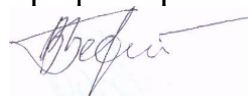


МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего
образования «Московский государственный университет технологий и управления
имени К.Г. Разумовского (Первый казачий университет)»
Донской казачий государственный институт пищевых технологий и бизнеса (филиал)
Федеральное государственное бюджетное образовательное учреждение высшего
образования «Московский государственный университет технологий и управления
имени К.Г. Разумовского (Первый казачий университет)»

«УТВЕРЖДАЮ»

Заведующий кафедрой «МФиИТ»
доктор физико-математических наук,
профессор



В.Н. Беркович

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

«Программирование»

(наименование учебной дисциплины (модуля))

По направлению подготовки:

15.03.04 «Автоматизация технологических процессов и производств»

Профиль подготовки:

«Автоматизация технологических процессов и производств»


Квалификация:

Бакалавр

Ростов-на-Дону 2017 г.

Рабочая программа учебной дисциплины «Программирование» разработана на основании федерального государственного образовательного стандарта высшего образования по направлению подготовки 15.03.05 «Автоматизация технологических процессов и производств», профиль подготовки «Автоматизация технологических процессов и производств» (уровень бакалавриата), утвержденного приказом Министерства образования и науки Российской Федерации от 12 марта 2015 г. № 200 учебного плана по образовательной программе высшего образования «Автоматизация технологических процессов и производств».

Рабочая программа учебной дисциплины разработана рабочей группой в составе: доцент, Складов А.В.

Руководитель образовательной программы высшего образования
к.т.н., доцент  Павлова И.В.

Рабочая программа учебной дисциплины обсуждена и утверждена на заседании кафедры «Математика, физика и информационные технологии»
Протокол № 1 от «29» августа 2017 года

Заведующий кафедрой
ученая степень, ученое звание

д. физ – мат н,



Беркович В.Н.

(подпись)

Рабочая программа учебной дисциплины рекомендована к утверждению представителями организаций-работодателей:

ООО «ДонСетьСтройПроект»,
Начальник отдела АИИС КУЭ, МОП
и ТСБ



(подпись)

С.Б. Бурцев

ООО «Джинт»,
Генеральный директор, к.т.н.



(подпись)

И.В. Дерябкин

Содержание

1. Цели и задачи освоения дисциплины.....	4
2. Место дисциплины в структуре ООП.....	4
3. Общие требования к содержанию и уровню освоения дисциплины.....	4
4. Трудоемкость дисциплины и виды учебной работы.....	5
5. Содержание дисциплины	
5.1. Учебно-образовательные модули дисциплины, их трудоемкость и виды учебной работы.....	5
5.2. Содержание модулей дисциплины.....	6
6. Методические указания для студентов по освоению дисциплины.....	10
7. Образовательные технологии.....	11
8. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.....	11
9. Организация образовательного процесса для лиц с ограниченными возможностями.....	32
10. Учебно-методическое и информационное обеспечение дисциплины....	32
11. Материально-техническое обеспечение дисциплины.....	33

1. Цели и задачи освоения дисциплины

Целями освоения учебной дисциплины (модуля) «Программирование» являются:

- систематическое изучение языков программирования высокого уровня;
- формирование у студентов знаний, умений и владений в области алгоритмизации задач вычислительного характера и задач автоматизированной обработки данных;
- изучение сложных структур, данных и их применение для решения различных задач обработки данных на ЭВМ;
- расширение представлений о современном программном обеспечении, языках программирования высокого уровня;
- знакомство с современными технологиями программирования – ООП - технологией и визуальным конструированием программ.

Цели и задачи данного курса вытекают из необходимости практического применения ЭВМ и закрепления полученных умений и навыков работы со средствами вычислительной техники, применения различных языков и методов программирования для исследования математических и информационных моделей.

2. Место дисциплины в структуре ООП

Дисциплина «Программирование» включена в базовую часть профессионального цикла, код по учебному плану дисциплины – Б1.Б.12 Она определяет содержание базовой подготовки студентов в области использования программных средств вычислительной техники и решения задач с помощью ЭВМ. Являясь одной из центральных дисциплин по общей компьютерной подготовке, она играет важную роль в учебном процессе и дальнейшей научно-практической деятельности студентов.

Изучение данного курса должно способствовать формированию условий для успешного освоения и применения ЭВМ при изучении дисциплин естественно-научного цикла, а также дисциплин специальных курсов.

3. Общие требования к содержанию и уровню освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

- способностью выбирать методы и средства измерения эксплуатационных характеристик оборудования, средств и систем автоматизации, контроля, диагностики, испытаний и управления, настройки и обслуживания: системного, инструментального и прикладного программного обеспечения данных средств и систем (ПК-24);
- способностью участвовать во внедрении и корректировке технологических процессов, средств и систем автоматизации, управления, контроля, диагностики при подготовке производства новой продукции и оценке ее конкурентоспособности (ПК-32);
- способностью участвовать в разработке новых автоматизированных и

автоматических технологий производства продукции и их внедрении, оценке полученных результатов, подготовке технической документации по автоматизации производства и средств его оснащения (ПК-33).

В результате изучения дисциплины студент должен
знать:

- основные приемы алгоритмизации и программирования на языке высокого уровня;
- синтаксис и семантику основных конструкций языка программирования;
- способы организации структур данных и основные алгоритмы обработки этих данных;

уметь:

- разрабатывать алгоритмы решения и программировать решение простейших задач обработки данных в предметной области;

владеть:

- приемами разработки основных структур алгоритмов и программ на языках программирования высокого уровня.

4. Трудоемкость дисциплины и виды учебной работы

Общая трудоёмкость дисциплины «Программирование» составляет 288 академических часа, что соответствует 8-ми зачётным единицам.

Формы текущего контроля успеваемости: тестирование по модулям, проверка отчётов по лабораторным работам и их защита, проверка и защита курсовой работы.

Промежуточная аттестация: экзамен.

