

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
"Казанский (Приволжский) федеральный университет"
Институт математики и механики им. Н.И. Лобачевского



УТВЕРЖДАЮ

Проректор
по образовательной деятельности КФУ
Проф. Таюрский Д.А.

_____ 20__ г.

Программа дисциплины

Разработка приложений в языке С# Б1.В.ДВ.3

Направление подготовки: 02.03.01 - Математика и компьютерные науки

Профиль подготовки: Математическое и компьютерное моделирование

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Автор(ы):

Липачев Е.К. , Насрутдинов М.Ф.

Рецензент(ы):

Маклецов С.В.

СОГЛАСОВАНО:

Заведующий(ая) кафедрой: Тронин С. Н.

Протокол заседания кафедры No ____ от " ____ " _____ 201__ г

Учебно-методическая комиссия Института математики и механики им. Н.И. Лобачевского :

Протокол заседания УМК No ____ от " ____ " _____ 201__ г

Регистрационный No

Казань
2019

Содержание

1. Цели освоения дисциплины
2. Место дисциплины в структуре основной образовательной программы
3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля
4. Структура и содержание дисциплины/ модуля
5. Образовательные технологии, включая интерактивные формы обучения
6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов
7. Литература
8. Интернет-ресурсы
9. Материально-техническое обеспечение дисциплины/модуля согласно утвержденному учебному плану

Программу дисциплины разработал(а)(и) доцент, к.н. (доцент) Липачев Е.К. кафедра компьютерной математики и информатики отделение педагогического образования , elipachev@gmail.com ; заместитель директора по образовательной деятельности Насрутдинов М.Ф. Высшая школа информационных технологий и интеллектуальных систем КФУ , Marat.Nasrutdinov@kpfu.ru

1. Цели освоения дисциплины

Изучение современных технологий программирования

2. Место дисциплины в структуре основной образовательной программы высшего профессионального образования

Данная учебная дисциплина включена в раздел "Б1.В.ДВ.3 Дисциплины (модули)" основной образовательной программы 02.03.01 Математика и компьютерные науки и относится к дисциплинам по выбору. Осваивается на 4 курсе, 7 семестр.

Дисциплина входит в программу подготовки бакалавров по специальности Математика и компьютерные науки (Математическое и компьютерное моделирование) Института Математики и Механики КФУ

3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля

В результате освоения дисциплины формируются следующие компетенции:

Шифр компетенции	Расшифровка приобретаемой компетенции
опк-4	способностью находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем

В результате освоения дисциплины студент:

1. должен знать:

синтаксис и основные конструкции языка C#, знать принципы технологии .NET, навыки разработки программ на языке C#

2. должен уметь:

работать в среде разработки MS Visual Studio, создавать и отлаживать проекты. Владеть приемами программирования в среде MS Visual Studio.

3. должен владеть:

приемами программирования в среде MS Visual Studio

4. должен демонстрировать способность и готовность:

Записывать алгоритмы в виде программ на языке C#, выполнить отладку и создание приложений в среде MS Visual Studio.

4. Структура и содержание дисциплины/ модуля

Общая трудоемкость дисциплины составляет 2 зачетных(ые) единиц(ы) 72 часа(ов).

Форма промежуточного контроля дисциплины: зачет в 7 семестре.

Суммарно по дисциплине можно получить 100 баллов, из них текущая работа оценивается в 50 баллов, итоговая форма контроля - в 50 баллов. Минимальное количество для допуска к зачету 28 баллов.

- 86 баллов и более - "отлично" (отл.);
 71-85 баллов - "хорошо" (хор.);
 55-70 баллов - "удовлетворительно" (удов.);
 54 балла и менее - "неудовлетворительно" (неуд.).

