

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное учреждение
высшего профессионального образования
"Казанский (Приволжский) федеральный университет"
Институт вычислительной математики и информационных технологий



подписано электронно-цифровой подписью

Программа дисциплины
Программирование С# М2.ДВ.2

Направление подготовки: 010400.68 - Прикладная математика и информатика

Профиль подготовки: Математическое моделирование

Квалификация выпускника: магистр

Форма обучения: очное

Язык обучения: русский

Автор(ы):

Тумаков Д.Н.

Рецензент(ы):

Бахтиева Л.У.

СОГЛАСОВАНО:

Заведующий(ая) кафедрой: Плещинский Н. Б.

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

Учебно-методическая комиссия Института вычислительной математики и информационных технологий:

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

Регистрационный No 9125514

Казань
2014

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) Тумаков Д.Н. Кафедра прикладной математики отделение прикладной математики и информатики ,
Dmitri.Tumakov@kpfu.ru

1. Цели освоения дисциплины

Овладение теоретическими и практическими знаниями языка С#

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

Данная учебная дисциплина включена в раздел " М2.ДВ.2 Профессиональный" основной образовательной программы 010400.68 Прикладная математика и информатика и относится к дисциплинам по выбору. Осваивается на 2 курсе, 3 семестр.

Данная учебная дисциплина включена в раздел "М2.ДВ.2 Профессиональный" основной образовательной программы по направлению "010400.68 Прикладная математика и информатика" и относится к вариативной части. Осваивается на 2 курсе, 3 семестр.

Дисциплина М2.ДВ.2 "Программирование С#" относится к профессиональному циклу дисциплин, предназначена для студентов 2 курса (3 семестр), опирается на знания, полученные студентом по курсам программирования и численным методам. Считается, что студент владеет одним из языков программирования, умеет разрабатывать алгоритмы для решения простейших задач по информатике.

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

В результате освоения дисциплины формируются следующие компетенции:

Шифр компетенции	Расшифровка приобретаемой компетенции
ОК-8 (общекультурные компетенции)	уметь применять методы и средства познания, обучения и самоконтроля для своего интеллектуального развития, повышения культурного уровня, профессиональной компетенции, сохранения своего здоровья, нравственного и физического самосовершенствования
ПК-10 (профессиональные компетенции)	способность применять в профессиональной деятельности современные языки программирования и языки баз данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии
ПК-11 (профессиональные компетенции)	способность приобретать и использовать организационно-управленческие навыки в профессиональной и социальной деятельности
ПК-8 (профессиональные компетенции)	способность формировать суждения о значении и последствиях своей профессиональной деятельности с учетом социальных, профессиональных и этических позиций
ПК-9 (профессиональные компетенции)	способность решать задачи производственной и технологической деятельности на профессиональном уровне, включая: разработку алгоритмических и программных решений в области системного и прикладного программирования
ОК-6 (общекультурные компетенции)	владеть наследием отечественной научной мысли, направленной на решение общегуманитарных и общечеловеческих задач

Шифр компетенции	Расшифровка приобретаемой компетенции
ОК-7 (общекультурные компетенции)	владеть культурой мышления, способностью к анализу, обобщению информации, постановке целей и выбору путей их достижения, владеет культурой устной и письменной речи
ОК-9 (общекультурные компетенции)	способность занимать гражданскую позицию в социально-личностных конфликтных ситуациях

В результате освоения дисциплины студент:

1. должен знать:

принципы "классического" программирования

2. должен уметь:

программировать на языке C

3. должен владеть:

основами языка C

4. должен демонстрировать способность и готовность:

практические навыки программирования и разработки начальных приложений в среде Visual Studio на языке C#

4. Структура и содержание дисциплины/ модуля

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

Форма промежуточного контроля дисциплины зачет в 3 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 MS.NET	3	1	0	0	2	устный опрос
2.	Тема 2. Обзор языка C#	3	2	0	0	2	дискуссия

