

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Сыров Игорь Анатольевич
Должность: Директор
Дата подписания: 30.10.2023 10:59:38
Уникальный программный ключ:
b683afe664d7e9f64175886cf9626a196149ad36

СТЕРЛИТАМАКСКИЙ ФИЛИАЛ
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»

Факультет Математики и информационных технологий
Кафедра Прикладной информатики и программирования

Аннотация рабочей программы дисциплины (модуля)

дисциплина ***Б1.О.19 Программирование***

обязательная часть

Направление

02.03.03 ***Математическое обеспечение и администрирование информационных систем***

код наименование направления

Программа

Сетевое программирование и администрирование информационных систем

Форма обучения

Очная

Для поступивших на обучение в
2023 г.

Стерлитамак 2023

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с установленными в образовательной программе индикаторами достижения компетенций

| Формируемая компетенция (с указанием кода) | Код и наименование индикатора достижения компетенции | Результаты обучения по дисциплине (модулю) |
|---|---|--|
| ОПК-3. Способен применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения | ОПК-3.1. Знает основные положения и концепции прикладного и системного программирования, архитектуры компьютеров и сетей (в том числе и глобальных), современные языки программирования, технологии создания и эксплуатации программных продуктов и программных комплексов. | Обучающийся должен: Обучающийся должен знать: основные алгоритмические конструкции и основные операторы языков программирования C# и C++; основные структуры данных, способы их создания и обработки; основы структурного, модульного и визуального программирования; основные этапы разработки алгоритмов и программ; современные языки программирования и пакеты программ в области программирования; современные способы и средства работы с информацией для изучения основных конструкций современных языков программирования; общие принципы и навыки практического применения объектно-ориентированного программирования. |
| | ОПК-3.2. Умеет использовать их в профессиональной деятельности. | Обучающийся должен : уметь выбирать структуры данных, необходимые для решения поставленной задачи; составлять алгоритмы обработки данных; разрабатывать программы для ЭВМ на языках программирования C# и C++, проводить их отладку и тестирование; ориентироваться в постановках задач, при решении поставленных задач обоснованно строить алгоритмы, реализовывать их на языках программирования C# и C++. |

| | | |
|--|---|--|
| | ОПК-3.3. Имеет практические навыки разработки программного обеспечения. | Обучающийся должен: владеть методами использования в профессиональной деятельности языков программирования C# и C++; современными методами практического программирования конкретных задач в определенной языковой среде; умениями и навыками использования библиотек объектов (классов) для решения практических задач; современными методами и приемами эффективной обработки информации средствами языков программирования. |
|--|---|--|

2. Цели и место дисциплины (модуля) в структуре образовательной программы

Цели изучения дисциплины:

Дисциплина реализуется в рамках обязательной части. Для освоения дисциплины необходимы знания, полученные в рамках школьного курса информатики.

Дисциплина «Программирование» занимает важное место среди изучаемых дисциплин. В процессе работы студенты должны на основе изученных тем и рассмотренных примеров приобрести практические навыки и умения в конструировании программ для ЭВМ, предназначенных для решения различных прикладных задач, в том числе связанных с информационными процессами и системами.

Дисциплина изучается на 1, 2 курсах в 1, 2, 3, 4 семестрах

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических или астрономических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость (объем) дисциплины составляет 17 зач. ед., 612 акад. ч.

| Объем дисциплины | Всего часов |
|--|----------------------|
| | Очная форма обучения |
| Общая трудоемкость дисциплины | 612 |
| Учебных часов на контактную работу с преподавателем: | |
| лекций | 64 |
| практических (семинарских) | 96 |
| лабораторных | 64 |
| другие формы контактной работы (ФКР) | 5,8 |
| Учебных часов на контроль (включая часы подготовки): | 104,4 |

| | |
|---|-------|
| экзамен | |
| зачет | |
| курсовая работа | |
| Учебных часов на самостоятельную работу обучающихся (СР): | 277,8 |
| курсовая работа | |

| Формы контроля | Семестры |
|-----------------|----------|
| экзамен | 1, 2, 4 |
| зачет | 3 |
| курсовая работа | 4 |