4.1 Структура и содержание аудиторной работы по дисциплине/ модулю

Тематический план дисциплины/модуля

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. Обзор платформы Microsoft.Net	7		1	1	0	
2.	Тема 2. Обзор языка программирования С#	7		1	1	0	
3.	Тема 3. Система типов языка С#	7		2	2	0	
4.	Тема 4. Операторы в С#	7		2	2	0	
5.	Тема 5. Инструкции управления	7		1	1	0	
6.	Тема 6. Массивы и строки	7		2	2	0	
7.	Тема 7. Принципы объектно-ориентированного программирования			2	2	0	
8.	Тема 8. Использование ссылочных типов. Подробнее о методах и классах	7		2	2	0	
9.	Тема 9. Перегрузка операторов	7		1	1	0	
10.	Тема 10. Наследование в С#	7		2	2	0	
11.	Тема 11. Ввод и вывод в С#	7		2	2	0	
	Тема . Итоговая форма контроля	7		0	0	0	Зачет
	Итого			18	18	0	

4.2 Содержание дисциплины

Тема 1. Обзор платформы Microsoft.Net

лекционное занятие (1 часа(ов)):

Обзор платформы Microsoft.Net. Основные понятия. .NET Framework - инфраструктура платформы Microsoft.NET. CLS (Common Language Specification) - общая спецификация языков программирования. CLS как это основа межъязыкового взаимодействия в рамках платформы Microsoft.NET. CLR (Common Language Runtime) ? основной компонент NET Framework - Среда Времени Выполнения или Виртуальная Машина, обеспечивающая выполнение сборки. Управляемый код - программный код, который при своем выполнении способен использовать службы, предоставляемые CLR. FCL (.NET Framework Class Library) соответствующая CLS- спецификации объектно-ориентированная библиотека классов, интерфейсов и системы типов (типов-значений), которые включаются в состав платформы Microsoft.NET. .NET-приложение - приложение, разработанное для выполнения на платформе Microsoft.NET Реализуется на языках программирования, соответствующих CLS. MSIL (Microsoft Intermediate Language) - промежуточный язык платформы Microsoft.NET.

практическое занятие (1 часа(ов)):

NET Framework - инфраструктура платформы Microsoft.NET. CLS (Common Language Specification) - общая спецификация языков программирования.

Тема 2. Обзор языка программирования С#

лекционное занятие (1 часа(ов)):

Синтаксис языка программирования С#. Типы языка программирования С#. Объявление констант и переменных. Классы и объекты. Составляющие класса. Класс object. Пространства имен. Важные пространства имен. Доступ к пространствам имен. Типы приложений. Создание консольного приложения. Базовое приложение на языке программирования С# с вводом и выводом на консоль.

практическое занятие (1 часа(ов)):

Создание консольного приложения. Базовое приложение на языке программирования С# с вводом и выводом на консоль.

Тема 3. Система типов языка С#

лекционное занятие (2 часа(ов)):

Система типов: типы значений (типы ?значения), ссылочные типы (типы-ссылки), параметризованные типы (типы-шаблоны). Простые (элементарные) типы - это типы, имя и основные свойства которых известны компилятору. Sbyte, Byte, Short, Ushort, Int, uint, long, ulong, char, float, double, decimal, bool. Среди множества массов выделяют предопределенные ссылочные типы object и string, которым соответствуют FCL-типы System.Object и System.String. Литералы. Шестнадцатеричные литералы. Управляющие последовательности символов. Строковые литералы. Преобразование и приведение типов. Автоматическое преобразование типов. Приведение несовместимых типов. Преобразование типов в выражениях. Приведение типов в выражениях.

практическое занятие (2 часа(ов)):

Преобразование и приведение типов. Автоматическое преобразование типов. Приведение несовместимых типов. Преобразование типов в выражениях. Приведение типов в выражениях.

Тема 4. Операторы в С#

лекционное занятие (2 часа(ов)):

Арифметические операторы. Инкремент и декремент. Операторы отношений и логические операторы. Сокращенные логические операторы. Оператор присваивания. Составные операторы присваивания. Поразрядные операторы. Поразрядные операторы И, ИЛИ, исключающее ИЛИ и НЕ. Операторы сдвига. Поразрядные составные операторы присваивания. Оператор ?. Использование пробелов и круглых скобок. Приоритет операторов.

практическое занятие (2 часа(ов)):

Составные операторы присваивания. Поразрядные операторы. Поразрядные операторы И, ИЛИ, исключающее ИЛИ и НЕ.

