

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

**«УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра Высокопроизводительных вычислительных технологий и систем

АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ

УЧЕБНОЙ ДИСЦИПЛИНЫ

«ПРОГРАММИРОВАНИЕ»

Уровень подготовки: высшее образование – бакалавриат

Направление подготовки бакалавров

02.03.01 Математика и компьютерные науки

(код и наименование направления подготовки)

Направленность подготовки

Математическое и компьютерное моделирование

Квалификация (степень) выпускника

бакалавр

Форма обучения

очная

Уфа 2015

Исполнители:

доцент
должность


подпись

А.Т. Бикмеев
расшифровка подписи

ст. преподаватель
должность


подпись

А.А. Касаткин
расшифровка подписи

ассистент
должность


подпись

А.А. Гайнетдинова
расшифровка подписи

Заведующий кафедрой
ВВТиС


подпись

Р.К. Газизов
расшифровка подписи

1. Место дисциплины в структуре образовательной программы

Дисциплина «Программирование» является дисциплиной *базовой* части ОПОП по направлению подготовки бакалавров 02.03.01 Математика и компьютерные науки, направленность: Математическое и компьютерное моделирование.

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки 01.04.04 «Прикладная математика», утвержденного приказом Министерства образования и науки Российской Федерации от 07.08.2014 г. № 949. Является неотъемлемой частью основной образовательной профессиональной программы (ОПОП).

Целью освоения дисциплины является изучение общих подходов к построению алгоритмов решения задач и отработка этих подходов с использованием современных сред разработки программного обеспечения.

Задачи:

- сформировать знания о современных языках программирования;
- освоить современные технологии программирования, работу с сетью Internet;
- выработать способность самостоятельного изучения нового материала.

Предшествующими знаниями, на которых непосредственно базируется дисциплина «Программирование» являются знания, полученные при изучении школьных курсов информатики и языков программирования.

Дисциплинами, для которых освоение дисциплины «Программирование» необходимо как предшествующее являются: «Компьютерная графика», «Базы данных», «Операционные системы и сети», «Параллельное программирование», «Численные методы», а также ряд дисциплин по выбору.

2. Перечень результатов обучения

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций. Планируемые результаты обучения по дисциплине

| № | Формируемые компетенции | Код | Знать | Уметь | Владеть навыками |
|---|--|-------|---|--|---|
| 1 | способность находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем | ОПК-4 | современные способы конструирования программ, включая технологии, методы и средства программирования; | разработать приложения с графическим интерфейсом ОС Windows; | программирования в современных операционных средах; |
| 2 | способность использовать современные | ПКП-1 | типы и структуры данных и методы их обработки; способы | использовать различные механизмы | структурного программирования на языке Си; |

| | | | | |
|---|--|--|---|---|
| прикладные программные средства и осваивать современные технологии программирования | | и механизмы управления данными; основные этапы решения задач на ЭВМ; конкретный язык программирования; набор функций стандартной библиотеки; | управления данными; проводить разработку и анализ алгоритмов; программировать алгоритм, используя средства языка высокого уровня; использовать типовые алгоритмические конструкции; | объектно-ориентированного программирования; написания программы для ЭВМ, проведения ее отладки и тестирования, оформления документации на программу |
|---|--|--|---|---|

3. Содержание и структура дисциплины (модуля)

Общая трудоемкость дисциплины составляет 12 зачетных единиц (432 часа).