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)

| № п/п | Наименование раздела / темы дисциплины | Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость (в часах) | | | |
|----------|---|---|-----------|-----------|-------------|
| | | Контактная работа с преподавателем | | | СР |
| | | Лек | Пр/Сем | Лаб | |
| 1 | Основные приёмы программирования на языке C++. Структуры данных. | 16 | 16 | 16 | 60 |
| 1.1 | Введение в язык C++. | 2 | 2 | 0 | 8 |
| 1.2 | Управляющие структуры | 2 | 4 | 2 | 8 |
| 1.3 | Массивы и указатели. | 2 | 2 | 2 | 8 |
| 1.4 | Строки в C++. | 2 | 2 | 4 | 8 |
| 1.5 | Функции как средство структуризации программы. | 2 | 2 | 4 | 8 |
| 1.6 | Динамические структуры. | 2 | 2 | 0 | 10 |
| 1.7 | Файлы. | 4 | 2 | 4 | 10 |
| 2 | Объектно-ориентированное программирование на C++. | 16 | 32 | 16 | 80 |
| 2.1 | Классы в C++. | 2 | 4 | 2 | 10 |
| 2.2 | Разработка структуры классов. | 4 | 4 | 4 | 14 |
| 2.3 | Дружественные функции и классы. | 2 | 6 | 2 | 16 |
| 2.4 | Иерархия классов. | 2 | 6 | 4 | 14 |
| 2.5 | Усложнение структуры классов. | 4 | 6 | 0 | 16 |
| 2.6 | Перегрузка операций. | 2 | 6 | 4 | 10 |
| 3 | Визуальное программирование на C# | 16 | 16 | 16 | 59,8 |
| 3.1 | Среда программирования Visual Studio. Элементы управления. Создание проекта. События и сообщения. | 4 | 2 | 0 | 0 |
| 3.2 | Основные визуальные и не визуальные компоненты. | 2 | 4 | 4 | 10 |
| 3.3 | Интерфейсные объекты: | 2 | 2 | 0 | 10 |

| | | | | | |
|----------|--|-----------|-----------|-----------|--------------|
| | управляющие элементы, окна и диалоги. | | | | |
| 3.4 | Составляющие пользовательского интерфейса. | 2 | 2 | 0 | 10 |
| 3.5 | Динамическое создание компонент. | 2 | 0 | 6 | 10 |
| 3.6 | Создание классов, работа со списками объектов. | 2 | 2 | 0 | 10 |
| 3.7 | Разработка приложений прикладного характера. | 2 | 4 | 6 | 9,8 |
| 4 | Система визуального программирования Delphi | 16 | 32 | 16 | 78 |
| 4.1 | Среда программирования Delphi. Элементы управления. Создание проекта. События и сообщения. | 2 | 4 | 0 | 10 |
| 4.2 | Основные визуальные компоненты (VCL). | 2 | 4 | 4 | 10 |
| 4.3 | Интерфейсные объекты: управляющие элементы, окна и диалоги. | 2 | 4 | 0 | 10 |
| 4.4 | Составляющие пользовательского интерфейса. | 2 | 4 | 0 | 10 |
| 4.5 | Графические примитивы. Классы TScreen и TGraphics. Метод Canvas. | 2 | 4 | 4 | 10 |
| 4.6 | Динамическое создание компонент. | 2 | 4 | 0 | 10 |
| 4.7 | Создание классов, работа со списками объектов. | 2 | 4 | 4 | 10 |
| 4.8 | Разработка приложений прикладного характера. | 2 | 4 | 4 | 8 |
| | Итого | 64 | 96 | 64 | 277,8 |

4.2. Содержание дисциплины, структурированное по разделам (темам)

