|  |  |  |  |
| --- | --- | --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  РОССИЙСКОЙ ФЕДЕРАЦИИ  ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  «Национальный исследовательский ядерный университет «МИФИ»  Озерский технологический институт -  филиал федерального государственного автономного образовательного учреждения  высшего образования «Национальный исследовательский ядерный университет «МИФИ»  (ОТИ НИЯУ МИФИ) | | | |
| Кафедра прикладной математики | | |
| УТВЕРЖДАЮ  ДИРЕКТОР  И. А. Иванов  « 24 » мая 2021 г.. | | |
| РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ | | |
| Программирование | | |
| (наименование дисциплины) | | |
|  | | |
| Направление подготовки (специальность): | 09.03.01 Информатика и вычислительная техника | |
|  |  | |
| Профиль подготовки: | Программное обеспечение средств вычислительной техники и автоматизированных систем | |
|  |  | |
| Наименование образовательной программы: | Программное обеспечение средств вычислительной техники и автоматизированных систем | |
|  |  | |
| Квалификация (степень) выпускника: | бакалавр |  |
| (бакалавр, магистр, специалист) |  |
|  |  | |
| Форма обучения: | очная |  |
| (очная, очно-заочная (вечерняя), заочная) |  |

г. Озёрск, 2021 г.

# ЦЕЛИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Целью освоения учебной дисциплины «Программирование» является формирование умений и навыков в области алгоритмизации, практических навыков в области программирования, отладки и тестирования программной части аппаратно-программных комплексов на языках программирования высокого уровня.

# МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Дисциплина «Программирование» входит в Блок 1 «Дисциплины (модули)» в обязательную часть основной образовательной программы бакалавриата «Программное обеспечение средств вычислительной техники и автоматизированных систем», модуль «Общепрофессиональный».

# КОМПЕТЕНЦИИ СТУДЕНТА, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ / ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ ОБРАЗОВАНИЯ И КОМПЕТЕНЦИИ СТУДЕНТА ПО ЗАВЕРШЕНИИ ОСВОЕНИЯ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

В результате освоения данной дисциплины у студента формируются следующие компетенции и планируются следующие результаты обучения по дисциплине:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Компетенция / Индикатор** | **Содержание** | **Результаты обучения по дисциплине** |
|  | **ОПК-1** | **Способен использовать положения, законы и методы естественных наук и математики для решения задач инженерной деятельности** |  |
|  | ОПК-1.1 | З-ОПК-1 Знать: основы математики, физики, общеинженерных знаний, вычислительной техники и программирования |  |
|  | ОПК-1.2 | У-ОПК-1 уметь: решать нестандартные профессиональные задачи с применением естественнонаучных, математических и общеинженерных знаний |  |
|  | ОПК-1.3 | В-ОПК-1 владеть: навыками решения нестандартных задач профессиональной деятельности, в том числе в междисциплинарном контексте |  |
|  | **ОПК-2** | **Способен самостоятельно проводить экспериментальные исследования и использовать основные приемы обработки и представления полученных данных** |  |
|  | ОПК-2.1 | З-ОПК-2 Знать: современные методы и средства обработки и представления данных экспериментальных исследований |  |
|  | ОПК-2.2 | У-ОПК-2 уметь: выбирать современные методы и средства обработки и представления данных экспериментальных исследований, составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули |  |
|  | ОПК-2.3 | В-ОПК-2 владеть: навыками применения методов и средств обработки и представления данных экспериментальных исследований, языками программирования, навыками отладки и тестирования работоспособности программ, применяемых для решения профессиональных задач |  |
|  | **ОПК-3** | **Способен применять методы поиска, хранения, обработки, анализа и представления в требуемом формате информации из различных источников и баз данных, соблюдая при этом основные требования информационной безопасности** |  |
|  | ОПК-3.1 | З-ОПК-3 Знать: методы поиска, хранения, обработки, анализа и представления в требуемом формате профессиональной информации с применением информационнокоммуникационных технологий и с учетом основных требований информационной безопасности |  |
|  | ОПК-3.2 | У-ОПК-3 Уметь: анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями с применением информационнокоммуникационных технологий и с учетом основных требований информационной безопасности |  |
|  | ОПК-3.3 | В-ОПК-3 Владеть: навыками поиска, хранения, обработки, анализа и представления в требуемом формате информации из различных источников и баз данных, соблюдая при этом основные требования информационной безопасности |  |
|  | **ОПК-8** | **Способен разрабатывать алгоритмы и программы, пригодные для практического применения** |  |
|  | ОПК-8.1 | З-ОПК-8 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения |  |
|  | ОПК-8.2 | У-ОПК-8 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули |  |
|  | ОПК-8.3 | В-ОПК-8 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы |  |

В результате освоения дисциплины студент должен знать:

- технологию разработки алгоритмов и программ, методы отладки и решения задач на ЭВМ.

В результате освоения дисциплины студент должен уметь:

- ставить задачу и разрабатывать алгоритм её решения, использовать прикладные системы программирования, разрабатывать основные программные документы.

В результате освоения дисциплины студент должен владеть:

- языком процедурного структурного программирования Си, навыками разработки и отладки программ на алгоритмическом процедурном структурном языке программирования высокого уровня Си.

# СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Общая трудоемкость дисциплины составляет: |  |  |  |  |
| Кредитов: | 6 |  |  |  |
| Часов: | 216 |  |  |  |

в том числе: контактных рабочих 116, самостоятельная работа 64, контроль 36

По семестрам

1 семестр: контактная работа 64 (лекции 32, практики 32), самостоятельная работа 44, зачет

2 семестр: контактная работа 52 (лекции 26, практики 26), самостоятельная работа 20, контроль 36 (экзамен)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № п/п | Раздел учебной дисциплины | Недели | Виды учебной деятельности, включая СРС, трудоемкость (в часах) | | | | Текущий контроль успеваемости (*неделя, форма*) | Аттестация раздела  (*неделя, форма*) | Макс. балл за раздел |
| Лекции | Практ. занятия/ семинары | Лаб. раб. | СРС |
|  | 1 семестр | | | | | | | | |
| 1 | Алгоритмизация и программирование | 1-3 | 6 | 6 |  | 8 | 1ПР1  2ПР2  3ПР3  4ПР4 | 6КР1 | 10 |
| 2 | Формализация понятия алгоритма | 4-6 | 6 | 6 |  | 8 | 5ПР5  6ПР6  7ПР7 | 8ДЗ1 | 8 |
| 3 | Типы данных | 7-9 | 6 | 6 |  | 8 | 8ПР8  9ПР9  10ПР10 | 18КР2 | 8 |
| 4 | Язык Си | 10-13 | 6 | 6 |  | 8 | 11ПР11  12ПР12  13ПР13  14ПР14 | 18КР2 | 12 |
| 5 | Структурное программирование на языке Си | 14-17 | 8 | 8 |  | 12 | 15ПР15  16ПР16  17ПР1718ПР18 | 18КР2  17ДЗ2 | 12 |
|  | Итого баллов за семестр: |  | 36 | 36 |  | 36 |  |  | 50 |
|  | Зачет: |  |  |  |  |  |  |  | 50 |
|  | Итого за 2 семестр: |  |  |  |  |  |  |  | 100 |
| 2 семестр | | | | | | | | | |
| 6 | Пользовательские типы данных в языке Си | 1-4 | 6 | 6 |  | 5 | 1ПР1  2ПР2  3ПР3  4ПР4 | 4КР1  8ДЗ1 | 12 |
| 7 | Основные возможности языка Си++, расширяющие язык Си | 5-8 | 6 | 6 |  | 5 | 5ПР5  6ПР6  7ПР7  8ПР8 | 8КР2  12ДЗ2 | 14 |
| 8 | Языки программирования и трансляция | 9-13 | 8 | 8 |  | 5 | 9ПР9  10ПР10  11ПР11  12ПР12  13ПР13 | 16КР3 | 14 |
| 9 | Методологии программирования | 14-16 | 6 | 6 |  | 5 | 14ПР14  15ПР15  16ПР16 | 16КР3  16ДЗ3 | 10 |
|  | Итого баллов за семестр: |  | 32 | 32 |  | 44 |  |  | 50 |
|  | Зачет: |  |  |  |  |  |  |  | 50 |
|  | Итого за 2 семестр: |  |  |  |  |  |  |  | 100 |

Обозначения оценочных средств: ПР - практическая работа, КР - контрольная работа, ДЗ - индивидуальное домашнее задание.

В разделе 1 «Алгоритмизация и программирование» рассматриваются следующие вопросы: алгоритмы, исполнители, свойства алгоритмов; формы записи алгоритма; базовые алгоритмические структуры, запись алгоритма в виде блок-схемы, ГОСТ 19.003-80, ГОСТ 19.701-90; уровни языка программирования, языки программирования высокого уровня, достоинства и недостатки машинных языков, преимущества алгоритмических языков перед машинными; компоненты, образующие язык программирования, основные понятия в алгоритмических языках: данные, константы, переменные, массивы, выражения, операторы.

В разделе 2 «Формализация понятия алгоритма» рассматриваются следующие вопросы: математическое уточнение понятия алгоритма, вычислимость функции, разрешимость множества, перечислимость множества, заключение о несводимости понятия алгоритма к понятию функции; машина Тьюринга, тезис Тьюринга, тезис Чёрча-Тьюринга; нормальные алгорифмы Маркова; машина Поста.

В разделе 3 «Типы данных» рассматриваются следующие вопросы: переменные, целочисленные переменные, вещественные переменные, машинный эпсилон, вещественные константы, символьные переменные, логические переменные и выражения, логические операторы, массивы данных, строки.

В разделе 4 «Язык Си» рассматриваются следующие вопросы: язык Си, происхождение языка Си; исходный файл на языке Си, исполняемый файл; элементы языка Си: алфавит, идентификаторы, ключевые слова, комментарии, константы, операции и выражения, операторы, описания; программа на языке Си, директивы препроцессора, главная функция, блок программы, общая структура программы; типы данных языка Си, описание данных в программе, объявление переменных и констант, определение размера данных, целые константы, константы с плавающей точкой, символьные константы, esc-последовательности, строковые константы; арифметические выражения, оператор присваивания, одноместные операции, сокращенная запись оператора присваивания; автоматическое преобразование типов, операция приведения типа, инициализация переменных, объявление констант; форматированный ввод, форматированный вывод.