Таблица 1

Распределение трудоемкости дисциплины по видам учебной работы

Вид учебной работы	Академические часы	Зачётные единицы
Общая трудоемкость дисциплины, в том числе:	288	8
Аудиторные занятия (всего), в том числе:	140	3,9
<i>Лекции</i>	34	0,9
<i>Лабораторные работы</i>	70	2
<i>Практические работы</i>	36	1
Самостоятельная работа (всего)	103	2,9
Контроль	45	1,2

5. Содержание дисциплины

5.1. Учебно-образовательные модули дисциплины, их трудоемкость и виды учебной работы

Таблица 2

Наименование модуля и темы	ВСЕГО	Аудиторная работа, час.			СРС, час.	Контроль
		Лекции	Лаб. раб.	Пр. раб.		
Модуль 1. Основные понятия алгоритмизации.	8	2	2	2	2	
Модуль 2. Основы программирования.						
Тема 2.1. Языки и системы программирования.	4	1			3	
Тема 2.2. Методы программирования.	4	1			3	
Тема 2.3. Основные элементы языка.	4	1			3	
Тема 2.4. Операторы языка.	34	2	12	12	8	
Тема 2.5. Массивы.	9	2	2	2	3	
Тема 2.6. Строки, множества, структуры.	9	2	2	2	3	
Тема 2.7. Процедуры и функции.	6	1	2		3	
Итого по модулю 2:	70	10	18	16	26	
Модуль 3. Файлы.	17	4	8		5	
Модуль 4. Библиотеки.	9	2	4		3	
Модуль 5. Основные принципы объектно-ориентированного программирования (ООП). Визуальное событийно-управляемое программирование.	70	7	16	4	22	21
Модуль 6. Меню и панели инструментов.	36	3	10		15	8
Модуль 7. Стандартные элементы управления.	42	3	6	10	15	8
Модуль 8. Диалоги. Графика.	36	3	6	4	15	8
ИТОГО:	288	34	70	36	103	*45

**Осуществляется в виде допуска к лабораторным работам и по результатам их защиты, а также тестирование по модулям дисциплины*

5.2. Содержание модулей дисциплины

Модуль 1. Основные понятия алгоритмизации.

Понятие алгоритма. Свойства алгоритмов. Формы записей алгоритмов. Общие принципы построения алгоритмов. Основные алгоритмические конструкции: линейные, разветвляющиеся, циклические. Данные: понятие и типы. Основные базовые типы данных и их характеристика. Структурированные типы данных и их характеристика. Методы сортировки данных.

Практическая работа №1. «Блок-схемы алгоритмов. ГОСТ»

Лабораторная работа №1. «Составление блок-схем алгоритмов»

Модуль 2. Основы программирования.

Тема 2.1. Языки и системы программирования.

Эволюция языков программирования. Классификация языков программирования. Элементы языков программирования. Понятие системы

программирования. Исходный, объектный и загрузочный модули. Интегрированная среда программирования.

Тема 2.2. Методы программирования.

Методы программирования: структурный, модульный, объектно-ориентированный. Достоинства и недостатки методов программирования. Общие принципы разработки программного обеспечения. Жизненный цикл программного обеспечения. Типы приложений. Консольные приложения. Оконные Windows приложения. Web-приложения. Библиотеки. Web-сервисы.

Тема 2.3. Основные элементы языка.

История развития языка программирования C#. Структурная схема программы на алгоритмическом языке C#. Лексика языка. Переменные и константы. Типы данных. Выражения и операции.

Тема 2.4. Операторы языка.

Синтаксис операторов: присваивания, ввода-вывода, безусловного и условного переходов, циклов. Составной оператор. Вложенные условные операторы. Цикл с параметром. Циклы с предусловием и постусловием.

Практическая работа № 2. «Алгоритмы следования»

Практическая работа № 3. «Алгоритмы ветвления»

Практическая работа № 4. «Алгоритмы выбора»

Практическая работа № 5. «Алгоритмы – цикл с параметром»

Практическая работа № 6. «Алгоритмы – цикл с предусловием»

Практическая работа № 7. «Алгоритмы – цикл с постусловием»

Лабораторная работа № 2. «Создание приложения, использующего ввод/вывод строк»

Лабораторная работа № 3. «Создание приложения, использующего стандартные числовые типы»

Лабораторная работа № 4. «Создание приложения, использующего условный оператор»

Лабораторная работа № 5. «Создание приложения, использующего переключатель»

Лабораторная работа № 6. «Создание приложения, использующего цикл «for»

Лабораторная работа № 7. «Создание консольного приложения, использующего цикл «while»

Тема 2.5. Массивы.

Массивы как структурированный тип данных. Объявление массива. Ввод и вывод одномерных массивов. Ввод и вывод двумерных массивов. Обработка массивов. Стандартные функции для массива целых и вещественных чисел.

Практическая работа № 8. «Алгоритмы обработки массивов»

Лабораторная работа № 8 «Создание приложения, использующего массивы»

Тема 2.6. Строки, множества, структуры.

Структурированные типы данных: строки, множества, структуры. Объявление строковых типов данных. Поиск, удаление, замена и добавление символов в строке. Операции со строками. Стандартные функции и процедуры для работы со строками. Объявление множества. Операции над множествами. Структуры.

Практическая работа № 9. «Алгоритмы обработки строк»

Лабораторная работа № 9. «Создание приложения, использующего класс string»

Тема 2.7. Процедуры и функции.

Понятие подпрограммы. Процедуры и функции, их сущность, назначение, различие. Организация процедур, стандартные процедуры. Процедуры, определенные пользователем: синтаксис, передача аргументов. Формальные и фактические параметры. Процедуры с параметрами, описание процедур. Функции: способы организации и описание. Вызов функций, рекурсия. Программирование рекурсивных алгоритмов. Стандартные функции.

Лабораторная работа №10. Использование процедур и функций.

Модуль 3. Файлы.

Типы файлов. Организация доступа к файлам. Файлы последовательного доступа. Открытие и закрытие файла последовательного доступа. Запись в файл и чтение из файла последовательного доступа. Исключения. Обработка исключений. Файлы произвольного доступа. Порядок работы с файлами произвольного доступа. Создание структуры записи. Открытие и закрытие файла произвольного доступа. Запись и считывание из файла произвольного доступа. Использование файла произвольного доступа. Стандартные процедуры и функции для файлов разного типа.

Лабораторная работа №11. «Создание приложения, взаимодействующего с файловой системой»

Лабораторная работа №12. «Создание приложения, использующего ввод/вывод в текстовый файл»

Лабораторная работа №13. «Создание приложения, использующего коллекции»

Лабораторная работа №14. «Создание приложения, использующего обработку исключений»

Модуль 4. Библиотеки.

Программирование модулей. Модуль: синтаксис, заголовок, разделы. Библиотеки подпрограмм: понятие и виды. Схемы вызова библиотек. Статическое и динамическое связывание. Использование библиотек подпрограмм.

Лабораторная работа № 15. Создание библиотеки подпрограмм.

Лабораторная работа № 16. Использование библиотеки подпрограмм.

Модуль 5. Основные принципы объектно-ориентированного программирования (ООП). Визуальное событийно-управляемое программирование.

