел 2	3-УКЦ-2, У-УКЦ-2, 3-ОПК-3, У-ОПК-3, В-ОПК-3, ОПК-4	Тестирование (письменно) Практические задания (письменно)
Промежуточная аттестация			
4	Зачет	3-УКЦ-2, У-УКЦ-2, В-УКЦ-2, 3-ОПК-3, У-ОПК-3, В-ОПК-3, ОПК-4	Вопросы к зачету (устно)

Входной контроль предназначен для выявления пробелов в знаниях студентов и готовности их к получению новых знаний. Оценочные средства для входного контроля представляют собой вопросы и практические задания, которые задаются студентам в письменной форме.

В качестве оценочного средства текущего контроля используются выполнение индивидуальных заданий на практических занятиях и устный опрос по результатам их выполнения.

В качестве оценочного средства аттестации разделов используются контрольные тесты, а для промежуточной аттестации предусмотрены теоретические вопросы.

По итогам обучения выставляется зачет.

Вопросы и задания входного контроля

1. В программе «:=» обозначает оператор присваивания, знаки «+», «-», «*» и «/» — соответственно операции сложения, вычитания, умножения и деления. Правила выполнения операций и порядок действий соответствуют правилам арифметики. Определите значение переменной *b* после выполнения алгоритма:

a := 2

b := 4

a := 2*a + 3*b

b := a/2*b

В ответе укажите одно целое число — значение переменной b.

2. Найдите значение выражения: $((0 \& 0) \vee 0) \& (1 \vee 1)$;
3. Зашифрованы русские слова (последовательности букв), вместо каждой буквы записан её код:

А	Д	К	Н	О	С
01	100	101	10	111	000

Расшифруйте закодированное слово 10111000:

4. Для какого из приведённых значений числа X ложно высказывание:
НЕ (X < 6) ИЛИ (X < 5)?
- a) 7
b) 6
c) 5
d) 4
5. Переведите двоичное число 1101101 в десятичную систему счисления.
6. Переведите число 135 из десятичной системы счисления в двоичную систему счисления. Сколько единиц содержит полученное число? В ответе укажите одно число — количество единиц.
7. Доступ к файлу **example.doc**, находящемуся на сервере **task.org**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети Интернет.
а) .txt б) :// в) http г) task е) / ж) .org з) example
8. Файл размером 64 Кбайт передаётся через некоторое соединение со скоростью 1024 бит в секунду. Определите размер файла (в Кбайт), который можно передать за то же время через другое соединение со скоростью 256 бит в секунду. В ответе укажите размер файла в Кбайт.
9. Статья, набранная на компьютере, содержит 48 страниц, на каждой странице 40 строк, в каждой строке 64 символа. Определите размер статьи в кодировке КОИ-8, в которой каждый символ кодируется 8 битами
10. Для хранения растрового изображения размером 32 x 32 пикселя отвели 512 байт памяти. Каково максимально возможное число цветов в палитре изображения?

Примерный тест (по вариантам) для аттестации раздела 1.

1. Какое из ключевых слов используется для описания вещественного типа?
- a) double
b) int
c) char
d) bool
2. В записи инициализации двумерного массива: `int x[2][3] = {{0,4,5},{2,2,8}}` число 5 будет находится по адресу:
- a) [0,0]
b) [1,2]
c) [0,2]
d) [2,3]
3. Чему будет равна переменная s в результате выполнения следующего программного кода?

```
7 int _tmain(int argc, _TCHAR* argv[])
8 {
9     int s = 0;
10    for(int i = 1; i <= 3; i++)
11    {
12        s = s + 2 * i;
13    }
14    printf("%d", s);
15    return 0;
16 }
```

- a) 2
b) 6
c) 12

- d) 18
4. С помощью какого оператора можно сравнить два числа в программе?
- =
 - ==
 - :=
 - ++
5. Что означает запись !=
- присвоить
 - равно
 - Неравно
 - Отрицание
6. Чему будет равно значение переменной u после выполнения последовательности команд:
int x=7; int u=x--;
- 5
 - 6
 - 7
 - 0
7. Какие служебные символы используются для обозначения начала и конца блока кода?
- <>
 - { }
 - ()
 - //

8. С помощью следующего программного кода:

```

7 int _tmain(int argc, _TCHAR* argv[])
8 {
9     int mas[3][3] = {{2, 7, 3}, {45, 6, 12}, {43, 23, 5}};
10    for(int i = 0; i < 3; i++)
11    {
12        for(int k = 0; k < 3; k++)
13        {
14            if(i==k)
15            {
16                printf("%d ",mas[i][k]);
17            }
18        }
19    }
20    return 0;
21 }
```

- выполняется инверсия одномерного массива
- выводятся на экран элементы главной диагонали матрицы
- выполняется сортировка каждой строки матрицы
- выводится на экран индексы матрицы

Какой из идентификаторов записан неверно?