В разделе 5 «Структурное программирование на языке Си» рассматриваются следующие вопросы: составной оператор, условный оператор, условная операция, оператор выбора, оператор break в операторе выбора; цикл с параметром, цикл с предусловием, цикл с постусловием, операторы break и continue в циклах; массивы, символьные массивы, указатели, операция взятия адреса, операция косвенной адресации, выражения и арифметические операции с указателями, связь между указателями и массивами, указатели при обработке символьных строк; многомерные массивы, подпрограммы в языке Си - функции, формальные и фактические параметры, возвращение значения, передача параметров в функцию по значению и через указатель, прототипы функций; область видимости и время жизни идентификаторов; рекурсивные функции; указатели на функции.

В разделе 6 «Пользовательские типы данных в языке Си» рассматриваются следующие вопросы: классы памяти, автоматический период хранения, статический период хранения, модификатор const, передача функции указателя с модификатором const, аргументы командной строки; структуры, операции со структурами, обращение к элементам структуры, использование структур с функциями; объединения, операции с объединениями; поразрядные операции, битовые поля, перечислимые константы.

В разделе 7 «Основные возможности языка Си++, расширяющие язык Си» рассматриваются следующие вопросы: потоковый ввод и вывод, пространства имён, встроенные функции, параметры-ссылки, аргументы по умолчанию, перегрузка функций, перегрузка операций, шаблоны функций.

В разделе 8 «Языки программирования и трансляция» рассматриваются следующие вопросы: понятия транслятора, компилятора, интерпретатора, эмулятора, макропроцессора; синтаксис и семантика; общие особенности языков программирования и трансляторов; обобщенная структура транслятора, варианты взаимодействия блоков транслятора: многопроходная организация взаимодействия блоков транслятора, однопроходная организация взаимодействия блоков транслятора, организация взаимодействия блоков интерпретатора.

В разделе 9 «Методологии программирования» рассматриваются следующие вопросы: нисходящее программирование, модульное программирование; методологии программирования, основные методологии, ядра методологий, методология императивного программирования, методология объектно-ориентированного программирования, методология функционального программирования. методология логического программирования, топологическая специфика методологий, реализационная специфика методологий.

# ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Образовательный процесс по дисциплине строится на основе комбинации образовательных технологий. При освоении разделов дисциплины используется сочетание видов учебной деятельности (лекции, практические занятия, самостоятельная работа студента) с использованием интерактивных форм проведения занятий в аудитории.

Используемые образовательные технологии при изучении данной дисциплины:

- контекстное обучение;

- метод проектов;

- работа в команде;

- дискуссия;

- тренинг;

Интерактивные формы проведения занятий составляют 28 часов или 20% от общего объема аудиторных занятий.

# ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ.

## Текущий контроль проводится в виде контроля выполнения практических работ.

Примерные темы практических работ в 1 семестре

1) Запись линейных алгоритмов в виде блок-схемы.

2) Запись разветвляющихся алгоритмов в виде блок-схемы (2 занятия).

3) Запись циклических алгоритмов в виде блок-схемы (2 занятия).

4) Запись алгоритмов с массивами в виде блок-схемы (2 занятия).

5) Машина Тьюринга.

6) Машина Поста.

7) Нормальные алгорифмы Маркова.

8) Знакомство со средой Microsoft Visual Studio. Создание первой программы на языке Си.

9) Программирование линейных алгоритмов.

10) Программирование разветвляющихся алгоритмов (условный оператор).

11) Программирование разветвляющихся алгоритмов (оператор выбора).

12) Использование подпрограмм.

13) Отладка и тестирование программ.

14) Программирование циклических алгоритмов, работа с массивами.

15) Работа с двумерными массивами.

Примерные темы практических работ в 2 семестре

1) Работа с текстовыми файлами: ввод и вывод.

2) Работа с текстовыми файлами. Аргументы командной строки

3) Работа с текстовыми файлами. Пакетная обработка файлов.

4) Работа с бинарными файлами.

5) Динамическое выделение памяти.

6) Модульное программирование (2 занятия).

7) Работа со структурами (4 занятия).

8) Потоковый ввод/вывод. Перегрузка операций.

9) Управление исключениями.

10) Перегрузка, передача параметров по умолчанию, шаблоны, inline-функции.

## Рубежный контроль (аттестация раздела) проводится в виде контрольных работ и индивидуальных домашних заданий.

### Контрольная работа № 1 «Составление блок-схемы алгоритма» (семестр 1).

Время проведения контрольной работы - 6 неделя.

Варианты контрольной работы:

1. Вывести на экран все двузначные числа, сумма цифр которых не меняется при умножении числа на 2.

2. Вывести на экран все трехзначные числа, сумма цифр которых равна заданному целому числу.

3. Вывести на экран все трехзначные числа, средняя цифра которых равна сумме первой и третьей цифр.

4. Определить и вывести на экран делители заданного числа.

5. Вывести на экран все двузначные числа, сумма квадратов цифр которых делится на 17.

6. Найти и вывести на экран факториал заданного числа. Примечание: ( – обозначение для факториала числа n).

7. Найти и вывести на экран такое двузначное число, куб суммы цифр которого равен квадрату самого числа.

8. Найти и вывести на экран двузначное число, равное утроенному произведению его цифр.