История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Событийно-управляемая модель программирования. Компонентно-ориентированный подход. Классы объектов. Компоненты и их свойства. Классы объектно-ориентированного языка программирования: виды,

назначение, свойства, методы, события. Объявление класса, свойств и методов экземпляра класса. Наследование. Перегрузка методов. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение. Дополнительные элементы управления. Свойства компонентов (элементов управления). Виды свойств. Синтаксис определения свойств. Категория свойств. Назначение свойств и их влияние на результат. Управление объектом через свойства. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий. Процедуры, определенные пользователем: синтаксис, передача аргументов. Вызов событий. Разработка функционального интерфейса приложения. Создание интерфейса.

Практическая работа №10. «Событийно управляемое программирование»

Практическая работа №11. «Изучение среды Visual Studio»

Лабораторная работа №17. «Создание приложения, использующего класс»

Лабораторная работа № 18. «Создание приложения, использующего конструктор класса»

Лабораторная работа № 19. «Создание приложения, использующего поля класса»

Лабораторная работа № 20. «Создание приложения, использующего методы класса»

Лабораторная работа № 21. «Создание приложения, использующего свойства класса»

Лабораторная работа № 22. «Создание приложения, использующего события класса»

Лабораторная работа № 23. «Создание главной формы приложения»

Лабораторная работа № 24. «Создание вспомогательных форм»

Модуль 6. Меню и панели инструментов.

Меню и панели инструментов: ContainerMenuStrip, MenuStrip, StatusStrip, ToolStrip. Использование командных кнопок. Использование меток и полей ввода: Label, RichTextBox, TextBox.

Лабораторная работа № 25. «Создание главного меню приложения»

Лабораторная работа № 26. «Создание контекстного меню приложения»

Лабораторная работа № 27. «Создание строки статуса»

Лабораторная работа № 28. «Создание панели инструментов»

Лабораторная работа № 29. «Использование меток и полей ввода»

Модуль 7. Стандартные элементы управления.

Стандартные элементы управления: CheckBox, RadioButton, ListBox, ComboBox. Использование TreeView и ListView. Использование DataGridView, DateTimePicker.

Практическая работа №12. «Элементы управления Windows Forms»

Практическая работа №13. «Элементы управления Label, Button, TextBox»

Практическая работа №14. «Элементы управления MainMenu, ContextMenu»

Практическая работа №15. «Элементы управления CheckBox, RadioButton, GroupBox»

Практическая работа №16. «Элементы управления ListBox, ComboBox, RichTextBox»

Лабораторная работа № 30. «Использование флажков и радио кнопок»

Лабораторная работа № 31. «Создание списков»

Лабораторная работа № 32 «Использование TreeView и ListView»

Модуль 8. Диалоги. Графика.

Использование стандартных диалогов. Использование PictureBox. Рисование линий и фигур. Заливка фигур. Вывод текста в графическом режиме.

Практическая работа №17. «Диалоговые окна»

Практическая работа №18. «Класс Application»

Лабораторная работа № 33. «Использование стандартных диалогов»

Лабораторная работа № 34. «Использование PictureBox»

Лабораторная работа № 35. «Вывод текста в графическом режиме»

6. Методические указания для студентов по освоению дисциплины

Методические рекомендации для студентов по отдельным формам самостоятельной работы

Работа с учебной литературой

При работе с учебной литературой необходимо подобрать литературу, научиться правильно ее читать, вести записи.

Для получения допуска к лабораторным работам необходимо подготовить в бумажном или электронном носителе теоретическую часть лабораторной работы. При подготовке можно использовать литературу, которая перечислена в данной рабочей программе, а также воспользоваться конспектом лекций.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс.

Самопроверка

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на лабораторных занятиях и самостоятельно студенту рекомендуется воспроизвести по памяти определения, выводы формул, а также воспроизвести решение некоторых задач, приводимых в учебных материалах (лекциях, учебниках).

Консультации

Если в процессе самостоятельной работы над изучением теоретического материала или при решении задач у студента возникают вопросы, разрешить которые самостоятельно не удастся, необходимо обратиться к преподавателю для получения у него разъяснений или указаний. В своих вопросах студент должен четко выразить, в чем он испытывает затруднения, характер этого затруднения.

Подготовка к промежуточной аттестации

Подготовка к промежуточной аттестации способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к промежуточной аттестации, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. При подготовке студент может пройти репетиционное тестирование в электронном учебном курсе по данной дисциплине в системе Moodle. Результат прохождения теста каждый студент видит сразу и это позволяет обучающимся самостоятельно определять свои личностные достижения и уровень освоения материала в процессе подготовки промежуточной аттестации.

7. Образовательные технологии

Современные технологии в педагогическом образовании рассматриваются как средство, с помощью которого может быть реализована система педагогического образования.

В результате использования в образовательном процессе современных технологий можно достичь следующих результатов: вызвать у студентов устойчивую мотивацию к учебной деятельности, способность к рефлексии и оцениванию самими обучающимися своего прогресса (чувство компетентности), проявление обучающимися инициативы и полноценная их само-реализация.

Современные информационные технологии в рамках данной дисциплины предполагают использование комплекса технического, учебно-методического, программного и организационного обеспечения на компьютерной основе и цифровых образовательных ресурсов, к которым относятся компьютеры, интерактивные доски, принтеры, проекционные устройства, устройства для ввода графической информации, цифровые учебники.

Для достижения поставленных целей преподавания дисциплины реализуются следующие средства, способы и организационные мероприятия:

- изучение теоретического материала дисциплины на лекциях с использованием компьютерных технологий;
- самостоятельное изучение теоретического материала дисциплины с использованием Internet-ресурсов, специальной учебной и научной литературы, а также с помощью электронных учебных курсов в системе Moodle;
- закрепление теоретического материала при проведении лабораторных работ с использованием современной вычислительной техники и пакетов прикладных программ MS Office, выполнения проблемно-ориентированных, поисковых, творческих заданий.

От общего количества аудиторных занятий доля лекционных учебных занятий составляет 0,5%.

8. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

Допуск к лабораторным работам и их защита

В течение семестра студенты выполняют лабораторные работы. Каждый студент должен заранее подготовиться к выполнению лабораторной работы, т.е. теоретически ознакомиться с предстоящим заданием или хотя бы иметь общее представление о том, что необходимо будет сделать на занятии.

Если базовые знания в теме лабораторной работы и вовсе отсутствуют, преподаватель имеет полное право не допустить к практической части занятия. И тогда пересдавать лабораторную работу будет необходимо в индивидуальном порядке.