- zzz
- polnaja_summa
- A&X
- as55

9. В результате выполнения программы в окно консоли будет выведена строка:

```

8  int _tmain(int argc, _TCHAR* argv[])
9  {
10     char s1[] = "1 2 3 4 5";
11     for(int j = 0; j <= strlen(s1)-1; j++)
12     {
13         if(s1[j] == ' ')
14         {
15             *(s1 + j) = '+';
16         }
17     }
18     printf("%s", s1);
19     return 0;
20 }

```

- a) 12345
- b) 1+2+3+4+5
- c) 1+2+3+4+5+
- d) +1+2+3+4+

Студент на контрольном тестировании выполняет задания письменно. За каждое правильное решение начисляется баллы. Максимально за контрольный тест - 10 баллов.

Баллы для аттестации раздела включают в себя баллы текущего контроля и контрольного теста.

Шкала оценивания раздела 1

Текущий контроль успеваемости	Аттестация раздела	Максимальный / минимальный балл для аттестации раздела
Задание 1 - 2 балла Задание 2 – 4 балла Задание 3 – 4 балла Задание 4 – 4 балла Задание 5 – 3 балла Задание 6 – 3 балла	Контрольный тест 1 - 10 б.	30 баллов / 18 баллов

Примерный тест (по вариантам) для аттестации раздела 2.

1. Формальные параметры - это:
 - a) собственно параметры функции, объявленные в заголовке функции
 - b) значения, которые передаются функции
 - c) аргументы функции
 - d) переменные, объявленные в функции main
2. В результате выполнения программного кода переменная a будет равна:

```

8  int _tmain(int argc, _TCHAR* argv[])
9  {
10     int a = 5;
11     printf("Начальное значение a=%d\n", a);
12     Inc(&a);
13     printf("После передачи по ссылке a=%d\n", a);
14     return 0;
15 }
16 void Inc(int* x)
17 {
18     *x = *x + 1;
19     printf("Increment: %d\n", *x);
20 }

```

- a) 5
 - b) 6
 - c) 2
 - d) 0
3. Динамическую структуру, работающую по алгоритму FIFO (first in, first out - "первый вошел - первый вышел") называют:
 - a) двусвязанным списком

- b) стеком
 - c) хэш-таблицей
4. Очередью Сколько указателей используется в односвязных списках?
- a) 1
 - b) 2
 - c) 3
 - d) сколько угодно
5. Линейный список, в котором доступен только последний элемент, называется
- a) стеком
 - b) очередью
 - c) массивом
 - d) деревом
6. Элемент дерева, на который нет ссылок, называется:
- a) корнем
 - b) промежуточным элементом
 - c) листом
7. Для работы с файлами и устройствами ввода-вывода необходимо подключить заголовочный файл:
- a) `stdlib.h`
 - b) `stdio.h`
 - c) `locale.h`
 - d) `math.`
8. Какой режим работы с файлом определяется спецификатором "rb"
- a) бинарный файл открывается для записи
 - b) бинарный файл открывается для чтения
 - c) текстовый файл открывается для записи. Если файл ранее существовал, то он пересоздается и записывается заново
 - d) текстовый файл создается для записи. Если файл ранее существовал, то при первой записи после открытия он пересоздается и записывается заново. А при последующих записях после открытия данные добавляются в него без перезаписи.
9. Слово `void` при описании функции указывает на то, что:
- a) функция объявлена как статическая
 - b) функция ничего не возвращает
 - c) функция возвращает числовое значение
 - d) функция обязана вернуть значение типа `void`
10. При работе с переменной типа указатель на структуру используется оператор?
- a) `.`
 - b) `->`
 - c) `::`
 - d) `!`

Студент на контрольном тестировании выполняет задания письменно. За каждое правильное решение начисляется баллы. Максимально за контрольный тест - 10 баллов.