9. Даны два целых числа A и B (A < B). Вывести все целые числа, расположенные между данными числами (не включая сами эти числа), в порядке их убывания, а также количество этих чисел.

10. Дано целое число N > 1. Вывести наибольшее целое K, при котором выполняется неравенство 3K < N, и само значение 3K.

11. Дано вещественное число A и целое число N > 0. Вычислить при помощи цикла и вывести на экран значение AN.

12. Дано вещественное число A и целое число N > 0. Вывести на экран значение суммы A + 2A + 3A + ... + NA.

13. Дано число A и целое число B > 0. Вывести на экран значение суммы A/B + 2A/(B-1) + 3A/(B-2) + ... + BA/1.

14. Определить и вывести на экран, является ли заданное число совершенным, то есть равным сумме всех своих положительных делителей, кроме самого этого числа (например, число 6 совершенно: 6=1+2+3).

15. Составить алгоритм вывода всех двузначных чисел, сумма квадратов цифр которых делится на 13.

16. Дано натуральное число n. Вычислить и вывести на экран .

17. Дано натуральное число n. Вычислить и вывести на экран

18. Дано натуральное число n. Вычислить и вывести на экран .

19. Дано десятизначное число. Определить и вывести на экран, упорядочены ли цифры в записи этого числа по убыванию.

20. Дано десятизначное число. Определить и вывести на экран, верно ли, что число состоит только из 0 и 1.

### Контрольная работа № 2 «Базовые алгоритмические структуры и подпрограммы» (семестр 1)

Время проведения контрольной работы - 18 неделя.

Варианты контрольной работы:

Написать функцию в соответствии с заданием и продемонстрировать её работу в программе.

1. Функция принимает натуральное число и возвращает произведение чисел, представленных цифрами этого числа.

2. Функция принимает два целых числа и возвращает остаток от деления первого числа на второе. Функция не должна использовать операцию встроенную операцию взятия по модулю %.

3. Функция принимает вещественное число и возвращает другое вещественное число – дробную часть исходного числа (с учетом его знака).

4. Функция принимает два целых числа и возвращает результат возведения одного числа в степень, равную второму числу. Функция не должна использовать функцию pow из библиотеки math.h.

5. Функция принимает целое число и возвращает число 2, возведенное в степень, равную этому числу. Функция не должна использовать функцию pow из библиотеки math.h.

6. Функция принимает целое число и возвращает факториал этого числа.

7. Функция принимает целое число и возвращает его максимальный делитель (не являющийся самим этим числом).

8. Функция принимает два целых числа и возвращает наибольший общий делитель этих чисел.

9. Функция принимает два целых числа и возвращает наименьшее общее кратное этих чисел.

10. Функция принимает целое число и возвращает исходное число, цифры в котором поменяны местами, если это число двузначное, и 0 в остальных случаях.

11. Функция принимает целое число и возвращает исходное число, цифры в котором записаны в обратном порядке.

12. Функция принимает 8 вещественных чисел – координат выпуклого четырехугольника, и возвращает его площадь.

13. Функция принимает 3 целых числа – часы, минуты и секунды, и возвращает время в секундах.

14. Функция принимает шестизначное число и возвращает 1, если данное число является счастливым (сумма первых трех его цифр равна сумме последних трех его цифр), и 0 в противном случае.

15. Функция принимает целое число и возвращает 1, если число является простым числом Софи Жермен (такое простое число p, что число 2p + 1 также простое), и 0 в противном случае.

16. Функция принимает целое число и возвращает 1, если число является избыточным (таким, что сумма всех его положительных делителей (отличных от самого числа) превышает само число).

17. Функция принимает целое число и возвращает 1, если это число является автоморфным (десятичная запись квадрата которого оканчивается цифрами самого этого числа), и 0 в противном случае.

18. Функция принимает целое число и возвращает 1, если это число является триморфным (десятичная запись куба которого оканчивается цифрами самого этого числа), и 0 в противном случае.

19. Функция принимает натуральное число и возвращает сумму чисел, представленных цифрами этого числа.

20. Функция принимает целое число и возвращает 1, если это число является тау-числом (делящемся на число своих делителей), и 0 в противном случае.

### Контрольная работа № 1 «Работа с файлами» (семестр 2).

Время проведения контрольной работы - 4 неделя.

Варианты контрольной работы:

1) написать программу для определения количества гласных букв в файле; имя файла задается из командной строки.

2) написать программу для определения количества согласных букв в файле; имя файла задается из командной строки.

### Контрольная работа № 2 «Работа со структурами» (семестр 2)

Время проведения контрольной работы - 8 неделя.

Составить программу реализующую ввод данных и запись их в элементы структуры «Мел» (цвет, количество, цена; партия мела одного цвета может иметь разную цену).

Варианты контрольной работы:

1) После окончания ввода программа должна вывести список имеющегося мела заданного цвета.

2) После окончания ввода программа должна вывести список имеющегося мела количество, которого самое большое.

### Контрольная работа № 3 «Итоговая контрольная работа» (семестр 2)

Время проведения контрольной работы - 16 неделя.

Контрольная работа проводится в виде тестирования по изученным темам.

### Индивидуальное домашнее задание № 1 «Машина Тьюринга» (1 семестр)

Индивидуальное домашнее задание выдается на 5 неделе.

