

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 (специальность)
01.03.02 Прикладная математика и информатика

Направленность подготовки (профиль, специализация)
Математическое моделирование и вычислительная математика

Квалификация (степень) выпускника
бакалавр

Форма обучения
очная

Исполнители

Бикмеев А.Т.
Касаткин А.А.
Гайнетдинова А.А.

Заведующий кафедрой высокопроизводительных
вычислительных технологий и систем

Газизов Р.К.

Уфа 2015

Место дисциплины в структуре образовательной программы

Дисциплина «Программирование» является дисциплиной базовой части.

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки 01.03.02 Прикладная математика и информатика, утвержденного приказом Министерства образования и науки Российской Федерации от 12.03.2015 г. № 228.

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.

Перечень результатов обучения

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций.

Планируемые результаты обучения по дисциплине

№	Формируемые компетенции	Код	Знать	Уметь	Владеть
1	Способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	ОПК-3	- современные способы конструирования программ, включая технологии, методы и средства программирования; - основные этапы решения задач на ЭВМ; - конкретный язык программирования; - набор функций стандартной библиотеки; - типы и структуры данных и методы их обработки; - способы и механизмы управления данными.	- разработать приложения с графическим интерфейсом под ОС Windows; - проводить разработку и анализ алгоритмов; - программировать алгоритм, используя средства языка высокого уровня; - использовать типовые алгоритмические конструкции; использовать - различные механизмы управления данными.	- навыками программирования в современных операционных средах; - навыками объектно-ориентированного программирования; - навыками написания программы для ЭВМ, проведения ее отладки и тестирования, оформления документации на программу; - навыками структурного программирования на языке Си.

Содержание разделов дисциплины

№	Наименование и содержание разделов
1	<p>Введение в языки программирования.</p> <p>Основные этапы решения задач на ЭВМ. Понятие алгоритма для ЭВМ, свойства алгоритмов. Базовые конструкции для записи алгоритмов. Примеры.</p> <p>Понятие языка программирования. Виды языков программирования: машинные, машинно-ориентированные (ассемблера), высокого уровня. История развития языков программирования, основные языки программирования, примеры.</p> <p>Современные способы конструирования программ. Понятие структурного и объектно-ориентированного программирования. Типовые среды разработки программ, интегрированные среды. Среда программирования языков Си и Си++. Методы и основные этапы трансляции.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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	<p>Типы данных.</p> <p>Понятие ссылки, передача значений в функцию по ссылке, особенности ссылок в языке Си.</p> <p>Понятие указателя как типа данных, объявление и инициализация указателей в языке Си. Типы указателей. Операции над указателями: разыменование, 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>Прекомпиляторная обработка программы. Прекомпиляторная обработка программы на языке Си. Директивы прекомпилятора. Прекомпиляторные макросы. Организация условной компиляции с помощью прекомпиляторных директив.</p>
7	<p>Основы объектно-ориентированного программирования. Понятие структуры и объединения. Объявление и инициализация структур и объединений в языке Си. Связь структур и функций. Динамические структуры. Работа с битовыми полями структуры. 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. Перегрузка операций на объекты классов, операторные функции, ограничения на перегрузку в языке Си++. 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. 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. Понятие исключения, механизм обработки исключений в языке Си++. Обработчики исключений. Исключения при динамическом выделении памяти. 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. Разработка программ с графическим пользовательским интерфейсом под ОС Windows. Элементы графического интерфейса: окна, формы, диалоги, меню. Современные среды и технологии разработки приложений с графическим интерфейсом.</p>
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. Платформа .NET. Сравнение процесса компиляции в C, C# и Java. Промежуточный язык, среда выполнения. Сборки .NET в сравнении с библиотеками C/C++. 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. 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. 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.</p>

	<p>Ключевые слова <code>base</code> и <code>this</code>. Модификаторы доступа (<code>public</code>, <code>private</code>, <code>protected</code>, <code>internal</code>), инкапсуляция. Модификаторы <code>readonly</code> и <code>const</code> для переменных, отличия от C++.</p> <p>Наследование. Интерфейсы, их назначение и отличие от абстрактных классов. Правила множественного наследования. Преобразование ссылочных типов, операция проверки <code>is</code>. Переопределение методов. Использование ключевых слов <code>override</code> и <code>virtual</code>. Полиморфизм. Модификатор <code>sealed</code>.</p>
12	<p>Дополнительные возможности C#.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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#: <code>std::vector</code> и <code>List</code>. 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.</p> <p>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.</p> <p>Сравнение произвольных объектов. Компараторы.</p> <p>Двоичное дерево поиска (BST). Процедуры поиска, добавления и удаления элементов. Хранение деревьев в памяти.</p> <p>Понятие о сбалансированных деревьях. Красно-черное дерево и AVL-дерево: основные правила построения, принципы работы, примеры.</p> <p>Двоичная куча как частный случай двоичного дерева: хранение в памяти, правила построения, добавление и удаление элемента.</p> <p>Приоритетная очередь на основе дерева поиска или двоичной кучи. Случай повторяющихся элементов (приоритетов).</p>
18	<p>Множества и ассоциативные массивы.</p>

	<p>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# (std::set, std::map, SortedSet, SortedDictionary).</p> <p>Хранение данных без сортировки. Хэш-код объекта. Свойства и качество хеш-функции. Хэш-таблица. Проблема коллизий. Открытое хеширование (метод цепочек). Закрытое хеширование. Примеры работы.</p>
19	<p>Разработка многооконых приложений.</p> <p>Разделение модель-представление-контроллер (MVC). Передача данных между формами и компонентами. События обновления. Фоновые процессы.</p> <p>Работа с файловой системой, компонентами отображения форматированного текста, списками и таблицами.</p> <p>Обработка текстовых данных. Регулярные выражения.</p>
20	<p>Абстрактные вычислительные машины.</p> <p>Математическое понятие алгоритма. Формализация входных и выходных данных. Машина Тьюринга. Применимость программы к данным, функции, вычислимы по Тьюрингу. Неразрешимые по Тьюрингу задачи. Нормальные алгоритмы Маркова.</p>
21	<p>Принципы компиляции и анализа текста программ.</p> <p>Лексический, синтаксический и семантический анализ.</p> <p>Лексемы, синтаксическое дерево.</p> <p>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.