| № | Наименование и содержание раздела |
|---|---|
| | 1 семестр |
| 1 | <p>Введение в языки программирования Основные этапы решения задач на ЭВМ. Понятие алгоритма для ЭВМ, свойства алгоритмов. Базовые конструкции для записи алгоритмов. Примеры. Понятие языка программирования. Виды языков программирования: машинные, машинно-ориентированные (ассемблера), высокого уровня. История развития языков программирования, основные языки программирования, примеры. Современные способы конструирования программ. Понятие структурного и объектно-ориентированного программирования. Типовые среды разработки программ, интегрированные среды. Среды программирования языков Си и Си++. Методы и основные этапы трансляции. Основные понятия языков программирования: алфавит и лексемы, идентификаторы, служебные слова, константы, знаки операций. Синтаксис и семантика языка программирования. Формальные способы описания языков программирования. Алфавит, лексемы и операции языка Си, приоритет операций.</p> |
| 2 | <p>Механизмы управления данными, функции и процедуры Типы данных: понятие типа, классификация типов. Простые типы: перечислимые, числовые, символьные, логические, ссылки, указатели. Составные типы: массив, структура, файл, класс. Базовые (встроенные) типы языка Си, размеры базовых типов. Массивы в языке Си. Явное и неявное приведение типов. Контроль типов: статическая и динамическая типизация. Способы и механизмы управления данными, понятие управляющей структуры, виды управляющих структур: структуры следования, выбора и повторения. Структуры с единственным, двойным и множественным выбором. Структуры повторения: циклы с предусловием, постусловием, итерационные циклы. Синтаксис управляющих структур языка Си, операторы передачи управления. Итерационные алгоритмы. Понятие процедуры и функции, структура программы. Особенности функций в языке Си. Описание функции и ее вызов. Заголовочные файлы. Функции ввода-вывода. Встраиваемые функции. Особенности передачи массивов в функции. Рекуррентные вычисления, рекурсивные алгоритмы, рекурсивные функции. Взаимосвязь итерации и рекурсии. Перегрузка функций. Шаблоны функций.</p> |
| 3 | Типы данных |

| | |
|---|--|
| | <p>Понятие ссылки, передача значений в функцию по ссылке, особенности ссылок в языке Си. Понятие указателя как типа данных, объявление и инициализация указателей в языке Си. Типы указателей. Операции над указателями: разыменованье, преобразование типов, присваивание, получение адреса, аддитивные операции, инкремент и декремент, сравнение. Взаимосвязь массивов и указателей, запись указатель-индекс и указатель-смещение, указатели и многомерные массивы, массивы указателей, массивы динамической памяти. Указатели на функции.</p> <p>Символы, строки, символьные и литеральные константы. Особенности обработки строк в языке Си. Функции работы со строками в Си.</p> |
| 4 | <p>Алгоритмы сортировки</p> <p>Базовая операция сортировки. Оценка трудоемкости алгоритмов сортировки. Типовые алгоритмы: пузырьковая сортировка, сортировка вставками, сортировка подсчетом, сортировка Шелла, быстрая сортировка.</p> |
| 5 | <p>Организация ввода-вывода</p> <p>Организация ввода-вывода, стандартные потоки ввода-вывода, буферизация ввода-вывода. Система ввода-вывода языка Си. Функции консольного ввода-вывода, чтение и запись символов и строк. Форматированный ввод-вывод данных, спецификаторы и модификаторы формата. Понятие файловой системы, файлы последовательного и прямого доступа. Работа с файлами в языке Си.</p> |
| 6 | <p>Препроцессорная обработка программы</p> <p>Препроцессорная обработка программы на языке Си. Директивы препроцессора. Препроцессорные макросы. Организация условной компиляции с помощью препроцессорных директив.</p> |
| | <p>2 семестр</p> |
| 7 | <p>Основы объектно-ориентированного программирования</p> <p>Понятие структуры и объединения. Объявление и инициализация структур и объединений в языке Си. Связь структур и функций. Динамические структуры. Работа с битовыми полями структуры.</p> <p>Объектный подход к программированию. Понятие класса и объекта, компоненты класса. Классы в языке Си++: структура класса, его объявление, компонентные данные и компонентные функции, обращение к компонентам класса. Конструкторы и деструкторы классов. Доступ к компонентам класса, спецификаторы доступа. Статические компоненты класса. Указатели на компоненты класса, указатель this. Дружественные функции классов.</p> <p>Перегрузка операций на объекты классов, операторные функции, ограничения на перегрузку в языке Си++.</p> <p>Наследование классов, базовые и производные классы. Организация наследования классов в языке Си++. Определение статуса доступа к компонентам класса при наследовании. Множественное наследование. Полиморфные классы и виртуальные функции. Абстрактные классы. Локальные классы. Связь классов и шаблонов.</p> <p>Библиотека потокового ввода-вывода языка Си++. Стандартные потоки ввода-вывода и работа с ними. Форматированный ввод-вывод в Си++, флаги форматирования, манипуляторы потоков. Перегрузка операций помещения в поток и выборки из потока. Компонентные функции классов для работы с потоками. Работа с файлами в Си++.</p> |
| 8 | <p>Обработка исключительных ситуаций. Парадигмы программирования</p> <p>Понятие исключения, механизм обработки исключений в языке Си++. Обработчики исключений. Исключения при динамическом выделении памяти.</p> <p>Современные технологии программирования, методы программирования. Парадигмы и стили программирования. Последовательная и параллельная модели программирования: модели параллелизма данных, разделяемой памяти, передачи сообщений. Конструкции распределенного и параллельного программирования.</p> |
| 9 | <p>Разработка программ с графическим интерфейсом</p> <p>Разработка программ с графическим пользовательским интерфейсом под ОС Windows. Элементы графического интерфейса: окна, формы, диалоги, меню. Современные среды и</p> |