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
3.	Тема 3. Использование структурных переменных	3	3	0	0	2	домашнее задание
4.	Тема 4. Операторы и исключения	3	4	0	0	2	письменная работа
5.	Тема 5. Методы и параметры	3	5	0	0	4	домашнее задание
6.	Тема 6. Массивы	3	6-7	0	0	2	домашнее задание
7.	Тема 7. Основы объектно-ориентированного программирования	3	8	0	0	2	домашнее задание
8.	Тема 8. Использование ссылочных типов данных	3	9-10	0	0	4	домашнее задание
9.	Тема 9. Создание и удаление объектов	3	11-12	0	0	2	домашнее задание
10.	Тема 10. Наследование в C#	3	13-14	0	0	2	письменная работа
11.	Тема 11. Операции, делегаты, события	3	15	0	0	2	домашнее задание
12.	Тема 12. Свойства и индексаторы	3	16	0	0	2	домашнее задание
13.	Тема 13. Атрибуты	3	17	0	0	2	домашнее задание
14.	Тема 14. Агрегации, пространства имен, сборки и модули	3	18	0	0	2	домашнее задание
	Тема . Итоговая форма контроля	3		0	0	0	зачет
	Итого			0	0	32	

4.2 Содержание дисциплины

Тема 1. Обзор платформы MS.NET

лабораторная работа (2 часа(ов)):

Обзор платформы MS.NET. Общеязыковая среда выполнения. Языки MS.NET Framework.

Тема 2. Обзор языка C#

лабораторная работа (2 часа(ов)):

Обзор языка С#. Структура программы на С#. Основные операции ввода/вывода.

Рекомендации по оформлению кода.

Тема 3. Использование структурных переменных

лабораторная работа (2 часа(ов)):

Использование структурных переменных. Общая система типов (Common Type System). Использование встроенных типов данных. Пользовательские типы данных. Преобразование типов.

Тема 4. Операторы и исключения

лабораторная работа (2 часа(ов)):

Операторы и исключения. Операторы в С#. Обработка исключений.

Тема 5. Методы и параметры

лабораторная работа (4 часа(ов)):

Методы и параметры. Использование методов. Использование параметров. Перегрузка методов.

Тема 6. Массивы

лабораторная работа (2 часа(ов)):

Массивы.

Тема 7. Основы объектно-ориентированного программирования

лабораторная работа (2 часа(ов)):

Основы объектно-ориентированного программирования. Классы и объекты . Инкапсуляция данных. Наследование и полиморфизм.

Тема 8. Использование ссылочных типов данных

лабораторная работа (4 часа(ов)):

Использование ссылочных типов данных. Reflection (рефлексия). Пространства имен. Приведение типов данных.

Тема 9. Создание и удаление объектов

лабораторная работа (2 часа(ов)):

Создание и удаление объектов. Использование конструкторов. Уничтожение объектов.

Тема 10. Наследование в С#

лабораторная работа (2 часа(ов)):

Наследование в С#. Использование интерфейсов. Использование абстрактных классов.

Тема 11. Операции, делегаты, события

лабораторная работа (2 часа(ов)):

Операции, делегаты, события. Операции. Создание и использование делегатов . События.

Тема 12. Свойства и индексаторы

лабораторная работа (2 часа(ов)):

Свойства. Индексаторы.

Тема 13. Атрибуты

лабораторная работа (2 часа(ов)):

Условные атрибуты. Атрибут DLLImport. Пользовательские атрибуты.

Тема 14. Агрегации, пространства имен, сборки и модули

лабораторная работа (2 часа(ов)):

Агрегации, пространства имен, сборки и модули. Использование внутренних (internal) классов, методов и данных. Использование агрегаций. Фабрики классов. Пространства имен. Модули и сборки.