Баллы для аттестации раздела включают в себя баллы текущего контроля и контрольного теста.

Шкала оценивания раздела 2

Текущий контроль успеваемости	Аттестация раздела	Максимальный / минимальный балл для аттестации раздела
Задание 1 - 4 балл Задание 2 – 8 баллов Задание 3 – 8 баллов	Контрольный тест 2 - 10 б.	30 баллов / 18 баллов

Вопросы для подготовки к зачёту

1. Этапы разработки программного обеспечения, жизненный цикл ПО. Архитектура современных программных приложений.

2. Языки программирования, эволюция языков программирования. Компилируемые и интерпретируемые языки программирования. Стандарты языков программирования.

3. Среды разработки программных продуктов. Системы компиляции, этапы сборки исполняемого файла, трансляция программы.
 4. Основы алгоритмизации, виды и свойства алгоритмов, способы записи алгоритмов.
 5. Алгоритмические языки, характеристики алгоритмического языка. Алфавит языка программирования, ключевые слова и идентификаторы, структура программы.
 6. Типы данных, выражения, инструкции, разделители, унарные и бинарные операторы.
 7. Организация ввода и вывода данных, форматный ввод и вывод данных.
 8. Программирование разветвляющихся алгоритмов, условные операторы, полные и неполные условные операторы, вложенные условные операторы, тернарный оператор, оператор выбора.
 9. Программирование циклических алгоритмов, операторы повтора с предусловием, с постусловием и заданным числом итераций, бесконечные циклы, вложенные циклы.
 10. Массивы, инициализация массивов, способы заполнения массивов, обход массива, массивы переменной длины, размерность массива, обработка массивов: вычисление суммы элементов массива, поиск элемента в массиве, сортировка массивов.
 11. Символьные массивы - строки. Создание строк, строка как набор символов, обращение к символу строки. Массивы строк, ввод и вывод строк, преобразование строк к числовому типу. Работа со строками: копирование строк, конкатенация строк, сравнение строк, разбиение строк на лексемы, поиск подстроки в строке.
 12. Понятие и назначение указателя, операции получения адреса объекта и разыменования указателя, операции над указателями., адресация и индексация.
 13. Алгоритмы обработки числовых массивов с использованием указателей, массивы указателей, указатели на массивы, массивы с разной длиной строк, динамическое изменение размера массива.
 14. Функции, библиотечные функции и функции, созданные пользователем, синтаксическое построение пользовательской функции, вызов функции, выход из функции, возврат значения функции.
 15. Параметры функции, формальные и фактические параметры, функции с неопределенным числом параметров, передача параметров в функцию, указатели на функцию, рекурсия.
 16. Структуры, инициализация структур, доступ к членам структуры, массивы структур, вложенные структуры, передача структуры в качестве параметра.
 17. Объединения, перечисления. Примеры использования объединений и перечислений в программах.
 18. Файлы в программировании, открытие и закрытие потоков, текстовые и бинарные файлы, режимы открытия и закрытия текстовых и бинарных файлов.
 19. Чтение из файла и запись в файл, форматируемый ввод-вывод, позиционирование в потоке.
 20. Линейные динамические структуры: односвязные и двусвязные списки, стек, очередь. Добавление и удаление элементов в линейную динамическую структуру. Поиск элемента в линейной динамической структуре.
 21. Нелинейные динамические структуры: деревья, бинарные деревья, добавление элементов в дерево, поиск элемента в дереве, удаление элемента из дерева, способы обхода деревьев.
 22. Графы, способы представления графов.
 23. Препроцессор языка C, директивы препроцессора: создание макроопределений, директива включения, директивы условной компиляции, директива определяемая реализацией.
- Зачет проводится в устной форме, путем ответа на 2 вопроса из вышеприведенного перечня вопросов. При этом оценивается правильность и полнота ответа. Максимальный балл за зачет – 40 баллов (20 баллов за каждый вопрос).