Срок сдачи индивидуального домашнего задания - 8 неделя.

Варианты заданий:

1.1) Построить машину Тьюринга, вычисляющую функцию: U(a|, b|) = a+b, где a и b заданы в унарной форме. Ответ получить в виде натурального числа.

1.2) Построить машину Тьюринга, реализующее обращение слова из 0 и 1. Если на входе 1010010, на выходе должно быть 0100101.

2.1) Построить машину Тьюринга, вычисляющую функцию: U(a|, b|) = a-b, где a и b заданы в унарной форме, a < b. Ответ получить в виде натурального числа.

2.2) Построить машину Тьюринга, сортирующую нули и единицы в двоичном слове. Если на входе 10011010, на выходе должно быть 00001111.

3.1) Построить машину Тьюринга, вычисляющую функцию: U(a|) = 2 \* a + 1, где a задано в унарной форме. Ответ получить в виде натурального числа.

3.2) Построить машину Тьюринга, вычисляющую функцию: U(a|)=a2 – преобразование числа из унарной записи в двоичную.

4.1) Построить машину Тьюринга, вычисляющую функцию: U(a|,b|) = (a-b)|, где a и b заданы в унарной форме. Ответ получить унарной форме.

4.2) Построить машину Тьюринга, удваивающую строку из 1. Если на входе 111, на выходе должно быть 111111.

### Индивидуальное домашнее задание № 2 «Обработка строк» (1 семестр)

Индивидуальное домашнее задание выдается на 9 неделе.

Срок сдачи индивидуального домашнего задания - 17 неделя.

Задание: написать функцию, реализующую операцию со строками, не используя специализированные функции для работы со строками библиотеки С.

Варианты задач:

1) int strcat\_s(char \*strDestination, int number-OfElements, const char \*strSource) – добавляет strSource к strDestination, а затем к результирующей строке завершающий нуль-символ. Начальный символ strSource пере-записывает конечный нуль-символ strDestination. Возвраща-ет 0, в случае успеха, в противном случае – число, отличное от 0.

Подробная информация о функции:

https://msdn.microsoft.com/ru-ru/library/d45bbxx4.aspx

2) char \*strchr(const char \*string, int sym) – нахо-дит первое вхождение sym в str и возвращает указатель на это вхождение или возвращает NULL, если sym не найден. Нулевой конечный символ включен в поиск.

Подробная информация о функции:

https://msdn.microsoft.com/ru-ru/library/b34ccac3.aspx

3) int strcmp(const char \*string1, const char \*string2) – сравнивает string1 и string2 лексикографически и возвращает значение, которое указывает их отношение: -1, если string1 меньше чем string2; 0, если string1 идентична string2; 1, если string1 больше чем string2.

Подробная информация о функции:

https://msdn.microsoft.com/ru-ru/library/e0z9k731.aspx

4) int \_stricmp(const char \*string1, const char \*string2) – лексикографически сравнивает версии string1 и string2 в нижнем регистре и возвращает значение, показыва-ющее их взаимосвязь: -1, если string1 меньше чем string2; 0, если string1 идентична string2; 1, если string1 больше чем string2.

Подробная информация о функции:

https://msdn.microsoft.com/ru-ru/library/k59z8dwe.aspx

5) int strcpy\_s(char \*strDestination, int number-OfElements, const char \*strSource) – копирует содержимое в адресе strSource, включая конечный нуль-символ, в распо-ложение, указанное strDestination. Строка назначения должна быть достаточно велика для хранения строки источника и его конечного нуль-символа. Возвращает 0 в случае успешного вы-полнения; в противном случае – отличное от 0 значение.

Подробная информация о функции:

https://msdn.microsoft.com/ru-ru/library/td1esda9.aspx

6) char \*ucwords(char \*string) – преобразует в верхний регистр первый символ каждого слова в строке. Возвращает строку, где первая буква каждого слова записана в верхнем ре-гистре. Под словом данная функция понимает последователь-ность одной и более букв, которая располагается в начале стро-ки, либо которой предшествует не буква и не цифра.

7) int strrpos(const char \*string, const char \* str) – возвращает позицию последнего вхождения str в string. Если str не входит в string, возвращает -1.

8) char \*strstr(const char \*str, const char \*strSearch) – возвращает указатель на первое вхождение strSearch в str. Поиск не распространяется на завершающий нулевой символ. Возвращает NULL, если strSearch не появляет-ся в str. Если strSearch указывает на строку нулевой длины, то функция возвращает str.

Подробная информация о функции:

http://msdn.microsoft.com/ru-ru/library/z9da80kz.aspx

9) char \*strtrim(char \*string) – удаляет пробелы, сим-волы табуляции и перевода строки в начале и в конце строки. Возвращает указатель на строку, в которой отсутствуют пробе-лы, символы табуляции и перевода строки в начале и в конце строки.

10) int strewth(const char \*str, int numberOfElements, const char \*endStr, int length) – определяет, заканчивается ли строка str размером numberOfElements строкой endStr размером length. Возвраща-ет 1 в случае, если str заканчивается endStr, 0 – в противном случае.

11) char \*\_strspnp(const char \*str, const char \*charset) – возвращает указатель на первый символ в заданной строке str, не входящий в другую заданную строку charset. Если такого символа нет, возвращает NULL.