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 MS.NET	3	1	подготовка к устному опросу	2	устный опрос
2.	Тема 2. Обзор языка C#	3	2	подготовка к дискуссии	2	дискуссия
3.	Тема 3. Использование структурных переменных	3	3	подготовка домашнего задания	2	домашнее задание
4.	Тема 4. Операторы и исключения	3	4	подготовка к письменной работе	4	письменная работа
5.	Тема 5. Методы и параметры	3	5	подготовка домашнего задания	4	домашнее задание
6.	Тема 6. Массивы	3	6-7	подготовка домашнего задания	2	домашнее задание
7.	Тема 7. Основы объектно-ориентированного программирования	3	8	подготовка домашнего задания	2	домашнее задание
8.	Тема 8. Использование ссылочных типов данных	3	9-10	подготовка домашнего задания	4	домашнее задание
9.	Тема 9. Создание и удаление объектов	3	11-12	подготовка домашнего задания	2	домашнее задание
10.	Тема 10. Наследование в C#	3	13-14	подготовка к письменной работе	4	письменная работа
11.	Тема 11. Операции, делегаты, события	3	15	подготовка домашнего задания	4	домашнее задание
12.	Тема 12. Свойства и индексы	3	16	подготовка домашнего задания	4	домашнее задание
13.	Тема 13. Атрибуты	3	17	подготовка домашнего задания	2	домашнее задание
14.	Тема 14. Агрегации, пространства имен, сборки и модули	3	18	подготовка домашнего задания	2	домашнее задание
	Итого				40	

5. Образовательные технологии, включая интерактивные формы обучения

Активные и интерактивные формы занятий в сочетании с внеаудиторной работой

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

Тема 1. Обзор платформы MS.NET

устный опрос , примерные вопросы:

домашнее задание, примерные вопросы: 1. Что такое платформа MS.NET? Каковы её преимущества? 2. Перечислите основные понятия платформы .NET. 3. Охарактеризуйте компоненты MS.NET Framework. 4. Что такое общезыковая среда выполнения? Какие ее основные сервисы? 5. Перечислите основные классы библиотеки .NET Framework 2 балла

Тема 2. Обзор языка C#

дискуссия , примерные вопросы:

дискуссия, примерные вопросы: 1. Откуда начинается выполнение приложения C#? 2. Когда приложение заканчивает работу? 3. Сколько классов может содержать приложение C#? 4. Сколько методов Main может содержать приложение? 5. Как прочитать данные, введенные пользователем с клавиатуры? 6. В каком пространстве имен находится класс Console? 7. Что произойдет при необработанном в приложении исключении? 2 балла

Тема 3. Использование структурных переменных

домашнее задание , примерные вопросы:

домашнее задание, примерные вопросы: 1. Что такое общезыковая система типов? 2. Может ли структурная переменная иметь значение null? 3. Можно ли не инициализировать переменные в C#? Почему? 4. Можно ли потерять данные в результате неявного преобразования? 2 балла

Тема 4. Операторы и исключения

письменная работа , примерные вопросы:

письменная работа, примерные задания: Упражнение 4.1 Написать программу, которая читает с экрана число от 1 до 365 (номер дня в году), переводит это число в месяц и день месяца. Например, число 40 соответствует 9 февраля (високосный год не учитывать). Упражнение 4.2 Добавить к задаче из предыдущего упражнения проверку числа введенного пользователем. Если число меньше 1 или больше 365, программа должна выработать исключение, и выдавать на экран сообщение. Домашнее задание 4.1 Изменить программу из упражнений 4.1 и 4.2 так, чтобы она учитывала год (високосный или нет). Год вводится с экрана. Год високосный если выполняются одновременно два условия: 1) год делится на 4; 2) не делится на 100 и не делится на 400. 2 балла

Тема 5. Методы и параметры

домашнее задание , примерные вопросы:

домашнее задание, примерные вопросы: 1. Объясните что такое методы и почему они важны? 2. Опишите три возможных пути передачи параметров и соответствующие ключевые слова C#. 3. Когда создаются и уничтожаются локальные переменные? 4. Какое ключевое слово должно быть добавлено к определению метода, чтобы его можно было вызывать из другого метода? 5. Что входит в сигнатуру метода? 2 балла

Тема 6. Массивы

домашнее задание , примерные вопросы:

домашнее задание, примерные вопросы: 1. В чем отличия коллекций от массивов? 2. Перечислите основные свойства и методы класса System.Array. 3. Приведите примеры описания массивов. 4. Как передавать и возвращать массивы из методов. 5. Объясните принцип работы цикла foreach. 4 балла

Тема 7. Основы объектно-ориентированного программирования

домашнее задание , примерные вопросы:

домашнее задание, примерные вопросы: 1. Объясните концепцию абстракции, и почему она важна для программной инженерии? 2. Назовите два принципа инкапсуляции. 3. Опишите наследование в контексте ООП. 4. Что такое полиморфизм? Как он связан с ранним и поздним связыванием? 5. Опишите разницу между интерфейсами, абстрактными классами и конкретными классами. 2 балла