Курс лекционных занятий

| № | Наименование раздела / темы дисциплины | Содержание |
|----------|---|--|
| 1 | Основные приёмы программирования на языке C++. Структуры данных. | |
| 1.1 | Введение в язык C++. | Алфавит, синтаксис, семантика языка. Структура программы. Простейшие операторы ввода-вывода. Порядковые типы. Целые, логические, символьные типы, их физическое представление. Другие типы. Вещественные типы, их физическое представление. Описание констант и переменных стандартных типов. Приведение типов. Выражения. |
| 1.2 | Управляющие структуры | Полная и неполная форма ветвления. Множественное ветвление. Оператор безусловного перехода goto. Циклы с предусловием и с постусловием. Оператор цикла с параметром. Операторы break и continue. Организация цикла с помощью рекурсивных процедур. |
| 1.3 | Массивы и указатели. | Адреса и указатели. Объявление указателей. Операции над указателями. Регулярные типы (одномерные и многомерные массивы), комбинированные типы их |

| | | |
|----------|---|--|
| | | физическое представление. |
| 1.4 | Строки в C++. | Физическое представление строковых величин, операции со строками. Тип struct в языке программирования C++. |
| 1.5 | Функции как средство структуризации программы. | Описание функции и ее вызов. Локальные переменные. Способы передачи параметров функций (по значению и по адресу). Фактические и формальные параметры. Ссылки и ссылочные параметры. Функции с переменным количеством параметров. Прототипы функций. Перегрузка функций. Шаблоны функций. |
| 1.6 | Динамические структуры. | Конструирование объектов: динамические массивы, стеки, списки, очереди. |
| 1.7 | Файлы. | Операции открытия, чтения, записи, закрытия файла. Отличие файлов прямого и последовательного доступа. Генерирование имен временных файлов. |
| 2 | Объектно-ориентированное программирование на C++. | |
| 2.1 | Классы в C++. | Основные понятия. Классы. Конструкторы и деструкторы. Область действия и доступ к членам класса. |
| 2.2 | Разработка структуры классов. | Статические поля. Конструктор копирования. Указатель this. Передача значения по ссылке. |
| 2.3 | Дружественные функции и классы. | Организация взаимодействия нескольких объектов разных классов посредством дружественных функций. Взаимодействие классов на примере описания структуры «Клиент – Банк». |
| 2.4 | Иерархия классов. | Инкапсуляция. Вложенные классы в C++. Наследование и полиморфизм. Множественное наследование. Виртуальные методы. |
| 2.5 | Усложнение структуры классов. | Абстрактные классы. Виртуальные базовые классы. Шаблоны классов. Нетипизированные аргументы. Частичная специализация шаблона. |
| 2.6 | Перегрузка операций. | Механизм перегрузки операций. Перегрузка унарных и бинарных операций. Перегрузка операции вызова функции и операции индексирования. Перегрузка операций «поместить в поток» и «взять из потока». |
| 3 | Визуальное программирование на C# | |
| 3.1 | Среда программирования Visual Studio. Элементы управления. Создание проекта. События и сообщения. | Меню и панели инструментов среды разработки, панели компонентов. Элементы управления. Простейшее приложение. Понятие события и сообщения. |
| 3.2 | Основные визуальные и не визуальные компоненты. | Компоненты панели элементов: Стандартные элементы управления, Контейнеры, Меню и панели инструментов, Компоненты, Диалоговые окна. |
| 3.3 | Интерфейсные объекты: управляющие элементы, окна и диалоги. | Интерфейсные объекты. Типы окон. Диалоговые окна. Взаимодействие отдельных окон приложения. Основное и контекстное меню приложения. |
| 3.4 | Составляющие пользовательского интерфейса. | Составляющие пользовательского интерфейса: классы Application, Form, Clipboard, InstalledFormCollection. |
| 3.5 | Динамическое создание | Разработка приложений с использованием элементов |