Подробная информация о функции:

https://msdn.microsoft.com/ru-ru/library/k254awe8.aspx

12) int strcspn(const char \*str, const char \*strCharSet) – возвращает индекс первого вхождения символа в строку str, принадлежащего набору символов strCharSet. Если в str нет символов из strCharSet, то возвращается длина str.

Подробная информация о функции:

https://msdn.microsoft.com/ru-ru/library/xe8sk0x7.aspx

13) int strncat\_s(char \*strDest, int numberOfEle-ments, const char \*strSource, int count) – пытается доба-вить первые D символов строки strSource в конец строки strDest, где D – меньшее из величины count и длины strSource. Если эти добавляемые D символов поместятся внут-ри strDest (чей размер задается как numberOfElements), и по-прежнему останется место для завершающего нуль-символа, то-гда эти символы добавляются, начиная с исходного нуль-символа strDest, и добавляется новый завершающий нуль-символ; в противном случае strDest[0] становится равным нуль-символу, и возвращается отличное от 0 значение. В случае успеха возвращает 0.

Подробная информация о функции:

https://msdn.microsoft.com/ru-ru/library/w6w3kbaf.aspx

14) int \_strnset\_s(char \*str, int numberOfElements, int c, int count) – задаёт, по крайней мере, первые count символов str в c. Если count больше, чем размер str, размер str используется вместо count. В случае успеха возвращает 0. Ошибка возникает, если count больше numberOfElements, и оба этих параметра больше размера str, тогда возвращает значение, отличное от нуля.

Подробная информация о функции:

https://msdn.microsoft.com/ru-ru/library/ms175783.aspx

15) char \*strpbrk(const char \*str, const char \*strCharSet) – возвращает указатель на первое вхождение символа, принадлежащего набору символов strCharSet, в str. Поиск не распространяется на завершающий нулевой символ. Возвращает указатель на первое вхождение любого символа из strCharSet в str или указатель на NULL, если строковые аргу-менты не имеют общих символов.

Подробная информация о функции:

https://msdn.microsoft.com/ru-ru/library/217yyhy9.aspx

16) char \*strrchr(const char \*str, int c) – находит последнее вхождение c (преобразованного к char) в str. Поиск включает конечный нуль-символ. Возвращает указатель на по-следнее вхождение c в str или NULL, если c не найден.

Подробная информация о функции:

https://msdn.microsoft.com/ru-ru/library/ftw0heb9.aspx

17) char \*\_strrev(char \*str) – обращает порядок сим-волов в str. Завершающий нулевой символ остается на месте. Возвращает указатель на измененную строку.

Подробная информация о функции:

https://msdn.microsoft.com/ru-ru/library/9hby7w40.aspx

18) int strspn(const char \*str, const char \*strCharSet) – возвращает индекс первого символа в str, ко-торый не входит в набор символов strCharSet. Поиск не рас-пространяется на завершающий нулевой символ. Если str начинается с символа, не содержащегося в strCharSet, функция возвращает 0.

Подробная информация о функции:

https://msdn.microsoft.com/ru-ru/library/kthzzay9.aspx

19) int strins\_s(char \*str, int numberOfElements, const char\* subStr, int length, int position) – вставляет подстроку subStr длиной length в строку str длиной number-OfElements, начиная с позиции position. В случае, если вставка прошла успешно, возвращает 0, иначе – 1.

20) char \*strprc\_s(const char \*str, int numberOfEl-ements, const char \*strExt, int length) – определяет, яв-ляется ли какая-либо конечная часть строки str длиной number-OfElements начальной частью строки strExt длиной length. Возвращает указатель на символ, начиная с которого окончание строки str является началом строки strExt, и NULL, если окон-чание строки str не является началом строки strExt.

### Индивидуальное домашнее задание № 1 «Разработка алгоритма» (2 семестр)

Индивидуальное домашнее задание выдается на 1 неделе.

Срок сдачи индивидуального домашнего задания - 8 неделя.

Варианты заданий:

1) Проверить знаки препинания в тексте.

2) Реализовать алгоритм пирамидальной сортировки.

3) Реализовать алгоритм поразрядной сортировки.

4) Вывод сортированного по алфавиту списка файлов в каталоге (аналог программы dir).

5) Замена одного текста на другой внутри файла.

6) Замена одних слов на другие в списке файлов.

7) Копирование файлов (аналог программы xcopy.exe).

8) Определить кодировку текста DOS866 или Windows 1251 и, если необходимо выполнить перекодировку.

9) Поиск информации в тексте файла.

10) Поиск файлов в каталоге, содержащих заданный текст (аналог программы find).

11) Создать дерево каталогов (аналог программы tree).

12) Составить алфавитный словарь слов текста.

13) Слить два алфавитных словаря в один.

14) Соединить строки в абзацы.

15) Создать программу схожую с программой заставки Norton Commander (звездное небо).

16) Создать программу схожую с программой заставки Far (сквозь вселенную).

17) Сравнение директорий, с выводом разницы (проверка находящихся файлов в директории по имени и размеру).

18) Проверить структуру текста на соответствие Си.

### Индивидуальное домашнее задание № 2 «Разработка программы» (2 семестр)

Индивидуальное домашнее задание выдается на 8 неделе.

Срок сдачи индивидуального домашнего задания - 12 неделя.