| | |
|----|--|
| | технологии разработки приложений с графическим интерфейсом. |
| | 3 семестр |
| 10 | <p>Язык C# и его отличия от C/C++ Типичные ошибки при работе с памятью: выход за пределы выделенных участков, обращение к неинициализированным данным или после удаления объекта, утечки памяти. Автоматическое управление памятью в современных языках, сборка мусора и умные указатели. Области памяти (стек, куча и т.д.). Область видимости переменных и время жизни объектов.</p> <p>Платформа .NET. Сравнение процесса компиляции в C, C# и Java. Промежуточный язык, среда выполнения. Сборки .NET в сравнении с библиотеками C/C++.</p> <p>Базовый синтаксис C#. Встроенные типы значений и их преобразование. Арифметические, логические и побитовые операции. Сокращенное вычисление логических выражений. Инициализация полей и локальных переменных. Управляющие конструкции, циклы. Операторы break и continue. Массивы. Многомерные массивы и массивы массивов.</p> <p>Отличие ссылочных типов и типов значений в C#. Классы и структуры. Передача параметров в функцию по значению и по ссылке (ref, out). Пустая ссылка null. Nullable-типы. Перечислимые типы enum.</p> <p>Классы исключений. Порождение и обработка исключений (try, catch, finally, throw).</p> |
| 11 | <p>Объектно-ориентированное программирование в C# Классы. Статические и нестатические члены класса. Наследование. Базовый класс object. Конструкторы, порядок инициализации класса. Цепочка вызова конструкторов. Ключевые слова base и this. Модификаторы доступа (public, private, protected, internal), инкапсуляция.</p> <p>Модификаторы readonly и const для переменных, отличия от C++.</p> <p>Наследование. Интерфейсы, их назначение и отличие от абстрактных классов. Правила множественного наследования. Преобразование ссылочных типов, операция проверки is. Переопределение методов. Использование ключевых слов override и virtual. Полиморфизм. Модификатор sealed.</p> |
| 12 | <p>Дополнительные возможности C# Свойства как члены классов и интерфейсов. Наследование свойств. Индексаторы. Статические классы. Вложенные классы.</p> <p>Атрибуты. Методы расширения классов.</p> <p>Переопределение операций и функций преобразования типов. generic-типы и их отличие от шаблонов C++.</p> <p>Делегаты. Вызов существующего метода с помощью делегатов. Операции с делегатами. Сравнение с функторами и указателями на функции в C++. События класса. Лямбда-функции. Базовые возможности LINQ.</p> |
| 13 | <p>Визуальное и программное создание графического интерфейса приложений Компоненты Windows Forms. Дизайнер форм Visual Studio. Визуальное задание свойств и обработчиков событий. Управляющие элементы и невидимые компоненты. Настройка взаимного расположения управляющих элементов, компоненты-контейнеры. Строки меню, выпадающие меню, панели инструментов.</p> <p>Программное создание графического интерфейса, многооконные приложения, передача данных между формами и компонентами.</p> |
| 14 | <p>Система контроля версий и разделение приложения на файлы Системы контроля версий и совместная разработка приложений. Система git. Фиксация изменений, ветви. Слияние ветвей. Разрешение конфликтов.</p> <p>Выделение логических составляющих приложений в отдельные файлы (классы, интерфейсы, частичные классы, компоненты для дизайнера форм, сборки .NET)</p> |
| 15 | <p>Стандартные линейные структуры данных Оценка сложности операций со структурами данных.</p> <p>Структуры с произвольным доступом по номеру.</p> <p>Хранение набора данных в массиве. Структуры данных на основе массива переменного размера. Примеры в C++ и C#: std::vector и List. Сортировка и двоичный поиск.</p> |