Тема 8. Использование ссылочных типов данных

домашнее задание , примерные вопросы:

домашнее задание, примерные вопросы: 1. Как распределяется память для переменных ссылочного типа? 2. Какое значение присваивают ссылочной переменной, чтобы показать, что она не указывает на объект? Что произойдет, если обратиться к ней как к объекту? 3. Какой класс является базовым для всех классов C#? 4. Объясните разницу между операцией преобразования типа (cast) и оператором as. 2 балла

Тема 9. Создание и удаление объектов

домашнее задание, примерные вопросы:

домашнее задание, примерные вопросы: 1. Как происходит создание и удаление объектов? 2. В каких случаях используются закрытые конструкторы? 3. Возможна ли перегрузка конструкторов и деструкторов? Приведите примеры. 4. Какой метод вызывает сборщик мусора даже, если память ещё не переполнена? 5. В чем смысл использования оператора using? 2 балла

Тема 10. Наследование в C#

письменная работа, примерные вопросы:

письменная работа, примерные задания: Упражнение 10.1. Создать интерфейс ICipher, который определяет методы поддержки шифрования строк. В интерфейсе объявляются два метода encode() и decode(), которые используются для шифрования и дешифрования строк, соответственно. Создать класс ACipher, реализующий интерфейс ICipher. Класс шифрует строку посредством сдвига каждого символа на одну ?алфавитную? позицию выше. Например, в результате такого сдвига буква А становится буквой Б. Создать класс BCipher, реализующий интерфейс ICipher. Класс шифрует строку, выполняя замену каждой буквы, стоящей в алфавите на i-й позиции, на букву того же регистра, расположенную в алфавите на i-й позиции с конца алфавита. Например, буква В заменяется на букву Э. Написать программу, демонстрирующую функционирование классов. Упражнение 10.2. Создать класс Figure для работы с геометрическими фигурами. В качестве полей класса задаются цвет фигуры, состояние ?видимое/невидимое?. Реализовать операции: передвижение геометрической фигуры по горизонтали, по вертикали, изменение цвета, опрос состояния (видимый/невидимый). Метод вывода на экран должен выводить состояние всех полей объекта. Создать класс Point (точка) как потомок геометрической фигуры. Создать класс Circle (окружность) как потомок точки. В класс Circle добавить метод, который вычисляет площадь окружности. Создать класс Rectangle (прямоугольник) как потомок точки, реализовать метод вычисления площади прямоугольника. Точка, окружность, прямоугольник должны поддерживать методы передвижения по горизонтали и вертикали, изменения цвета. Подумать, какие методы можно объявить в интерфейсе, нужно ли объявлять абстрактный класс, какие методы и поля будут в абстрактном классе, какие методы будут виртуальными, какие перегруженными. 2 балла

Тема 11. Операции, делегаты, события

домашнее задание, примерные вопросы:

домашнее задание, примерные вопросы: 1. Может ли арифметическое присваивание (+=, -=, *=, /= и %=) быть перегружено? 2. В каком случае операция преобразования типа должна быть явной? 3. Что такое делегат? 4. Как можно подписаться на событие? 5. Каким образом можно вызвать метод, подписавшийся на событие? 7 балла

Тема 12. Свойства и индексы

домашнее задание, примерные вопросы:

домашнее задание, примерные вопросы: 1. Каким образом можно объявить свойство только для чтения? 2. Может ли класс содержать индексатор, зависящий от двух параметров разного типа? 3. Как в классе объявить индексатор только для чтения? 7 балла

Тема 13. Атрибуты

домашнее задание, примерные вопросы:

домашнее задание, примерные вопросы: 1. Можно ли отметить один объект класса используя атрибут? 2. Где хранятся значения атрибутов? 3. Какой механизм используется для определения значения атрибутов при выполнении приложения? 2 балла

Тема 14. Агрегации, пространства имен, сборки и модули

домашнее задание, примерные вопросы:

домашнее задание, примерные вопросы: 1. Пусть есть два cs файла. Файл alpha.cs содержит класс Alpha с internal методом Method. Файл beta.cs содержит класс Beta с internal методом с тем же названием Method. Может ли Alpha.Method вызвать Beta.Method, и наоборот? 2. Агрегации это связь объектов или классов? 3. Откомпилируется ли следующий код без ошибок? namespace Outer.Inner { class Wibble{ } } namespace Test { using Outer.Inner; class SpecialWible:Inner.Wible{ } } 12 балла

Тема . Итоговая форма контроля

Примерные вопросы к зачету:

1. Что такое платформа MS.NET? Каковы её преимущества?
2. Перечислите основные понятия платформы .NET.
3. Охарактеризуйте компоненты MS.NET Framework.
4. Что такое общезыковая среда выполнения? Какие ее основные сервисы?
5. Перечислите основные классы библиотеки .NET Framework
6. Откуда начинается выполнение приложения C#?
7. Когда приложение заканчивает работу?
8. Сколько классов может содержать приложение C#?
9. Сколько методов Main может содержать приложение?
10. Как прочесть данные, введенные пользователем с клавиатуры?
11. В каком пространстве имен находится класс Console?
12. Что произойдет при необработанном в приложении исключении?
13. Что такое общезыковая система типов?
14. Может ли структурная переменная иметь значение null?
15. Можно ли не инициализировать переменные в C#? Почему?
16. Можно ли потерять данные в результате неявного преобразования?
17. Перечислите основные группы операторов, выделив их характерные особенности.
18. Что такое исключение? Какие классы для обработки исключительных ситуаций вы знаете?
19. Как организовать обработку исключительной ситуации?
20. Объясните что такое методы и почему они важны?
21. Опишите три возможных пути передачи параметров и соответствующие ключевые слова C#.
22. Когда создаются и уничтожаются локальные переменные?
23. Какое ключевое слово должно быть добавлено к определению метода, чтобы его можно было вызывать из другого метода?
24. Что входит в сигнатуру метода?
25. В чем отличия коллекций от массивов?
26. Перечислите основные свойства и методы класса System.Array.
27. Приведите примеры описания массивов.
28. Как передавать и возвращать массивы из методов.
29. Объясните принцип работы цикла foreach.
30. Объясните концепцию абстракции, и почему она важна для программной инженерии?
31. Назовите два принципа инкапсуляции.
32. Опишите наследование в контексте ООП.
33. Что такое полиморфизм? Как он связан с ранним и поздним связыванием?
34. Опишите разницу между интерфейсами, абстрактными классами и конкретными классами.
35. Как распределяется память для переменных ссылочного типа?
36. Какое значение присваивают ссылочной переменной, чтобы показать, что она не указывает на объект? Что произойдет, если обратиться к ней как к объекту?
37. Какой класс является базовым для всех классов C#?

38. Объясните разницу между операцией преобразования типа (cast) и оператором as.
39. Как происходит создание и удаление объектов?
40. В каких случаях используется закрытые конструкторы?
41. Возможна ли перегрузка конструкторов и деструкторов? Приведите примеры.
42. Какой метод вызывает сборщик мусора даже, если память ещё не переполнена?
43. В чем смысл использования оператора using?
44. В чем отличие между public, private и protected полями?
45. Как переопределить метод базового класса у класса-потомка?
46. Что такое абстрактный класс?
47. Что такое интерфейс? В чем отличие интерфейса от абстрактного класса?
48. Допустимо ли множественное наследование?
49. Пусть есть два cs файла. Файл alpha.cs содержит класс Alpha с internal методом Method. Файл beta.cs содержит класс Beta с internal методом с тем же названием Method. Может ли Alpha.Method вызвать Beta.Method, и наоборот?
50. Агрегации это связь объектов или классов?
51. Может ли арифметическое присваивание (+=, -=, *=, /= и %=) быть перегружено?
52. В каком случае операция преобразования типа должна быть явной?
53. Что такое делегат?
54. Как можно подписаться на событие?
55. Каким образом можно вызвать метод, подписавшийся на событие?
56. Каким образом можно объявить свойство только для чтения?
57. Может ли класс содержать индексатор, зависящий от двух параметров разного типа?
58. Как в классе объявить индексатор только для чтения?
59. Можно ли отметить один объект класса используя атрибут?
60. Где хранятся значения атрибутов?
61. Какой механизм используется для определения значения атрибутов при выполнении приложения?