Тема 5. Инструкции управления

лекционное занятие (1 часа(ов)):

Инструкция if. Вложенные if-инструкции. Конструкция if-else-if. Инструкция switch. Вложенные инструкции switch. Цикл for. Вариации на тему цикла for. Использование нескольких управляющих переменных цикла. Условное выражение. Отсутствие элементов в определении цикла. Бесконечный цикл. Циклы без тела. Объявление управляющей переменной в цикле for. Цикл while. Цикл do-while. Цикл foreach. Использование инструкции break для выхода из цикла. Использование инструкции continue. Инструкция return. Инструкция goto.

практическое занятие (1 часа(ов)):

Цикл while. Цикл do-while. Цикл foreach. Использование инструкции break для выхода из цикла. Использование инструкции continue.

Тема 6. Массивы и строки

лекционное занятие (2 часа(ов)):

Массивы. Одномерные массивы. Инициализация массива. Соблюдение "пограничного режима". Многомерные массивы. Двумерные массивы. Массивы трех и более измерений. Инициализация многомерных массивов. Рваные массивы. Присвоение значений ссылочным переменным массивов. Использование свойства Length. Использование свойства Length при работе с рваными массивами. Цикл foreach. Строки. Создание строк. Работа со строками. Массивы строк. Постоянство строк. Использование строк в switch-инструкциях.

практическое занятие (2 часа(ов)):

Многомерные массивы. Двумерные массивы. Массивы трех и более измерений. Инициализация многомерных массивов. Рваные массивы.

Тема 7. Принципы объектно-ориентированного программирования

лекционное занятие (2 часа(ов)):

Общая форма определения класса. Определение класса. Создание объектов. Переменные ссылочного типа и присвоение им значений. Методы. Добавление методов в класс Building. Возвращение из метода. Возврат значения. Использование параметров. Добавление параметризованного метода в класс Building. Как избежать написания недостижимого кода. Конструкторы. Параметризованные конструкторы. Добавление конструктора в класс Building. Использование оператора new. Применение оператора new к переменным типа значений. Сбор "мусора" и использование деструкторов. Деструкторы. Ключевое слово this.

практическое занятие (2 часа(ов)):

Конструкторы. Параметризованные конструкторы. Добавление конструктора в класс Building. Использование оператора new.

Тема 8. Использование ссылочных типов. Подробнее о методах и классах

лекционное занятие (2 часа(ов)):

Управление доступом к членам класса. Спецификаторы доступа C#. Применение спецификаторов доступа public и private. Передача объектов методам. Как происходит передача аргументов. Использование ref- и out-параметров. Использование модификатора ref. Использование модификатора out. Использование модификаторов ref и out для ссылочных параметров. Использование переменного количества аргументов. Возвращение методами объектов. Возвращение методами массивов. Перегрузка методов. Перегрузка конструкторов. Вызов перегруженного конструктора с помощью ссылки this. Метод Main(). Возвращение значений из метода Main(). Передача аргументов методу Main(). Рекурсия. Использование модификатора типа static. Статические конструкторы.

практическое занятие (2 часа(ов)):

Передача объектов методам. Как происходит передача аргументов. Использование ref- и out-параметров. Использование модификатора ref. Использование модификатора out. Использование модификаторов ref и out для ссылочных параметров.

Тема 9. Перегрузка операторов

лекционное занятие (1 часа(ов)):

Основы перегрузки операторов. Перегрузка бинарных операторов. Перегрузка унарных операторов. Выполнение операций над значениями встроенных C#-типов. Перегрузка операторов отношений. Перегрузка операторов true и false. Перегрузка логических операторов. Простой случай перегрузки логических операторов. Включение операторов, действующих по сокращенной схеме вычислений. Операторы преобразования. Рекомендации и ограничения по созданию перегруженных операторов.

практическое занятие (1 часа(ов)):

Включение операторов, действующих по сокращенной схеме вычислений. Операторы преобразования. Рекомендации и ограничения по созданию перегруженных операторов.

Тема 10. Наследование в C#

лекционное занятие (2 часа(ов)):

Основы наследования. Доступ к членам класса и наследование. Использование защищенного доступа. Конструкторы и наследование. Вызов конструкторов базового класса. Наследование и сокрытие имен. Использование ключевого слова base для доступа к скрытому имени. Создание многоуровневой иерархии. Последовательность вызова конструкторов. Ссылки на базовый класс и объекты производных классов. Виртуальные методы и их переопределение. Зачем переопределять методы. Применение виртуальных методов. Использование абстрактных классов. Использование ключевого слова sealed для предотвращения наследования. Класс object. Приведение к объектному типу и восстановление значения. Использование класса object в качестве обобщенного типа данных.