Шкалы оценки образовательных достижений

Баллы (итоговой рейтинговой оценки)	Оценка (балл за ответ на зачете)	Требования к знаниям

60-100	«зачтено» 24 - 40 баллов	<p>– Оценка «зачтено» выставляется студенту, если он имеет знания основного материала, если он прочно усвоил программный материал, последовательно, четко и логически стройно его излагает его на зачете, владеет необходимыми навыками и приемами их выполнения, умеет тесно увязывать теорию с практикой</p> <p>– Учебные достижения в семестровый период и результатами рубежного контроля демонстрируют достаточную степень овладения программным материалом.</p>
0-59	«не зачтено» 0 -23 баллов	<p>– Оценка «не зачтено» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки. Как правило, оценка «не зачтено» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.</p> <p>– Учебные достижения в семестровый период и результатами рубежного контроля демонстрировали не высокую степень овладения программным материалом по минимальной планке.</p>

Учебно-методическое и информационное обеспечение учебной дисциплины

Основная литература:

1. Андрианова, А. А. Алгоритмизация и программирование. Практикум : учебное пособие / А. А. Андрианова, Л. Н. Исмагилов, Т. М. Мухтарова. — Санкт-Петербург : Лань, 2019. — 240 с. — ISBN 978-5-8114-3336-0. — Текст: электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/113933>
2. Калентьев, А. А. Новые технологии в программировании : учебное пособие / А. А. Калентьев. — Москва : ТУСУР, 2014. — 176 с. <https://e.lanbook.com/reader/book/110361/#1>
3. Орлов С. Теория и практика языков программирования. Учебник для вузов. Стандарт 3-го поколения / С. Орлов. - Санкт-Петербург : Питер, 2014. - 688 с. - URL: <https://ibooks.ru/bookshelf/26402/reading>
4. Подбельский, В. В. Курс программирования на языке Си : учебник / В. В. Подбельский, С. С. Фомин. — Москва : ДМК Пресс, 2012. — 384 с. — URL: <https://e.lanbook.com/book/4148>
5. Рацеев, С. М. Программирование на языке Си : учебное пособие для вузов / С. М. Рацеев. — Санкт-Петербург : Лань, 2022. — 332 с.— URL: <https://e.lanbook.com/book/193320>

Дополнительная литература:

6. Гуркова, М. А. Программирование на языке Си: Практикум : учебное пособие / М. А. Гуркова, Э. Р. Резникова. — Москва : РУТ (МИИТ), 2020. — 70 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/175947>
7. Тракимус, Ю. В. Основы программирования : учебное пособие / Ю. В. Тракимус, В. П. Хиценко. — Новосибирск : НГТУ, 2020. — 66 с. — ISBN 978-5-7782-4089-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/152224>
8. Программирование. Сборник задач: учебное пособие / О. Г. Архипов, В. С. Батасова, П. В. Гречкина [и др.] ; под редакцией М. М. Марана. — Санкт-Петербург : Лань, 2019. — 140 с. — ISBN 978-5-8114-3857-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/121485>

Программное обеспечение и Интернет-ресурсы:

Процесс реализации образовательной программы обеспечен необходимым комплектом лицензионного программного обеспечения..

Ресурсы информационно-телекоммуникационной сети «Интернет»:

1. Электронно-библиотечная система «Лань» - <http://e.lanbook.com>.
2. Интернет-Университет Информационных Технологий - <http://www.intuit.ru>

Материально-техническое обеспечение учебной дисциплины

Лекционные занятия по дисциплине проводятся в аудитории с мультимедийным оборудованием. Для проведения практических занятий и выполнения самостоятельной работы используются учебные компьютерные классы с выходом в Интернет и лицензионным программным обеспечением.

Учебно-методические рекомендации для студентов

1. Указания для прослушивания лекций

Перед началом занятий внимательно ознакомьтесь с учебным планом проведения лекций и списком рекомендованной литературы.

Перед посещением очередной лекции освежить в памяти основные концепции пройденного ранее материала. Подготовить при необходимости вопросы преподавателю. Не надо опасаться, что вопросы могут быть простыми.

На лекции основное внимание следует уделять не формулам и математическим выкладкам, а содержанию изучаемых вопросов, определениям и постановкам задач.

В процессе изучения лекционного курса необходимо по возможности часто возвращаться к основным понятиям и методам решения задач (здесь возможен выборочный контроль знаний студентов).

Желательно использовать конспекты лекций, в которых используется принятая преподавателем система обозначений.