7.1. Основная литература:

1. Объектно-ориентированное программирование на C# : [учебное пособие] / Андрианова А. А., Исмагилов Л. Н., Мухтарова Т. М. ; Казан. (Приволж.) федер. ун-т, Ин-т вычисл. математики и информ. технологий .- Казань : [Казанский (Приволжский) федеральный университет], 2012 . - 140 с.
2. Практикум по курсу "Объектно-ориентированное программирование" на языке C# : [учебное пособие] / А. А. Андрианова, Л. Н. Исмагилов, Т. М. Мухтарова ; Казан. (Приволж.) федер. ун-т, Ин-т вычисл. математики и информ. технологий .- Казань : Казанский университет, 2012 . - 115 с.
3. Программирование на языке C# : учебно-методическое пособие / И. Л. Александрова, Д.Н. Тумаков ; Казан. федер. ун-т, Ин-т вычисл. мат. и информ. технологий .- Казань : [б. и.], 2011 .- 103 с. ; 21.
4. Введение в программирование на языке Visual C#: Учебное пособие / С.Р. Гуриков. - М.: Форум: НИЦ ИНФРА-М, 2013. - 448 с.: 70x100 1/16. - (Высшее образование: Бакалавриат). (переплет) ISBN 978-5-91134-738-3, 500
<http://www.znaniyum.com/bookread.php?book=404441>
5. Культин, Н. Б. Microsoft Visual C# 2010 в задачах и примерах / Никита Культин.- СПб.: БХВ-Петербург, 2009.- 310 с.. - ISBN 978-5-9775-0410-2.
<http://www.znaniyum.com/bookread.php?book=350678>
6. Культин Н. Б. Основы программирования в Microsoft Visual C# 2010. ? СПб.: БХВ-Петербург, 2011. ? 364 с. ? (Самоучитель). - ISBN 978-5-9775-0589-5.

<http://www.znaniium.com/bookread.php?book=351294>

7.2. Дополнительная литература:

1. Новиков, Ф. А. Дискретная математика для программистов: Учеб. пособие / Ф.А.Новиков. - 2-е изд. - СПб. и др.: Питер, 2004. - 363 с.
2. Delphi. Программирование на языке высокого уровня: Учеб. для вузов, обучающихся по направлению подгот. дипломир. спец. "Информатика и вычислительная техника" / В.В.Фаронов.- СПб. и др.: Питер, 2004.- 639с.: ил. - (Учебник для вузов).- Библиогр.: с.628.
3. Информатика и программирование: учебник для студ. вузов / Е. П. Истомина, С. Ю. Неклюдов, В. И. Романченко.- СПб.: Андреевский изд. дом, 2006.- 248 с.
4. Структуры данных и алгоритмы: Учеб. пособие / Альфред В.Ахо, Джон Э.Хопкрофт, Джеффри Д.Ульман ; Пер. с англ. и ред. А.А.Минько.- М. и др.: Издат. дом "Вильямс", 2000.- 382с.: ил. - Библиогр.: с.369-374.

7.3. Интернет-ресурсы:

Microsoft Virtual Academy -

<http://www.microsoftvirtualacademy.com/training-courses/exciting-programming-c-sharp-rus>

Информатика: Курс лекций. - <http://znaniium.com/catalog.php?bookinfo=204273>

Методы программирования. Компьютерные вычисления -

<http://znaniium.com/bookread.php?book=350418>

Практикум по курсу "Алгоритмизация и программирование" -

http://libweb.ksu.ru/ebooks/09_63.pdf

Руководство по программированию на С# - <http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx>

8. Материально-техническое обеспечение дисциплины(модуля)

Освоение дисциплины "Программирование С#" предполагает использование следующего материально-технического обеспечения:

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.

Microsoft Visual Studio

Программа составлена в соответствии с требованиями ФГОС ВПО и учебным планом по направлению 010400.68 "Прикладная математика и информатика" и магистерской программе Математическое моделирование .

Автор(ы):

Тумаков Д.Н. _____

"__" _____ 201__ г.

Рецензент(ы):

Бахтиева Л.У. _____

"__" _____ 201__ г.