практическое занятие (2 часа(ов)):

Применение виртуальных методов. Использование абстрактных классов. Использование ключевого слова sealed для предотвращения наследования. Класс object. Приведение к объектному типу и восстановление значения. Использование класса object в качестве обобщенного типа данных.

Тема 11. Ввод и вывод в C#

лекционное занятие (2 часа(ов)):

Организация C#-системы ввода-вывода. Байтовые и символьные потоки. Встроенные потоки. Классы потоков. Класс Stream. Байтовые классы потоков. Символьные классы потоков. Двоичные потоки. Консольный ввод-вывод данных. Считывание данных из консольного входного потока. Запись данных в консольный входный поток. Класс FileStream и файловый ввод-вывод на побайтовой основе. Как открыть и закрыть файл. Считывание байтов из объекта класса FileStream. Запись данных в файл. Использование класса FileStream для копирования файла. Файловый ввод-вывод с ориентацией на символы. Использование класса StreamWriter. Использование класса StreamReader. Перенаправление стандартных потоков. Считывание и запись двоичных данных. Класс BinaryWriter. Класс BinaryReader.

практическое занятие (2 часа(ов)):

Консольный ввод-вывод данных. Считывание данных из консольного входного потока. Запись данных в консольный входный поток. Класс FileStream и файловый ввод-вывод на побайтовой основе. Как открыть и закрыть файл.

4.3 Структура и содержание самостоятельной работы дисциплины (модуля)

N	Раздел Дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
1.	Тема 1. Обзор платформы Microsoft.Net	7		Изучение платформы Visual Studio	2	Проверка практических навыков
2.	Тема 2. Обзор языка программирования C#	7		Написание простых программ	2	Проверка практических навыков

N	Раздел Дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
3.	Тема 3. Система типов языка C#	7		Написание кода с использованием различных типов	2	Проверка практических навыков
4.	Тема 4. Операторы в C#	7		Написание программ	2	Проверка практических навыков
5.	Тема 5. Инструкции управления	7		Написание компьютерных программ	2	Проверка практических навыков
6.	Тема 6. Массивы и строки	7		Написание программ с использованием массивов и строк	6	Проверка практических навыков
7.	Тема 7. Принципы объектно-ориентированного программирования			Написание компьютерных программ с использованием классов	2	Проверка практических навыков
8.	Тема 8. Использование ссылочных типов. Подробнее о методах и классах	7		Написание компьютерных программ с использованием ссылочных типов	4	Проверка практических навыков
9.	Тема 9. Перегрузка операторов	7		Написание программ	2	Проверка практических навыков
10.	Тема 10. Наследование в C#	7		Создание программ с использованием виртуальных методов	6	Проверка практических навыков
11.	Тема 11. Ввод и вывод в C#	7		Написание компьютерных программ	6	Проверка практических навыков
	Итого				36	

5. Образовательные технологии, включая интерактивные формы обучения

Занятия проводятся в компьютерном классе. Обучаемые выполняют подготовку программ, используя справочную службу среды разработки, а также сетевые ресурсы, включая Интернет

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

Тема 1. Обзор платформы Microsoft.Net

Проверка практических навыков, примерные вопросы:

CLS (Common Language Specification) - общая спецификация языков программирования. CLS как это основа межъязыкового взаимодействия в рамках платформы Microsoft.NET. CLR (Common Language Runtime) ? основной компонент NET Framework - Среда Времени Выполнения или Виртуальная Машина, обеспечивающая выполнение сборки.

Тема 2. Обзор языка программирования C#

Проверка практических навыков, примерные вопросы:

Синтаксис языка программирования C#. Типы языка программирования C#. Объявление констант и переменных. Классы и объекты. Составляющие класса. Класс object. Пространства имен. Важные пространства имен. Доступ к пространствам имен. Типы приложений. Создание консольного приложения.