Защита проделанной работы осуществляется в индивидуальном порядке даже тогда, когда задание было выполнено коллективно. У каждого студента должен быть свой оформленный отчет и минимальный багаж знаний по теме, которым в ближайшее время и придется поделиться с преподавателем.

Указания и требования по оформлению, выполнению и сдачи лабораторных работ (допуск и защита) вынесены в отдельное методическое пособие по дисциплине «Программирование».

Демонстрационный вариант теста рубежного контроля

По каждому модулю студенты должны пройти тестирование, которое проводится с помощью системы управления курсами – Moodle. Примерные вопросы теста:

1.	Microsoft.NET (.NET Framework)	1) – программная платформа. 2) – общая спецификация языков программирования. 3) – среда Времени Выполнения или Виртуальная Машина.
2.	CLS (Common Language Specification)	1) – программная платформа. 2) – общая спецификация языков программирования. 3) – среда Времени Выполнения или Виртуальная Машина.
3.	CLR (Common Language Runtime)	1) – программная платформа. 2) – общая спецификация языков программирования. 3) – среда Времени Выполнения или Виртуальная Машина.
4.	Классы для обращения к базам данных	1) System.Data, System.Data.Common, System.Data.OleDb, System.Data.SqlClient

		2) System.Collections 3) System.Diagnostics 4) System.Drawing , System.Drawing.Drawing2D, System.Drawing.Printing
5.	Классы для работы с контейнерными объектами	1) System.Data, System.Data.Common, System.Data.OleDb, System.Data.SqlClient 2) System.Collections 3) System.Diagnostics 4) System.Drawing , System.Drawing.Drawing2D, System.Drawing.Printing
6.	Классы для трассировки и отладки кода	1) System.Data, System.Data.Common, System.Data.OleDb, System.Data.SqlClient 2) System.Collections 3) System.Diagnostics 4) System.Drawing , System.Drawing.Drawing2D, System.Drawing.Printing
7.	Классы графической поддержки	1) System.Data, System.Data.Common, System.Data.OleDb, System.Data.SqlClient 2) System.Collections 3) System.Diagnostics 4) System.Drawing , System.Drawing.Drawing2D, System.Drawing.Printing
8.	Поддержка ввода-вывода	1) System.IO 2) System.Net 3) System.Reflection , System.Reflection.Emit

		4) System.Runtime.InteropServices, System.Runtime.Remoting
9.	Поддержка передачи данных по сетям	1) System.IO 2) System.Net 3) System.Reflection , System.Reflection.Emit 4) System.Runtime.InteropServices, System.Runtime.Remoting
10.	Работа с пользовательскими типами во время выполнения приложения	1) System.IO 2) System.Net 3) System.Reflection , System.Reflection.Emit 4) System.Runtime.InteropServices, System.Runtime.Remoting
11.	Поддержка взаимодействия с “обычным кодом” – DLL, COM-серверы, удалённый доступ	1) System.IO 2) System.Net 3) System.Reflection , System.Reflection.Emit 4) System.Runtime.InteropServices, System.Runtime.Remoting
12.	Криптография, разрешения	1) System.Security 2) System.Threading 3) System.WEB 4) System.Windows.Form
13.	Работа с потоками	1) System.Security 2) System.Threading 3) System.WEB 4) System.Windows.Form
14.	Работа с web-приложениями	1) System.Security 2) System.Threading

		3) System.WEB 4) System.Windows.Form
15.	Работа с элементами интерфейса Windows	1) System.Security 2) System.Threading 3) System.WEB 4) System.Windows.Form
16.	Поддержка данных в формате XML	1) System.Threading 2) System.WEB 3) System.Windows.Form 4) System.XML
17.	Целый. 8-разрядное со знаком. Диапазон значений: 128 ... 127	1) sbyte 2) byte 3) short 4) ushort
18.	Целый. 8-разрядное без знака. Диапазон значений: 0 ... 255	1) sbyte 2) byte 3) short 4) ushort
19.	Целый. 16-разрядное со знаком. Диапазон значений: -32768 ... 32767	1) sbyte 2) byte 3) short 4) ushort
20.	Целый. 16-разрядное без знака. Диапазон значений: 0 ... 65535	1) sbyte 2) byte 3) short 4) ushort
21.	Целый. 32-разрядное со знаком. Диапазон значений: -2147483648 ... 2147483647	1) int

		2) uint 3) long 4) ulong
22.	Целый. 32-разрядное без знака. Диапазон значений: 0 ... 4294967295	1) int 2) uint 3) long 4) ulong
23.	Целый. 64-разрядное со знаком. Диапазон значений: - 9223372036854775808 ... 9223372036854775807	1) int 2) uint 3) long 4) ulong
24.	Целый. 64-разрядное без знака. Диапазон значений: 0 ... 18446744073709551615	1) int 2) uint 3) long 4) ulong
25.	16 (!) разрядный символ UNICODE.	1) char 2) float 3) double 4) decimal
26.	Плавающий. 32 разряда. Стандарт IEEE.	1) char 2) float 3) double 4) decimal
27.	Плавающий. 64 разряда. Стандарт IEEE.	1) char 2) float 3) double 4) decimal

28.	128-разрядное значение повышенной точности с плавающей точкой.	1) char 2) float 3) double 4) decimal
29.	Значение true или false.	1) sbyte 2) byte 3) short 4) bool
30.	Модификаторы прав доступа: обозначение для общедоступных членов класса. К ним можно обратиться из любого метода любого класса программы.	1) public 2) protected 3) internal 4) protected internal
31.	Модификаторы прав доступа: обозначение для членов класса, доступных в рамках объявляемого класса и из методов производных классов.	1) public 2) protected 3) internal 4) protected internal
32.	Модификаторы прав доступа: обозначение для членов класса, доступных из методов классов, объявляемых в рамках сборки, содержащей объявление данного класса.	1) public 2) protected 3) internal 4) protected internal
33.	Модификаторы прав доступа: обозначение для членов класса, доступных в рамках объявляемого класса и из методов производных классов, а также доступных из методов классов, объявляемых в рамках сборки, содержащей объявление данного класса.	1) public 2) protected 3) internal 4) protected internal
34.	Модификаторы прав доступа: обозначение для членов класса, доступных в рамках объявляемого класса.	1) protected 2) internal 3) protected internal