| | | |
|----------|--|---|
| | компонент. | управления. Динамическое создание элементов управления и других компонент, используемых в приложении. |
| 3.6 | Создание классов, работа со списками объектов. | Универсальный контейнер списков: класс List. Создание собственных классов, организация групп объектов на основе List. |
| 3.7 | Разработка приложений прикладного характера. | Технология разработки приложений прикладного характера. Создание главного меню и панели инструментов приложения. Реализация возможностей настройки параметров приложения. |
| 4 | Система визуального программирования Delphi | |
| 4.1 | Среда программирования Delphi. Элементы управления. Создание проекта. События и сообщения. | Меню и панели инструментов среды разработки, панели компонентов. Элементы управления. Простейшее приложение. Понятие события и сообщения. |
| 4.2 | Основные визуальные компоненты (VCL). | Компоненты панелей Standard, Additional, Win32, System, Dialogs. Библиотеки объектов. |
| 4.3 | Интерфейсные объекты: управляющие элементы, окна и диалоги. | Интерфейсные объекты. Типы окон. Диалоговые окна. Взаимодействие отдельных окон приложения. Основное и контекстное меню приложения. |
| 4.4 | Составляющие пользовательского интерфейса. | Составляющие пользовательского интерфейса: приложение (класс TApplication), класс TForm, класс «Буфер обмена» (TClipboard), виртуальный экран в Delphi (класс TScreen). |
| 4.5 | Графические примитивы. Классы TScreen и TGraphics. Метод Canvas. | Графические примитивы. Классы TScreen и TGraphics. Метод Canvas на примере компонента TImage в среде программирования Delphi. Использование функций WIN API. |
| 4.6 | Динамическое создание компонент. | Разработка приложений с использованием элементов управления. Динамическое создание элементов управления и других компонент, используемых в приложении. |
| 4.7 | Создание классов, работа со списками объектов. | Универсальный контейнер списков: класс TList. Создание собственных классов, организация групп объектов на основе TList. |
| 4.8 | Разработка приложений прикладного характера. | Технология разработки приложений прикладного характера. Создание главного меню и панели инструментов приложения. Реализация возможностей настройки параметров приложения. |

Курс практических/семинарских занятий

| № | Наименование раздела / темы дисциплины | Содержание |
|----------|---|--|
| 1 | Основные приёмы программирования на языке C++. Структуры данных. | |
| 1.1 | Введение в язык C++. | Структура программы. Описание переменных. Инструкции ввода-вывода. Использование ветвления при решении задач, содержащих разнообразные условия. Программирование составных условий. Инструкция switch. |
| 1.2 | Управляющие структуры | Изучение циклов с параметром, с предусловием и с после условием. Применения каждого вида циклов в |

| | | |
|----------|--|--|
| | | зависимости от требований конкретной задачи. Вложенные циклы. Инструкции break и continue. |
| 1.3 | Массивы и указатели. | Указатели. Операции над указателями. Одномерные и многомерные массивы. Использование указателей при работе с массивами. |
| 1.4 | Строки в C++. | Необходимость введения составного типа данных. Составление сложных структур данных на основе типа struct. Примеры использования типа struct. |
| 1.5 | Функции как средство структуризации программы. | Использование функций при написании программ. Передача массивов в функции. Создание меню программы на основе функций. Эффективное использование функций. Параметры по умолчанию и перегрузка функций. Прототипы. |
| 1.6 | Динамические структуры. | Особенности работы с символами и строками в C++. Классовый тип string. Организация ввода-вывода между программой и объектом string. |
| 1.7 | Файлы. | Работа с файлами: использование средств библиотек stdio и fstream. Файлы произвольного и последовательного доступа. Выбор типа файла для хранения данных в зависимости от требований задачи. |
| 2 | Объектно-ориентированное программирование на C++. | |
| 2.1 | Классы в C++. | Классы для описания простых математических и геометрических объектов. Инкапсуляция. Примеры разработки классов для различных областей использования. |
| 2.2 | Разработка структуры классов. | Организация взаимодействия нескольких объектов разных классов посредством дружественных функций. Взаимодействие классов на отдельных примерах. |
| 2.3 | Дружественные функции и классы. | Механизм наследования. Множественное наследование. Конструкторы и деструкторы базовых и производных классов. Раннее и позднее связывание, виртуальные функции. |
| 2.4 | Иерархия классов. | Назначение и использование абстрактных классов. Абстрактные классы как основа более сложных структурных объектов. Невозможность создания экземпляров объектов абстрактных классов. Чисто виртуальные функции. |
| 2.5 | Усложнение структуры классов. | Механизм перегрузки операций. Перегрузка унарных и бинарных операций. Перегрузка операции вызова функции и операции индексирования. Перегрузка операций «поместить в поток» и «взять из потока». |
| 2.6 | Перегрузка операций. | Практическое применение и обобщение приемов и методов разработки классов для описания математических и других (физических, социальных, и т.д.) объектов. |
| 3 | Визуальное программирование на C# | |
| 3.1 | Среда программирования Visual Studio. Элементы управления. Создание проекта. | Работа с основными компонентами панели элементов. Создание простых приложений. Разработка приложения «Калькулятор». |