Тема 3. Система типов языка C#

Проверка практических навыков, примерные вопросы:

типы значений (типы ?значения), ссылочные типы (типы-ссылки), параметризованные типы (типы-шаблоны). Простые (элементарные) типы - это типы, имя и основные свойства которых известны компилятору. Sbyte, Byte, Short, Ushort, Int, uint, long, ulong, char, float, double, decimal, bool.

Тема 4. Операторы в C#

Проверка практических навыков, примерные вопросы:

Арифметические операторы. Инкремент и декремент. Операторы отношений и логические операторы. Сокращенные логические операторы. Оператор присваивания. Составные операторы присваивания. Поразрядные операторы. Поразрядные операторы И, ИЛИ, исключающее ИЛИ и НЕ. Операторы сдвига. Поразрядные составные операторы присваивания. Оператор ?. Использование пробелов и круглых скобок. Приоритет операторов.

Тема 5. Инструкции управления

Проверка практических навыков, примерные вопросы:

Инструкция if. Вложенные if-инструкции. Конструкция if-else-if. Инструкция switch. Вложенные инструкции switch. Цикл for. Вариации на тему цикла for. Использование нескольких управляющих переменных цикла. Условное выражение. Отсутствие элементов в определении цикла. Бесконечный цикл. Циклы без тела. Объявление управляющей переменной в цикле for. Цикл while. Цикл do-while. Цикл foreach. Использование инструкции break для выхода из цикла. Использование инструкции continue. Инструкция return. Инструкция goto.

Тема 6. Массивы и строки

Проверка практических навыков, примерные вопросы:

Массивы. Одномерные массивы. Инициализация массива. Соблюдение "пограничного режима". Многомерные массивы. Двумерные массивы. Массивы трех и более измерений. Инициализация многомерных массивов. Рваные массивы. Присвоение значений ссылочным переменным массивов. Использование свойства Length. Использование свойства Length при работе с рваными массивами. Цикл foreach. Строки. Создание строк. Работа со строками. Массивы строк. Постоянство строк. Использование строк в switch-инструкциях.

Тема 7. Принципы объектно-ориентированного программирования

Проверка практических навыков, примерные вопросы:

Определение класса. Создание объектов. Переменные ссылочного типа и присвоение им значений. Методы. Добавление методов в класс Building. Возвращение из метода. Возврат значения. Использование параметров. Добавление параметризованного метода в класс Building. Как избежать написания недостижимого кода. Конструкторы. Параметризованные конструкторы. Добавление конструктора в класс Building. Использование оператора new. Применение оператора new к переменным типа значений. Сбор "мусора" и использование деструкторов. Деструкторы. Ключевое слово this.

Тема 8. Использование ссылочных типов. Подробнее о методах и классах

Проверка практических навыков, примерные вопросы:

Спецификаторы доступа C#. Применение спецификаторов доступа public и private. Передача объектов методам. Как происходит передача аргументов. Использование ref- и out-параметров. Использование модификатора ref. Использование модификатора out. Использование модификаторов ref и out для ссылочных параметров. Использование переменного количества аргументов. Возвращение методами объектов. Возвращение методами массивов. Перегрузка методов. Перегрузка конструкторов. Вызов перегруженного конструктора с помощью ссылки this.

Тема 9. Перегрузка операторов

Проверка практических навыков, примерные вопросы:

Перегрузка бинарных операторов. Перегрузка унарных операторов. Выполнение операций над значениями встроенных C#-типов. Перегрузка операторов отношений. Перегрузка операторов true и false. Перегрузка логических операторов. Простой случай перегрузки логических операторов. Включение операторов, действующих по сокращенной схеме вычислений. Операторы преобразования.

Тема 10. Наследование в C#

Проверка практических навыков, примерные вопросы:

Доступ к членам класса и наследование. Использование защищенного доступа. Конструкторы и наследование. Вызов конструкторов базового класса. Наследование и сокрытие имен. Использование ключевого слова base для доступа к скрытому имени. Создание многоуровневой иерархии. Последовательность вызова конструкторов. Ссылки на базовый класс и объекты производных классов. Виртуальные методы и их переопределение. Зачем переопределять методы. Применение виртуальных методов. Использование абстрактных классов.