Задание заключается в разработке программы на языке Си, алгоритм которой разработан при выполнении домашнего задания 1.

### Индивидуальное домашнее задание № 3 «Методологии программирования» (2 семестр)

Индивидуальное домашнее задание выдается на 12 неделе.

Срок сдачи индивидуального домашнего задания - 16 неделя.

Задание: подготовка доклада по истории развития языков программирования.

Варианты заданий:

1) FORTRAN

2) ALGOL

3) COBOL

4) BASIC

5) Pascal

6) C

7) Ada

8) Simula

9) Smalltalk

10) Cecil

11) C++

12) Object Pascal

13) Java

14) C#

15) LISP

16) РЕФАЛ

17) Scheme

18) FP

19) ML

20) Miranda

21) Haskell

22) Scala

23) F#

24) Prolog

25) LQF

26) Gödel

27) LOGLISP

28) Mercury

29) PARLOG

30) GHC

## Промежуточная аттестация во 1 семестре выполняется в виде зачета, в 2 семестре – в виде экзамена.

Примерный перечень вопросов к зачету

1) Введение в алгоритмы. Что такое алгоритм. Свойства алгоритмов.

2) Способы записи алгоритмов. Словесный, псевдокод.

3) Способы записи алгоритмов. Графический (блок - схема),

4) Связь алгоритма и программы на языке программирования.

5) Достоинства и недостатки машинных языков. Преимущества алгоритмических языков перед машинными языками.

6) Правила записи арифметических выражений. Правила записи логических выражений.

7) Представление основных алгоритмических структур: итерация, ветвление, повторение.

8) Процедуры и функции в алгоритмах.

9) Типы данных. Представление целых типов в компьютере.

10) Типы данных. Представление вещественных чисел в компьютере.

11) Типы данных. Представление символьных типов данных. Понятие кодировка. Строки символов.

12) Программа на языке высокого уровня: стандартные типы данных.

13) Программа на языке высокого уровня: общая структура программы. Время существования и видимость переменных. Классы памяти. Препроцессор.

14) Программа на языке высокого уровня: функции, формальные и фактические параметры. Указатели в качестве аргументов функции. Предварительное описание функций. Аргументы командной строки.

15) Программа на языке высокого уровня: адреса и указатели. Отождествление массивов и указателей. Адресная арифметика. Указатели на массивы и многомерные массивы.

16) Программа на языке высокого уровня: ввод и вывод, работа с потоками и файлами.

17) Программа на языке высокого уровня: арифметические операции и выражения. Операции отношения, логические операции и выражения. Преобразование типов и операция приведения.

18) Программа на языке высокого уровня: представление основных структур программирования: итерация, ветвление, повторение.

Примерный перечень вопросов к экзамену

1) Адреса и указатели, операция косвенной адресации (разыменования), выражения и арифметические операции с указателями.

2) Аргументы командной строки. Чтение и запись из файлов.

3) Битовые поля.

4) Понятие алгоритма. Свойства алгоритмов.

5) Ввод информации в программу.

6) Вывод информации из программы.

7) Динамическое выделение памяти.

8) Императивное программирование.

9) Классы памяти.

10) Компилятор. Схемы работы.

11) Константы и переменные, описание данных в программе, объявление констант.

12) Константы и переменные, описание данных в программе, объявление переменных.

13) Константы и переменные, определение размера данных.

14) Логическое программирование.

15) Массивы.

16) Машина Тьюринга.

17) Многомерные массивы.

18) Модификатор const. Передача параметров по значению. Передача функции указателя.

19) Область видимости и время жизни идентификаторов.

20) Оператор присваивания. Автоматическое преобразование типов.

21) Оператор присваивания. Инициализация переменных.

22) Оператор присваивания. Операция приведения типа.

23) Передача параметров в функцию через указатель.

24) Перечислимые константы.

25) Поразрядные операции.

26) Правила записи арифметических выражений. Правила записи логических выражений.

27) Представление основных алгоритмических структур: следование, ветвление, цикл.

28) Программа на языке высокого уровня: общая структура программы.

29) Связь между указателями и массивами.

30) Символьные константы, строковые константы.

31) Символьные массивы.

32) Способы записи алгоритмов. Графический (блок - схема).

33) Структурное программирование.

34) Структуры. Создание пользовательского типа.

35) Текстовые файлы и бинарные файлы.

36) Типы данных. Вещественные данные.

37) Типы данных. Логические данные. Логические операторы и выражения.

38) Типы данных. Массивы данных.

39) Типы данных. Машинный эпсилон.

40) Типы данных. Символьные данные.

41) Типы данных. Целочисленные данные.

42) Транслятор. Схемы работы.

43) Транслятор. Схемы работы.

44) Условный оператор if-else.

45) Формализация понятия алгоритма.

46) Функции для обработки символьных массивов.

47) Функции. Прототипы функций.

48) Функции. Аргументы по умолчанию.

49) Функции. Шаблоны функций.

50) Перегрузка функций

51) Функции. Формальные и фактические параметры функций.

52) Функциональное программирование.

53) Целые константы, константы с плавающей точкой.

54) Цикл for. Операторы break и continue.

55) Цикл while. Операторы break и continue.

56) Эмулятор. Схемы работы.