		4) private
35.	Escape-последовательность: Предупреждение (звонок)	1) \a 2) \b 3) \f 4) \n
36.	Escape-последовательность: Возврат на одну позицию	1) \a 2) \b 3) \f 4) \n
37.	Escape-последовательность: Переход на новую страницу	1) \a 2) \b 3) \f 4) \n
38.	Escape-последовательность: Переход на новую строку	1) \a 2) \b 3) \f 4) \n
39.	Escape-последовательность: Возврат каретки	1) \r 2) \t 3) \v 4) \0
40.	Escape-последовательность: Горизонтальная табуляция	1) \r 2) \t 3) \v 4) \0
41.	Escape-последовательность: Вертикальная табуляция	1) \r 2) \t

		3) \v 4) \0
42.	Escape-последовательность: Ноль	1) \r 2) \t 3) \v 4) \0
43.	Escape-последовательность: Одинарная кавычка	1) \0 2) \' 3) \" 4) \\
44.	Escape-последовательность: Двойная кавычка	1) \0 2) \' 3) \" 4) \\
45.	Escape-последовательность: Обратная косая черта	1) \0 2) \' 3) \" 4) \\
46.	Общие характеристики используемых в C# операций: + - * / %	1) Arithmetic 2) Логические (boolean и побитовые) 3) Строковые (конкатенаторы) 4) Increment, decrement
47.	Общие характеристики используемых в C# операций: & ^ ! ~ &&	1) Arithmetic 2) Логические (boolean и побитовые) 3) Строковые (конкатенаторы) 4) Increment, decrement
48.	Общие характеристики используемых в C# операций: +	1) Arithmetic

		2) Логические (boolean и побитовые) 3) Строковые (конкатенаторы) 4) Increment, decrement
49.	Общие характеристики используемых в C# операций: ++ --	1) Arithmetic 2) Логические (boolean и побитовые) 3) Строковые (конкатенаторы) 4) Increment, decrement
50.	Общие характеристики используемых в C# операций: >> <<	1) Сдвига 2) Сравнения 3) Присвоения 4) Member access
51.	Общие характеристики используемых в C# операций: == != < > <= >=	1) Сдвига 2) Сравнения 3) Присвоения 4) Member access
52.	Общие характеристики используемых в C# операций: = += -= *= /= %= &= = ^= <<= >>=	1) Сдвига 2) Сравнения 3) Присвоения 4) Member access
53.	Общие характеристики используемых в C# операций: .	1) Сдвига 2) Сравнения 3) Присвоения 4) Member access
54.	Общие характеристики используемых в C# операций: []	1) Индексации 2) Cast (приведение типа) 3) Conditional (трёхоперандная) 4) Delegate concatenation and removal

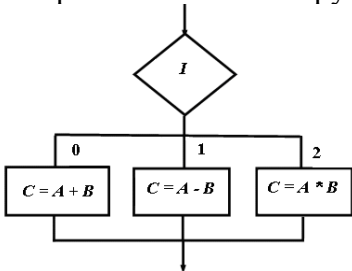
55.	Общие характеристики используемых в C# операций: ()	1) Индексации 2) Cast (приведение типа) 3) Conditional (трёхоперандная) 4) Delegate concatenation and removal
56.	Общие характеристики используемых в C# операций: ?:	1) Индексации 2) Cast (приведение типа) 3) Conditional (трёхоперандная) 4) Delegate concatenation and removal
57.	Общие характеристики используемых в C# операций: + -	1) Индексации 2) Cast (приведение типа) 3) Conditional (трёхоперандная) 4) Delegate concatenation and removal
58.	Общие характеристики используемых в C# операций: new()	1) Создания объекта 2) Type information 3) Overflow exception control (управление исключениями) 4) Indirection and Address (неуправляемый код)
59.	Общие характеристики используемых в C# операций: is sizeof typeof	1) Создания объекта 2) Type information 3) Overflow exception control (управление исключениями) 4) Indirection and Address (неуправляемый код)
60.	Общие характеристики используемых в C# операций: checked unchecked	1) Создания объекта 2) Type information 3) Overflow exception control (управление исключениями) 4) Indirection and Address (неуправляемый код)

61.	Общие характеристики используемых в C# операций: * -> [] &	1) Создания объекта 2) Type information 3) Overflow exception control (управление исключениями) 4) Indirection and Address (неуправляемый код)
62.	Операторы выбора. Вводятся ключевыми словами:	1) if, if ... else ..., switch; 2) while, do ... while, for, foreach; 3) goto, break, continue;
63.	Итеративные операторы. Вводятся ключевыми словами:	1) if, if ... else ..., switch; 2) while, do ... while, for, foreach; 3) goto, break, continue;
64.	Операторы перехода (в рамках методов). Вводятся ключевыми словами:	1) if, if ... else ..., switch; 2) while, do ... while, for, foreach; 3) goto, break, continue;
65.	Какие из перечисленных языков подходят для программирования?	1. PHP 2. Ассемблер 3. C, C++ 4. Java
66.	Не является принципом объектно-ориентированного программирования...	1. инкапсуляция 2. наследование 3. полиморфизм 4. использование функций
67.	В объектно-ориентированном программировании каждый объект по отношению к своему классу является...	1. кодом 2. экземпляром 3. функцией 4. фрагментом
68.	Дана блок-схема алгоритма. Если начальные значения переменных A, B и C равны 3, 3 и 1 соответственно, то значение переменной F будет равно ...	1. 6 2. 8 3. 7 4. 1

	<pre> graph TD Start([Ввод A, B, C]) --> Cond1{A = B} Cond1 -- нет --> Proc1[C = A + C] Proc1 --> Proc2[F = B - C] Cond1 -- да --> Cond2{B > C} Cond2 -- нет --> Proc3[A = A + B] Proc3 --> Proc4[F = A + C] Cond2 -- да --> Proc5[B = B + C] Proc5 --> Proc6[F = A + B] Proc2 --> End([Вывод F]) Proc4 --> End Proc6 --> End </pre>	
69.	<p>В приведенном фрагменте блок-схемы выполняется...</p> <pre> graph TD Start([Ввод A, B]) --> Proc1[C = A] Proc1 --> Proc2[A = B] Proc2 --> Proc3[B = C] Proc3 --> End([Вывод A, B]) </pre>	<ol style="list-style-type: none"> 1. обмен значениями переменных A и C 2. обмен значениями переменных A и B 3. сравнение переменных A, B, C 4. обмен значениями переменных B и C
70.	<p>Фрагмент блок-схемы</p> <pre> graph TD Cond1{Условие 1} -- да --> Proc1[Серия 1] Cond1 -- нет --> Cond2{Условие 2} Cond2 -- да --> Proc2[Серия 2] Cond2 -- нет --> Proc3[Серия 3] Proc1 --> End([]) Proc2 --> End Proc3 --> End </pre> <p>представляет собой алгоритм, который содержит ...</p>	<ol style="list-style-type: none"> 1. две команды ветвления в полной форме, одна из которых вложена в другую 2. команду ветвления в полной форме, в которую вложена команда ветвления в неполной форме 3. команду ветвления в полной форме и команду выбора, вложенную в команду ветвления 4. команду ветвления в полной форме, в которую вложена команда цикла
71.	<p>Данный алгоритм For(i=1;i<=10;i++) { A[i]:=10 - i + 1; } формирует массив A из 10 элементов вида:</p>	<ol style="list-style-type: none"> 1. (10, 9, 8, 7, 6, 5, 4, 3, 2, 1) 2. (9, 8, 7, 6, 5, 4, 3, 2, 1, 0) 3. (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) 4. (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