Тема 11. Ввод и вывод в C#

Проверка практических навыков, примерные вопросы:

Классы потоков. Класс Stream. Байтовые классы потоков. Символьные классы потоков. Двоичные потоки. Консольный ввод-вывод данных. Считывание данных из консольного входного потока. Запись данных в консольный входный поток. Класс FileStream и файловый ввод-вывод на побайтовой основе. Как открыть и закрыть файл. Считывание байтов из объекта класса FileStream. Запись данных в файл. Использование класса FileStream для копирования файла. Файловый ввод-вывод с ориентацией на символы. Использование класса StreamWriter. Использование класса StreamReader. Перенаправление стандартных потоков.

Итоговая форма контроля

зачет (в 7 семестре)

Примерные вопросы к зачету:

1. .NET Framework - инфраструктура платформы Microsoft.NET. CLS как это основа межъязыкового взаимодействия в рамках платформы Microsoft.NET. CLR (Common Language Runtime) - Среда Времени Выполнения. Управляемый код. FCL (.NET Framework Class Library). MSIL (Microsoft Intermediate Language) - промежуточный язык платформы Microsoft.NET.

2. Синтаксис языка программирования C#. Типы языка программирования C#. Объявление констант и переменных. Классы и объекты. Составляющие класса. Класс object. Пространства имен. Важные пространства имен. Доступ к пространствам имен. Типы приложений. Создание консольного приложения. Базовое приложение на языке программирования C# с вводом и выводом на консоль.

3. Система типов языка C#. Типы значений (типы значения), ссылочные типы (типы-ссылки), параметризованные типы (типы-шаблоны). Простые (элементарные) типы - это типы, имя и основные свойства которых известны компилятору. Sbyte, Byte, Short, Ushort, Int, uint, long, ulong, char, float, double, decimal, bool. Среди множества массов выделяют predetermined ссылочные типы object и string, которым соответствуют FCL-типы System.Object и System.String.

4. Литералы. Шестнадцатеричные литералы. Управляющие последовательности символов. Строковые литералы. Преобразование и приведение типов. Автоматическое преобразование типов. Приведение несовместимых типов. Преобразование типов в выражениях. Приведение типов в выражениях.
5. Арифметические операторы. Инкремент и декремент.
6. Операторы отношений и логические операторы. Сокращенные логические операторы. Оператор присваивания. Составные операторы присваивания. Поразрядные операторы. Поразрядные операторы И, ИЛИ, исключающее ИЛИ и НЕ.
7. Операторы сдвига. Поразрядные составные операторы присваивания.
8. Оператор ?.
9. Использование пробелов и круглых скобок. Приоритет операторов.
10. Инструкция if. Вложенные if-инструкции. Конструкция if-else-if. Инструкция switch. Вложенные инструкции switch.
11. Цикл for. Использование нескольких управляющих переменных цикла. Условное выражение. Отсутствие элементов в определении цикла. Бесконечный цикл. Циклы без тела. Объявление управляющей переменной в цикле for. Цикл while. Цикл do-while. Цикл foreach.
12. Использование инструкции break для выхода из цикла. Использование инструкции continue. Инструкция return. Инструкция goto.
13. Массивы. Одномерные массивы. Инициализация массива. Соблюдение "пограничного режима".
14. Многомерные массивы. Двумерные массивы. Массивы трех и более измерений. Инициализация многомерных массивов.
15. Рваные массивы. Присвоение значений ссылочным переменным массивов. Использование свойства Length. Использование свойства Length при работе с рваными массивами.
16. Цикл foreach.
17. Строки. Создание строк. Работа со строками. Массивы строк. Постоянство строк. Использование строк в switch-инструкциях.
18. Общая форма определения класса. Определение класса. Создание объектов. Переменные ссылочного типа и присвоение им значений. Методы. Добавление методов в класс Building. Возвращение из метода. Возврат значения. Использование параметров. Добавление параметризованного метода в класс Building. Как избежать написания недостижимого кода.
19. Конструкторы. Параметризованные конструкторы. Добавление конструктора в класс Building. Использование оператора new. Применение оператора new к переменным типа значений.
20. Сбор "мусора" и использование деструкторов. Деструкторы. Ключевое слово this.
21. Управление доступом к членам класса. Спецификаторы доступа C#. Применение спецификаторов доступа public и private.
22. Передача объектов методам. Как происходит передача аргументов. Использование ref- и out-параметров. Использование модификатора ref. Использование модификатора out. Использование модификаторов ref и out для ссылочных параметров. Использование переменного количества аргументов. Возвращение методами объектов. Возвращение методами массивов.
23. Перегрузка методов. Перегрузка конструкторов. Вызов перегруженного конструктора с помощью ссылки this.
24. Метод Main(). Возвращение значений из метода Main(). Передача аргументов методу Main(). Рекурсия. Использование модификатора типа static. Статические конструкторы.