57) Язык С. Основные свойства языка С.

58) Язык С. Элементы языка С.

59) Языки программирования. Общие особенности языков программирования и трансляторов.

60) Языки программирования. Основные понятия: транслятор, компилятор, интерпретатор, эмулятор.

61) Перегрузка операций

## Самостоятельная работа студента

Самостоятельная работа студента заключается в подготовке к лекциям, практическим занятиям, контрольным работам и выполнении индивидуальных домашних заданий.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Семестр | Раздел учебной дисциплины | Виды СРС | Часов |
| 1 | 1 | Все | ПЛ1-ПЛ18, ПП1-ПП18 | 18 |
| 2 | 1 | Алгоритмизация и программирование | ПК1 | 4 |
| 3 | 1 | Формализация понятия алгоритма | ДЗ1 | 4 |
| 4 | 1 | Типы данных | ПК2 | 4 |
| 5 | 1 | Язык Си | ПК2 | 4 |
| 6 | 1 | Структурное программирование на языке Си | ПК2 | 4 |
| 7 | 1 | Структурное программирование на языке Си | ДЗ2 | 6 |
| 8 | 2 | Все | ПЛ1-ПЛ16, ПП1-ПП16 | 8 |
| 9 | 2 | Пользовательские типы данных в языке Си | ПК1 | 2 |
| 10 | 2 | Пользовательские типы данных в языке Си | ДЗ1 | 2 |
| 11 | 2 | Основные возможности языка Си++, расширяющие язык Си | ПК2 | 2 |
| 12 | 2 | Основные возможности языка Си++, расширяющие язык Си | ДЗ2 | 2 |
| 13 | 2 | Языки программирования и трансляция | ПК3 | 2 |
| 14 | 2 | Методологии программирования | ПК3 | 1 |
| 15 | 2 | Методологии программирования | ДЗ3 | 1 |

ДЗ — индивидуальное домашнее задание, ПЛ — подготовка к лекциям, ПП — подготовка к практическим занятиям, ПК — подготовка к контрольной работе.

# УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

## Основная литература:

### Давыдов В.Г. Программирование и основы алгоритмизации: Учеб. пособие/ В.Г. Давыдов. – 2-е изд., стер.-М.: Высш.шк., 2005. – 447 с.: ил.

### Тюгашев А. Языки программирования: учебное пособие. Стандарт третьего поколения. – СПб.: Питер, 2014. – 366 с.: ил. – (Серия «Учебник для вузов»).

### Шереметьев, А.И. Информатика [Электронный ресурс] : учебное пособие / А. И. Шереметьев. - Москва : НИЯУ МИФИ.Ч.1 : Элементы программирования на языке Си. - [Б. м.], 2011. (Центр информационно-библиотечного обеспечения учебно-научной деятельности НИЯУ МИФИ. Электронно-библиотечная система http://libcatalog.mephi.ru/)

### Орлов С. А. Теория и практика языков программирования: Учебник для вузов. Стандарт 3-го поколения. – СПб.: Питер, 2014. – 688 с.: ил.

### Головин И.Г. Языки и методы программирования : учебник для студ. учреждений высш. проф. образования / И.Г. Головин, И.А. Волкова. – М. : Издательский центр «Академия», 2012. – 304 с. – (Сер. Бакалавриат).

### Одинцов И.О. Профессиональное программирование. Системный подход.— 2-е изд. перераб. и доп. — СПб.: БХВ-Петербург, 2014. — 611 с.: ил. (электронная библиотека ibooks.ru: http://ibooks.ru/reading.php?productid=18535)

## Дополнительная литература:

### Семакин И.Г. Основы алгоритмизации и программирования/ И.Г.Семакин, А.П . Шеста-ков. -М: Академия, 2013. -304 с.

### Семакин И.Г. Основы алгоритмизации и программирования, Практикум : уч. пособие по СПО/ И.Г Семакин. - М: Академия, 2013. -144 с.

## Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

### MSDN Library http://msdn.microsoft.com

### Портал «Информационно-коммуникационные технологии в образовании» http://www.ict.edu.ru/

### Электронно-библиотечная система IQlib http://www.iqlib.ru/

# МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

## Лекции проводятся в аудиторном классе, оборудованном доской, компьютером, мультимедийным проектором, экраном для проектора.

## Практические занятия проводятся в компьютерном классе (11 компьютеров).

Требуемое программное обеспечение

|  |  |  |
| --- | --- | --- |
| № | Программный продукт | Количество |
| 1 | Операционная система Microsoft Windows XP, 7, 8 | 1 шт. на компьютер |
| 2 | Среда программирования Microsoft Visual Studio .NET | 1 шт. на компьютер |
| 3 | Файловый менеджер FAR | 1 шт. на компьютер |
| 4 | Программа для просмотра документов формата PDF | 1 шт. на компьютер |

Программа составлена в соответствии с требованиями ОС НИЯУ МИФИ по направлению подготовки (специальности):

09.03.01 Информатика и вычислительная техника

|  |  |
| --- | --- |
| Автор(ы) | Е.Л. Шершнев |
| Рецензент(ы) | Малышев А.Ю., начальник отдела программного обеспечения СИТ ФГУП «ПО «МАЯК» |
| Программа одобрена на заседании  методического совета кафедры | 24.05.2021 протокол №5 |