Для более подробного изучения курса следует работать с рекомендованными литературными источниками и вновь появляющимися источниками.

2. Указания для участия в практических занятиях

Перед посещением уяснить тему практического занятия и самостоятельно изучить теоретические вопросы.

В конце занятия при необходимости выяснить у преподавателя неясные вопросы.

Основные результаты выполнения работы необходимо распечатать.

3. Самостоятельная работа студентов обычно складывается из нескольких составляющих:

- работа с текстами: учебниками, историческими первоисточниками, дополнительной литературой, в том числе материалами интернета, а также проработка конспектов лекций;
- написание докладов, рефератов;
- подготовка к практическим занятиям;
- подготовка к зачету непосредственно перед ним.

Таким образом, самостоятельная работа студентов является необходимым компонентом получения полноценного высшего образования.

Методические рекомендации для преподавателей

1. Указания для проведения лекций

На первой вводной лекции сделать общий обзор содержания курса и отметить новые методы и подходы к решению задач, рассматриваемых в курсе, довести до студентов требования кафедры, ответить на вопросы.

При подготовке к лекционным занятиям необходимо продумать план его проведения, содержание вступительной, основной и заключительной части лекции, ознакомиться с новинками учебной и методической литературы, публикациями периодической печати по теме лекционного занятия. Уточнить план проведения практического занятия по теме лекции. Перед изложением текущего лекционного материала напомнить об основных итогах, достигнутых на предыдущих лекциях. С этой целью задать несколько вопросов аудитории и осуществить выборочный контроль знания студентов.

В ходе лекционного занятия преподаватель должен назвать тему, учебные вопросы, ознакомить студентов с перечнем основной и дополнительной литературы по теме занятия. Раскрывая содержание учебных вопросов, акцентировать внимание студентов на основных категориях, явлениях и процессах, особенностях их протекания. Раскрывать сущность и содержание различных точек зрения и научных подходов к объяснению тех или иных явлений и

процессов.

Следует аргументировано обосновать собственную позицию по спорным теоретическим вопросам. Приводить примеры. Задавать по ходу изложения лекционного материала риторические вопросы и самому давать на них ответ. Это способствует активизации мыслительной деятельности студентов, повышению их внимания и интереса к материалу лекции, ее содержанию. Преподаватель должен руководить работой студентов по конспектированию лекционного материала, подчеркивать необходимость отражения в конспектах основных положений изучаемой темы, особо выделяя, категориальный аппарат. В заключительной части лекции необходимо сформулировать общие выводы по теме, раскрывающие содержание всех вопросов, поставленных в лекции. Объявить план очередного практического занятия, дать краткие рекомендации по подготовке студентов к практическому занятию. Определить место и время консультации студентам, пожелавшим выступить с докладами и рефератами.

На последней лекции уделить время для обзора наиболее важных положений, рассмотренных в курсе.

2. Указания для проведения практических занятий

Четко обозначить тему практической работы.

Обсудить основные понятия, связанные с темой работы.

В процессе решения задач вести дискуссию со студентами о правильности применения теоретических знаний.

Отмечать студентов, наиболее активно участвующих в решении задач и дискуссиях.

В конце занятия задать аудитории несколько контрольных вопросов, проконтролировать ход выполнения практической работы.

3. Указания по контролю самостоятельной работы студентов

По усмотрению преподавателя задание на самостоятельную работу может быть индивидуальным или фронтальным.

При использовании индивидуальных заданий требовать от студента письменный отчет о проделанной работе.

При применении фронтальных заданий вести коллективные обсуждения со студентами основных теоретических положений.

С целью контроля качества выполнения самостоятельной работы требовать индивидуальные отчеты (допустимо вместо письменного отчета применять индивидуальные контрольные вопросы).

Программа составлена в соответствии с требованиями ОС НИЯУ МИФИ и учебным планом основной образовательной программы по специальности 14.05.02 Атомные станции: проектирование, эксплуатация и инжиниринг.

Рабочую программу составил Несытых И.В.

Рецензент: доцент Мефедова Ю.А.

Программа одобрена на заседании УМКС 14.05.02 Атомные станции: проектирование, эксплуатация и инжиниринг.

Председатель учебно-методической комиссии Ефремова Т. А.