25. Основы перегрузки операторов. Перегрузка бинарных операторов. Перегрузка унарных операторов. Выполнение операций над значениями встроенных С#-типов. Перегрузка операторов отношений. Перегрузка операторов true и false. Перегрузка логических операторов. Простой случай перегрузки логических операторов. Включение операторов, действующих по сокращенной схеме вычислений. Операторы преобразования. Рекомендации и ограничения по созданию перегруженных операторов.
26. Основы наследования. Доступ к членам класса и наследование. Использование защищенного доступа. Конструкторы и наследование. Вызов конструкторов базового класса. Наследование и сокрытие имен. Использование ключевого слова base для доступа к скрытому имени.
27. Создание многоуровневой иерархии. Последовательность вызова конструкторов. Ссылки на базовый класс и объекты производных классов.
28. Виртуальные методы и их переопределение. Зачем переопределять методы. Применение виртуальных методов. Использование абстрактных классов. Использование ключевого слова sealed для предотвращения наследования. Класс object. Приведение к объектному типу и восстановление значения. Использование класса object в качестве обобщенного типа данных.
29. Пространства имен. Объявление пространства имен. Пространства имен предотвращают конфликты по совпадению имен. Ключевое слово using. Вторая форма использования директивы using. Аддитивность пространств имен. Пространства имен могут быть вложенными. Пространство имен по умолчанию.
30. Препроцессор. #define. #if и #endif. #else и #elif. #undef. #error. #warning. #line. #region и #endregion. Компонентные файлы и модификатор доступа internal . Модификатор доступа internal.
31. Делегаты. Многоадресная передача. Класс System.Delegate. Назначение делегатов. События. Пример события для многоадресной передачи. Сравнение методов экземпляров классов со статическими методами, используемыми в качестве обработчиков событий. Использование событийных средств доступа. Смешанные средства обработки событий. Рекомендации по обработке событий в среде .NET Framework. Использование встроенного делегата EventHandler.
32. Индексаторы в языке программирования С#. Создание одномерных индексаторов. Перегрузка индексаторов. Индексаторам не требуется базовый массив. Многомерные индексаторы.
33. Свойства в языке программирования С#. Правила использования свойств в языке программирования С#. Свойства только для чтения. Примеры свойств. Использование индексаторов и свойств.
34. Интерфейсы. Реализация интерфейсов. Использование интерфейсных ссылок. Интерфейсные свойства. Интерфейсные индексаторы. Наследование интерфейсов. Сокрытие имен с помощью наследования интерфейсов. Явная реализация членов интерфейса. Закрытая реализация. Как избежать неопределенности с помощью явной реализации. Выбор между интерфейсом и абстрактным классом. Стандартные интерфейсы среды .NET Framework.
35. Структуры. Зачем нужны структуры.
36. Перечисления. Инициализация перечислений. Задание базового типа перечисления. Использование перечислений.
37. Организация С#-системы ввода-вывода. Байтовые и символьные потоки. Встроенные потоки. Классы потоков. Класс Stream. Байтовые классы потоков. Символьные классы потоков. Двоичные потоки. Консольный ввод-вывод данных. Считывание данных из консольного входного потока. Запись данных в консольный входный поток. Класс FileStream и файловый ввод-вывод на побайтовой основе. Как открыть и закрыть файл. Считывание байтов из объекта класса FileStream.