| | |
|----|---|
| | <p>Структуры с доступом к первому и последнему элементу: стек, очередь, дек. Стек на основе массива. Очередь и дек на основе циклического массива. Односвязный и двусвязный список. Использование указателей или номеров элементов для хранения информации о связях. Примеры в C++ и C#: <code>std::linked_list</code>, <code>std::stack</code>, <code>std::queue</code>, <code>std::dequeue</code>, <code>Stack</code>, <code>Queue</code>, <code>LinkedList</code>.</p> <p>Последовательный просмотр элементов коллекций. Итераторы (перечислители) в стандартных библиотеках C# и C++.</p> |
| 16 | <p>Структуры данных для представления графов Матрица смежности, списки смежности, список ребер. Процедура поиска соседних вершин или выходящих ребер. Хранение дополнительной информации о вершинах и ребрах. Процедуры поиска в ширину и в глубину в графе (DFS, BFS).</p> |
| 17 | <p>Структуры данных с автоматической сортировкой объектов Сравнение произвольных объектов. Компараторы. Двоичное дерево поиска (BST). Процедуры поиска, добавления и удаления элементов. Хранение деревьев в памяти. Понятие о сбалансированных деревьях. Красно-черное дерево и AVL-дерево: основные правила построения, принципы работы, примеры. Двоичная куча как частный случай двоичного дерева: хранение в памяти, правила построения, добавление и удаление элемента. Приоритетная очередь на основе дерева поиска или двоичной кучи. Случай повторяющихся элементов (приоритетов).</p> |
| 18 | <p>Множества и ассоциативные массивы Основные доступные операции для множеств и ассоциативных массивов. Построение на основе деревьев поиска множеств и ассоциативных массивов. Примеры таких структур в стандартных библиотеках C++ и C# (<code>std::set</code>, <code>std::map</code>, <code>SortedSet</code>, <code>SortedDictionary</code>). Хранение данных без сортировки. Хэш-код объекта. Свойства и качество хеш-функции. Хеш-таблица. Проблема коллизий. Открытое хеширование (метод цепочек). Закрытое хеширование. Примеры работы.</p> |
| 19 | <p>Разработка многокомпонентных приложений Разделение модель-представление-контроллер (MVC). Передача данных между формами и компонентами. События обновления. Фоновые процессы. Работа с файловой системой, компонентами отображения форматированного текста, списками и таблицами. Обработка текстовых данных. Регулярные выражения.</p> |
| 20 | <p>Абстрактные вычислительные машины Математическое понятие алгоритма. Формализация входных и выходных данных. Машина Тьюринга. Применимость программы к данным, функции, вычислимые по Тьюрингу. Неразрешимые по Тьюрингу задачи. Нормальные алгоритмы Маркова.</p> |
| 21 | <p>Принципы компиляции и анализа текста программ Лексический, синтаксический и семантический анализ. Лексемы, синтаксическое дерево. Формальные грамматики и их использование для синтаксического анализа. Выводимость, дерево разбора. Типы грамматик и иерархия Хомского. Контекстно-свободные грамматики, LL-грамматики. Регулярные грамматики, связь с регулярными выражениями.</p> |

Подробное содержание дисциплины, структура учебных занятий, трудоемкость изучения дисциплины, входные и исходящие компетенции, уровень освоения, определяемый этапом формирования компетенций, учебно-методическое, информационное, материально-техническое обеспечение учебного процесса изложены в рабочей программе дисциплины.

ЗАКЛЮЧЕНИЕ
Научно-методического совета по УГСН
02.00.00 «Компьютерные и информационные науки»

Настоящим подтверждаю, что представленный комплект аннотаций рабочих программ учебных дисциплин по направлению подготовки бакалавров 02.03.01 «Математика и компьютерные науки» по профилю «Математическое и компьютерное моделирование», реализуемой по очной форме обучения соответствует рабочим программам учебных дисциплин указанной выше образовательной программы.

Председатель НМС



Н.И. Юсупова
«27» 05 _____ 2015 г.