| | | |
|----------|--|--|
| | События и сообщения. | |
| 3.2 | Основные визуальные и не визуальные компоненты. | Разработка интерфейса текстового редактора, содержащего верхнее меню и панель инструментов. Возможности редактора: настройка параметров шрифта, сохранение и открытие текстовых документов. |
| 3.3 | Интерфейсные объекты: управляющие элементы, окна и диалоги. | Обработка событий и исключительных ситуаций. Стандартные диалоги. Усовершенствование калькулятора и текстового редактора. |
| 3.4 | Составляющие пользовательского интерфейса. | Разработка приложения, содержащего динамически создаваемые компоненты (компоненты, создаваемые в процессе работы программы). Объяснение необходимости динамического создания компонент. Пример приложения. |
| 3.6 | Создание классов, работа со списками объектов. | Создание собственных классов и объектов. Практическое использование классов на примере классов «Линия», «Прямоугольник», «Эллипс». |
| 3.7 | Разработка приложений прикладного характера. | Практика разработки приложений прикладного характера |
| 4 | Система визуального программирования Delphi | |
| 4.1 | Среда программирования Delphi. Элементы управления. Создание проекта. События и сообщения. | Меню и панели инструментов среды разработки, панели компонентов. Элементы управления. Простейшее приложение. Понятие события и сообщения. |
| 4.2 | Основные визуальные компоненты (VCL). | Компоненты панели элементов: Стандартные элементы управления, Контейнеры, Меню и панели инструментов, Компоненты, Диалоговые окна. |
| 4.3 | Интерфейсные объекты: управляющие элементы, окна и диалоги. | Интерфейсные объекты. Типы окон. Диалоговые окна. Взаимодействие отдельных окон приложения. Основное и контекстное меню приложения. |
| 4.4 | Составляющие пользовательского интерфейса. | Составляющие пользовательского интерфейса: классы Application, Form, Clipboard, InstalledFormCollection. |
| 4.5 | Графические примитивы. Классы TScreen и TGraphics. Метод Canvas. | Разработка приложений с использованием элементов управления. Динамическое создание элементов управления и других компонент, используемых в приложении. |
| 4.6 | Динамическое создание компонент. | Универсальный контейнер списков: класс List. Создание собственных классов, организация групп объектов на основе List. |
| 4.7 | Создание классов, работа со списками объектов. | Практики разработки приложений прикладного характера. |
| 4.8 | Разработка приложений прикладного характера. | Технология разработки приложений прикладного характера. Создание главного меню и панели инструментов приложения. Реализация возможностей настройки параметров приложения. |