72.	В заданном фрагменте алгоритма обработки массива Matrix размерностью 10 на 10 S := 0; For(a=1;a<=10;a++) { S := S + Matrix[1, a]; } определяется сумма элементов, расположенных	1. Во всех ячейках матрицы 2. На главной диагонали матрицы 3. В первом столбце матрицы 4. В первой строке матрицы
73.	В алгоритме, определяющем количество отрицательных элементов массива A размерностью N S := 0; For J := 1 To N do If _____ Then S := S + 1; S := N - S; пропущен фрагмент	1. A[J] > S 2. A[J] >= 0 3. A[J] <= S 4. A[J] < 0
74.	Процедуры и функции — это:	1) операторы; 2) подпрограммы; 3) имена; 4) переменные.
75.	Файл — это:	1) база данных; 2) поименованный участок на внешних носителях памяти; 3) список; 4) раздел на жестком диске.
76.	Упорядоченный тип — это:	1) тип переменной, значения которой упорядочены в обычном смысле; 2) запись; 3) целые и вещественные; 4) значения переменных такого типа находятся в порядке, случайно выбранном.
77.	Выражение — это:	1) конструкция языка, значение которой может меняться; 2) текст программы, заключенный в операторные скобки; 3) множество символов, которые являются упорядоченными; 4) конструкция, задающая правила вычисления значений переменных.
78.	Рекурсия — это:	1) повторение выполнения функции или процедуры внутри себя; 2) оператор; 3) цикл; 4) метод определения функции или процедуры.
79.	Массив — это:	1) запись множества переменных разного типа; 2) неупорядоченная совокупность отличных

		<p>друг от друга однотипных элементов;</p> <p>3) последовательность, состоящая из фиксированного числа однотипных элементов;</p> <p>4) тип одномерных величин.</p>
80.	Множество — это:	<p>1) список элементов, заключенный в круглые скобки, вида: <имя поля>:<значение>;</p> <p>2) неупорядоченная совокупность отличных друг от друга однотипных элементов;</p> <p>3) совокупность с фиксированным числом однотипных элементов, отличных только индексами;</p> <p>4) совокупность отличных друг от друга элементов разных типов.</p>
81.	Запись — это:	<p>1) константное значение;</p> <p>2) последовательность, состоящая из фиксированного числа однотипных элементов;</p> <p>3) последовательность однотипных элементов, отличающихся индексами;</p> <p>4) последовательность, состоящая из фиксированного числа величин, называемых полями.</p>
82.	Глобальные переменные действуют:	<p>1) во всех процедурах;</p> <p>2) во всех функциях;</p> <p>3) во всех модулях;</p> <p>4) во всей программе.</p>
83.	Какие данные относятся к динамическим:	<p>1) записи, файлы, множества, массивы;</p> <p>2) целые числа, действительные числа;</p> <p>3) логические, символьные;</p> <p>4) списки, стеки, графы, деревья.</p>
84.	Сортировка — это:	<p>1) процесс нахождения в заданном множестве объекта;</p> <p>2) процесс перегруппировки заданного множества объектов в некотором порядке;</p> <p>3) установка индексов элементов в возрастающем порядке;</p> <p>4) обработка элементов в алфавитном порядке.</p>
85.	Объединение в одном объекте данных и методов их обработки называется:	<p>1) наследованием</p> <p>2) полиморфизмом</p> <p>3) инкапсуляцией</p> <p>4) виртуализацией</p>

86.	«Не равно» в языке СИ обозначается:	1) < >; 2) ; 3) !; 4) !=;
87.	Логическое «и» в языке СИ обозначается:	1) =; 2) ; 3) &; 4) &&.
88.	Логическое «или» в языке СИ обозначается:	1) =; 2) ; 3) &; 4) &&.
89.	Логическое «не» в языке СИ обозначается:	1) !; 2) !!; 3) ; 4) not.
90.	Точка с запятой в языке СИ является:	1) разделителем операторов; 2) частью оператора; 3) ключевым знаком языка Си; 4) спецсимвол.
91.	В языке Си тело функции ограничено операторными скобками:	1) begin end; 2) start finish; 3) []; 4) {}.
92.	Укажите последовательность команд, в результате выполнения которых значения переменных x и y поменяются местами.	1. y:=x; b:=x; x:=y; 2. c:=x; x:=y; x:=c; 3. x:=x+y; y:=x-y; x:=x-y; 4. b:=x; x:=y; y:=x;
93.	На блок-схеме представлена алгоритмическая конструкция 	1. Ветвление 2. Цикл с предусловием 3. Цикл с повторением 4. Выбор
94.	Допустимыми именами переменных являются	1. ?Gross 2. Sor1_t1_12 3. M O D 4. SchwarzKopf
95.	Выражению $S = \frac{XY}{ABC}$ соответствует оператор присваивания	1. S := X * Y / (A * B * C); 2. S := (X * Y) / A / B / C; 3. S := X * Y / A * B * C; 4. S := X / A * B / C * Y;

96.	После выполнения фрагмента программы с двойным циклом k := 1; For i := 1 To 5 do For j := i To 5 do k := k + 1; значение переменной k равно ...	1. 16 2. 25 3. 26 4. не определено
97.	Что называют операторными скобками в Pascal?	1) (); 2) { }; 3) begin..end; 4) [].
98.	Над логическими величинами определены операции:	1) +, -, *, /; 2) not, and, or, odd(); 3) sin(), cos(), tg(), abs(); 4) trunc(), round (), ord().
99.	Каков будет результат выполнения программы: Var n, i: integer; Begin n:=0; For i:=1 to 10 do n:=n+i end.	1) 55; 2) 0; 3) 5; 4) 25.
100	Создание новых классов на базе ранее определённых, при котором объекты-потомки сохраняют свойства своих родителей, называется:	1) наследованием 2) полиморфизмом 3) инкапсуляцией 4) методом

Примерный перечень вопросов и заданий к экзамену

Промежуточная аттестация по дисциплине – экзамен. Студенты сдают экзамен по билетам, которые состоят из теоретической и практической части.