38. Запись данных в файл. Использование класса FileStream для копирования файла. Файловый ввод-вывод с ориентацией на символы. Использование класса StreamWriter. Использование класса StreamReader. Перенаправление стандартных потоков. Считывание и запись двоичных данных. Класс BinaryWriter. Класс BinaryReader. Демонстрация использования двоичного ввода-вывода. Файлы с произвольным доступом. Использование класса MemoryStream. Использование классов StringReader и StringWriter. Преобразование числовых строк во внутреннее представление.

7.1. Основная литература:

Маклецов С.В. Компьютерный практикум (часть 2). Учебно-методическое пособие. - Казань: Казанский (Приволжский) Федеральный университет, 2013. - 101 с.
https://kpfu.ru//staff_files/F1013609142/Makletsov.pdf

Маклецов С.В. Основы компьютерных наук. Часть 1. - Казань: Казан. ун-т, 2015. - 116 с.
http://kpfu.ru/portal/docs/F799866254/Maklecov._Osnovy.kompjuternyh.nauk..Chast.1.pdf

7.2. Дополнительная литература:

Липачёв, Евгений Константинович. Технология программирования. Базовые конструкции C/C++ : учебно-справочное пособие / Е. К. Липачёв ; Казан. федер. ун-т . Казань : [Казанский университет], 2012 .139 с. URL:
https://kpfu.ru//staff_files/F614532834/%D2%E5%F5%ED%EE%EB%EE%E3%E8%FF%20%EF%F0%EE

7.3. Интернет-ресурсы:

C# Practical Learning - <http://www.functionx.com/cs3pl/index.htm>

Visual Studio | C# - <https://docs.microsoft.com/ru-ru/visualstudio/get-started/csharp/>

Введение в программирование на C# 2.0 - <https://www.intuit.ru/studies/courses/109/109/info>

Введение в язык C# и .NET Framework -

<https://docs.microsoft.com/ru-ru/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the->

Документация по Visual Studio - <https://docs.microsoft.com/ru-ru/visualstudio/>

8. Материально-техническое обеспечение дисциплины(модуля)

Освоение дисциплины "Разработка приложений в языке C#" предполагает использование следующего материально-технического обеспечения:

Мультимедийная аудитория, вместимостью более 60 человек. Мультимедийная аудитория состоит из интегрированных инженерных систем с единой системой управления, оснащенная современными средствами воспроизведения и визуализации любой видео и аудио информации, получения и передачи электронных документов. Типовая комплектация мультимедийной аудитории состоит из: мультимедийного проектора, автоматизированного проекционного экрана, акустической системы, а также интерактивной трибуны преподавателя, включающей тач-скрин монитор с диагональю не менее 22 дюймов, персональный компьютер (с техническими характеристиками не ниже Intel Core i3-2100, DDR3 4096Mb, 500Gb), конференц-микрофон, беспроводной микрофон, блок управления оборудованием, интерфейсы подключения: USB, audio, HDMI. Интерактивная трибуна преподавателя является ключевым элементом управления, объединяющим все устройства в единую систему, и служит полноценным рабочим местом преподавателя. Преподаватель имеет возможность легко управлять всей системой, не отходя от трибуны, что позволяет проводить лекции, практические занятия, презентации, вебинары, конференции и другие виды аудиторной нагрузки обучающихся в удобной и доступной для них форме с применением современных интерактивных средств обучения, в том числе с использованием в процессе обучения всех корпоративных ресурсов. Мультимедийная аудитория также оснащена широкополосным доступом в сеть интернет. Компьютерное оборудование имеет соответствующее лицензионное программное обеспечение.

Компьютерный класс, представляющий собой рабочее место преподавателя и не менее 15 рабочих мест студентов, включающих компьютерный стол, стул, персональный компьютер, лицензионное программное обеспечение. Каждый компьютер имеет широкополосный доступ в сеть Интернет. Все компьютеры подключены к корпоративной компьютерной сети КФУ и находятся в едином домене.

Среда разработки MS Visual Studi

Программа составлена в соответствии с требованиями ФГОС ВПО и учебным планом по направлению 02.03.01 "Математика и компьютерные науки" и профилю подготовки Математическое и компьютерное моделирование .

Автор(ы):

Насрутдинов М.Ф. _____

Липачев Е.К. _____

"__" _____ 201__ г.

Рецензент(ы):

Маклецов С.В. _____

"__" _____ 201__ г.