Курс лабораторных занятий

| № | Наименование раздела / темы дисциплины | Содержание |
|---|---|------------|
| 1 | Основные приёмы программирования на языке C++. Структуры данных. | |

| | | |
|----------|--|---|
| 1.2 | Управляющие структуры | Лабораторная работа №1. Программирование ветвлений и циклов. Практическое применение инструкций ветвления и циклов при решении задач по программированию. Совершенствование навыков разработки программ. |
| 1.3 | Массивы и указатели. | Лабораторная работа №2. Работа со статическими и динамическими массивами. Закрепление практических навыков работы с массивами. Обработка совокупностей однотипных данных (ввод, сортировка, преобразование, и т.д.), отработка умений выбирать приемлемые алгоритмы работы с потоками данных. |
| 1.4 | Строки в C++. | Лабораторная работа №3. Работа со строками. Обработка данных строкового типа. Преобразование строковых и числовых значений. Поиск данных, удовлетворяющих некоторым условиям. |
| 1.5 | Функции как средство структуризации программы. | Лабораторная работа №4. Функции. Структуры (struct) в C++. Работа со сложными (составными) структурами данных. Ввод, преобразование и хранение таких данных. |
| 1.7 | Файлы. | Лабораторная работа №5. Файлы в C++. Хранение данных в виде файлов. Приемы работы с библиотеками файлов. Создание простой базы данных. |
| 2 | Объектно-ориентированное программирование на C++. | |
| 2.1 | Классы в C++. | Лабораторная работа №6. Объекты и классы. Введение. Разработка классов для описания математических и геометрических объектов. Функции-утилиты. Конструкторы с параметрами по умолчанию. |
| 2.2 | Разработка структуры классов. | Лабораторная работа №7. Производные классы. Наследование. Создание и усложнение классов на основе ранее описанных более простых классов. Практическое применение механизма наследования при описании объектов более сложной структуры. |
| 2.3 | Дружественные функции и классы. | Лабораторная работа №8. Полиморфизм. Виртуальные функции. Полиморфное поведение функций при наследовании. Раннее и позднее связывание. Необходимость описания функции как виртуальной. |
| 2.4 | Иерархия классов. | Лабораторная работа №9. Виртуальные функции. Абстрактные классы. Практика описания и использования абстрактных классов и чисто виртуальных функций. |
| 2.6 | Перегрузка операций. | Лабораторная работа №10. Перегрузка операций. Практическое использование механизма перегрузки операций при работе с экземплярами объектов различных классов. |
| 3 | Визуальное программирование на C# | |
| 3.2 | Основные визуальные и не визуальные компоненты. | Лабораторная работа №12. Динамическое создание компонент. Динамическое создание компонент при проектировании приложений. Создание и использование компонент, настройка событий и свойств. Удаление компонентов. |
| 3.5 | Динамическое создание компонент. | Лабораторная работа №13. Разработка приложений прикладного характера. |

| | | |
|----------|--|---|
| | | Использование компонент панели элементов при разработке приложений. Разработка готового приложения, предназначенного для выполнения конкретной задачи (в рамках учебного проекта согласно предложенным вариантам заданий). |
| 3.7 | Разработка приложений прикладного характера. | Лабораторная работа №13. Разработка приложений прикладного характера. Использование компонент панели элементов при разработке приложений. Разработка готового приложения, предназначенного для выполнения конкретной задачи (в рамках учебного проекта согласно предложенным вариантам заданий). |
| 4 | Система визуального программирования Delphi | |
| 4.2 | Основные визуальные компоненты (VCL). | Лабораторная работа №14. Использование компонент Delphi при разработке приложений. Использование компонент панелей Standard, Additional и System для решений некоторых математических задач. Создание интерфейса простого приложения. |
| 4.5 | Графические примитивы. Классы TScreen и TGraphics. Метод Canvas. | Лабораторная работа №15. Графика в проектах Delphi. Метод Canvas для компонента TImage. Использование стилей и отображений линий в процессе разработки приложения. Приемы и методы работы с графикой. |
| 4.7 | Создание классов, работа со списками объектов. | Лабораторная работа №16. Классы и списки. Разработка собственных классов для решения практических задач. Работа со списками объектов. Добавление, изменение и уничтожение объектов. |
| 4.8 | Разработка приложений прикладного характера. | Лабораторная работа №17. Разработка приложений прикладного характера. Использование компонентов панелей Standard, Additional, Win32, System, Dialogs и других при разработке приложений. Разработка готового приложения, предназначенного для выполнения конкретной задачи (в рамках учебного проекта согласно предложенным вариантам заданий). |