Для подготовки к теоретической части экзамена студент может ориентироваться на следующие примерные вопросы, которые могут встречаться в экзаменационных билетах:

1. Тенденции развития программного обеспечения вычислительной техники.
2. Понятие алгоритма. Свойства алгоритмов.
3. Формы записей алгоритмов.
4. Общие принципы построения алгоритмов.
5. Основные алгоритмические конструкции: линейные, разветвляющиеся, циклические. Данные: понятие и типы. Основные базовые типы данных и их характеристика. Структурированные типы данных и их характеристика.
6. Методы сортировки данных
7. Алгебра логики. Логические операции.

8. Эволюция языков программирования.
9. Классификация языков программирования.
10. Элементы языков программирования.
11. Понятие системы программирования.
12. Исходный, объектный и загрузочный модули.
13. Интегрированная среда программирования.
14. Методы программирования: структурный, модульный, объектно-ориентированный. Достоинства и недостатки методов программирования.
15. Общие принципы разработки программного обеспечения.
16. Жизненный цикл программного обеспечения.
17. Типы приложений. Консольные приложения. Оконные Windows приложения. Web-приложения. Библиотеки. Web-сервисы.
18. История развития языка программирования C#.
19. Структурная схема программы на алгоритмическом языке C#.
20. Лексика языка C#.
21. Переменные и константы C#.
22. Типы данных C#.
23. Выражения и операции C#.
24. Самостоятельная работа обучающихся
25. Синтаксис операторов присваивания, ввода-вывода C#.
26. Синтаксис операторов безусловного и условного переходов C#.
27. Синтаксис операторов цикла C#.
28. Составной оператор C#.
29. Вложенные условные операторы C#.
30. Цикл с параметром C#.
31. Циклы с предусловием и постусловием C#.
32. Массивы как структурированный тип данных.
33. Объявление массивов C#.
34. Ввод и вывод одномерных массивов в C#.
35. Ввод и вывод двумерных массивов в C#.
36. Обработка массивов в C#.
37. Стандартные функции для массива целых и вещественных чисел C#.
38. Структурированные типы данных: строки, множества, структуры.
39. Объявление строковых типов данных C#.
40. Поиск, удаление, замена и добавление символов в строке в C#.
41. Операции со строками C#.
42. Стандартные функции и процедуры для работы со строками C#.
43. Объявление множества. Операции над множествами.
44. Структуры.
45. Понятие подпрограммы.
46. Процедуры и функции, их сущность, назначение, различие.
47. Организация процедур, стандартные процедуры.
48. Процедуры, определенные пользователем: синтаксис, передача аргументов. Формальные и фактические параметры.

- 49.Процедуры с параметрами, описание процедур.
- 50.Функции: способы организации и описание.
- 51.Вызов функций, рекурсия.
- 52.Программирование рекурсивных алгоритмов.
- 53.Стандартные функции C#.
- 54.Типы файлов.
- 55.Организация доступа к файлам.
- 56.Файлы последовательного доступа.
- 57.Открытие и закрытие файла последовательного доступа.
- 58.Запись в файл и чтение из файла последовательного доступа.
- 59.Исключения. Обработка исключений.
- 60.Файлы произвольного доступа.
- 61.Порядок работы с файлами произвольного доступа. Создание структуры записи. Открытие и закрытие файла произвольного доступа.
- 62.Запись и считывание из файла произвольного доступа.
- 63.Использование файла произвольного доступа.
- 64.Стандартные процедуры и функции для файлов разного типа.
- 65.Программирование модулей. Модуль: синтаксис, заголовок, разделы.
- 66.Библиотеки подпрограмм: понятие и виды.
- 67.Схемы вызова библиотек
- 68.Статическое и динамическое связывание.
- 69.Использование библиотек подпрограмм.
- 70.История развития ООП.
- 71.Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.
- 72.Основные принципы ООП: инкапсуляция, наследование, полиморфизм.
- 73.Событийно-управляемая модель программирования.
- 74.Компонентно-ориентированный подход.
- 75.Компоненты и их свойства.
- 76.Классы объектно-ориентированного языка программирования: виды, назначение, свойства, методы, события.
- 77.Объявление класса, свойств и методов экземпляра класса. Наследование. Перегрузка методов.
- 78.Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.
- 79.Дополнительные элементы управления.
- 80.Свойства компонентов (элементов управления). Виды свойств. Синтаксис определения свойств. Категория свойств. Назначение свойств и их влияние на результат. Управление объектом через свойства.
- 81.События компонентов (элементов управления), их сущность и назначение.
- 82.Создание процедур на основе событий.
- 83.Процедуры, определенные пользователем: синтаксис, передача аргументов.
- 84.Разработка функционального интерфейса приложения. Создание

интерфейса.

85. Меню и панели инструментов: ContainerMenuStrip, MenuStrip, StatusStrip, ToolStrip. Использование командных кнопок.
86. Использование меток и полей ввода: Label, RichTextBox, TextBox.
87. Стандартные элементы управления: CheckBox, RadioButton, ListBox, ComboBox. Использование TreeView и ListView.
88. Использование стандартных диалогов.
89. Использование PictureBox.
90. Вывод текста в графическом режиме.

Для подготовки к практической части экзамена студент может ориентироваться на следующие примерные задачи, которые могут встречаться в экзаменационных билетах:

1. Найти сумму четных делителей натурального числа.
2. Найти все четырехзначные числа, в которых есть две одинаковые цифры.
3. Из данного натурального числа удалить все цифры A (A вводится с клавиатуры).
4. Задан двумерный массив вещественных чисел. Необходимо каждый элемент соответствующей строки разделить на сумму элементов этой строки. Размерность массива (число столбцов и строк) и значения его элементов ввести с клавиатуры.
5. Задан двумерный массив вещественных чисел. Заменить все отрицательные элементы нулями и подсчитать сумму всех положительных элементов. Размерность массива (число столбцов и строк) и значения его элементов ввести с клавиатуры.
6. Задан двумерный массив вещественных чисел размерности 4x4. Найти номер строки и столбца, в которых находится наименьший элемент.
7. Задан двумерный массив вещественных чисел размерности 5x5. Определить сумму элементов главной диагонали.
8. Найти все натуральные числа a, b и c из интервала от 1 до 20, для которых выполняется равенство $a^2 + b^2 = c^2$.
9. Найти все натуральные числа из промежутка от 1 до 200, у которых количество делителей равно N (N вводить с клавиатуры).
10. Найти все четырехзначные числа, у которых сумма крайних цифр равна сумме средних цифр.

Критерии оценки

Критерии освоения дисциплины на экзамене:

Оценка «отлично».

Студент умеет увязывать теорию с практикой (решает задачи и формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, обосновывает свои суждения и даёт правильные ответы на вопросы преподавателя.

Оценка «хорошо».

Студент умеет увязывать теорию с практикой (решает задачи и формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, но содержание ответов имеют некоторые неточности и требуют уточнения и комментария со стороны преподавателя.

Оценка «удовлетворительно».

Студент знает и понимает материал по заданной теме, но изложение неполное, непоследовательное, допускаются неточности в определении понятий, решение задач с ошибками, студент не может обосновать свои ответы на уточняющие вопросы преподавателя.

Оценка «неудовлетворительно».

Студент допускает ошибки при решении задач и в определении понятий, искажающие их смысл, беспорядочно и неуверенно излагает материал. Делает ошибки в ответах на уточняющие вопросы преподавателя.

9. Организация образовательного процесса для лиц с ограниченными возможностями

Организация образовательного процесса для лиц с ограниченными возможностями осуществляется в соответствии с «Методическими рекомендациями по организации образовательного процесса для инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе оснащенности образовательного процесса» Министерства образования и науки РФ от 08.04.2014г. № АК-44/05вн.

В образовательном процессе используются социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе.

Студенты с ограниченными возможностями здоровья, в отличие от остальных студентов, имеют свои специфические особенности восприятия, переработки материала. Подбор и разработка учебных материалов производится с учетом индивидуальных особенностей.

Предусмотрена возможность обучения по индивидуальному графику, при составлении которого возможны различные варианты проведения занятий: в академической группе и индивидуально, на дому с использованием дистанционных образовательных технологий.

10. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. Голицина О. Л., Попов И. И. Основы алгоритмизации и программирования: Учебное пособие. – М.: ФОРУМ: ИНФРА-М, 2005.

2. Колдаев В. П. Основы алгоритмизации и программирования: Учебное пособие./ Под. ред. проф. Л. Г. Гагариной. – М.: ИД «ФОРУМ» ИНФРА-М, 2006.
3. Павловская Т. А. С#. Программирование на языке высокого уровня: Учебник для вузов. - СПб.: БХВ-Петербург, 2007.
4. Попов И. И., Партыка Т. Л. Языки программирования: Учебное пособие. – М.: ФОРУМ: ИНФРА-М, 2008.
5. Фаронов В. В., Создание приложений с помощью С#: Руководство программиста. – М.: Эксмо, 2008.

б) дополнительная литература:

1. Нортроп Тони, Уилдермьюс Шон, Райан Билл. Основы разработки приложений на платформе Microsoft .Net Framework. Учебный курс Microsoft / Пер. с англ. – М.: «Русская редакция», СПб. : «Питер», 2007.
2. Павловская Т. А. С/C++ Программирование на языке высокого уровня. – СПб.: Питер, 2005.
3. Сverdlov С. З. Языки программирования и методы трансляции. Учебное пособие. – СПб.: Питер, 2007.
4. Семакин И. Г., Шестаков А. П. Основы программирования: Учебник. – М.: Мастерство; НМЦ СПО; Высшая школа, 2001.

в) программное обеспечение

- пакет MS Office 2010;
- среда программирования MS Visual Studio 2010.

в) мультимедийные средства:

- проектор, специализированное программное обеспечение.

г) Интернет-ресурсы

- <http://www.edu.ru> – Российское образование: Федеральный портал;
- <http://www.office.microsoft.com/ru-ru/training> – изучение приложений Office при помощи учебных курсов для самостоятельного обучения;
- <http://do.vfmgtu.ru> – система дистанционного образования Moodle (самостоятельная учебная деятельность студентов);
- http://professorweb.ru/my/csharp/charp_theory/level1/index.php - изучение языка программирования С#;
- электронный каталог библиотек, который располагается на сайте регионального института <http://www.vfmgtu.ru/>;
- электронная библиотека в системе дистанционного обучения MOODLE <http://do.vfmgtu.ru/>;
- удаленные полнотекстовые ресурсы:
- а.) ЭБС «Издательство Лань» <http://e.lanbook.com> (соглашение о сотрудничестве от 21.01.2015г.);

- б.) ЭБС «Национальный цифровой ресурс «Руконт» <http://rucont.ru/> (договор от 13.01.2015 № 18/2222 - 2015);
- в.) ЭБС (ЭБС) Znanium издательства «Инфра – М».
- консорциум НП «НЕИКОН» www.neicon.ru/cons (соглашение о сотрудничестве от 01.03.2012г. № 741-ДС-2011);
- справочно-поисковые системы компаний «Консультант Плюс» <http://www.consultant.ru/>.

11. Материально-техническое обеспечение дисциплины

Для преподавания дисциплины «Программирование» предоставляется три компьютерных класса для проведения лабораторных работ. Компьютерные классы оснащены персональными ЭВМ, которые объединены в локальную компьютерную сеть института и имеют выход в глобальную сеть Интернет. На всех компьютерах установлена операционная система Windows 7.

Компьютеры имеют следующие характеристики:

- компьютерный класс №1312 (15 компьютеров): процессор Intel Original LGA-1155 Pentium G840, оперативная память – 2048 Mb DDR3, жёсткий диск – 500 Gb SATA-III Hitachi;
- компьютерный класс №1310 (15 компьютеров): процессор – Intel Pentium Sandy Bridge G860, оперативная память – DIMM DDR 2 Gb, жёсткий диск – 250 Gb Seagate;
- компьютерный класс №1302 (10 компьютеров): процессор Intel Original LGA-1155 Pentium G840, оперативная память – 2048 Mb DDR3, жёсткий диск – 500 Gb SATA-III Hitachi.

11. Лист регистрации изменений

№ п/п	Содержание изменения	Реквизиты документа об утверждении изменения	Дата введения изменения
1.	Утверждена и введена в действие решением кафедры «Математика, физика и информационные технологии» на основании федерального государственного образовательного стандарта высшего образования по направлению подготовки 15.03.04 «Автоматизация технологических процессов и производств», профиль подготовки «Автоматизация технологических процессов и производств» (уровень бакалавриата), утвержденного приказом Министерства образования и науки Российской Федерации от 12.03.2015 г № 200	Протокол заседания кафедры № 1 от «29» августа 2017 года	