

Балаковский инженерно-технологический институт – филиал федерального государственного автономного образовательного учреждения высшего образования  
«Национальный исследовательский ядерный университет «МИФИ»

Факультет атомной энергетики и технологий  
Кафедра «Информационные системы и технологии»

## **РАБОЧАЯ ПРОГРАММА**

по дисциплине

«Современные среды визуального программирования»

**Направления подготовки/специальность**

«09.03.02 Информационные системы и технологии»

**Основная профессиональная образовательная программа**

«Информационные системы и технологии»

**Квалификация выпускника**

Бакалавр

**Форма обучения**

Очная

## Цель освоения дисциплины

Целями освоения дисциплины «Современные среды визуального программирования» в соответствии с общими целями ООП ВО являются: формирование культуры исследовательской и инженерной деятельности; формализация и алгоритмизация поставленных задач; написание программного кода с использованием языков программирования, определения и манипулирования данными; проверка и отладка программного кода; проверка работоспособности программного обеспечения; анализ требований к программному обеспечению.

## Место дисциплины в структуре ООП ВПО

Необходимыми условиями для освоения дисциплины являются знания, умения и практические навыки по предшествующим дисциплинам и практикам:

Языки программирования

Учебная практика (по получению первичных профессиональных умений и навыков)

Знания, умения и навыки, приобретенные студентами в процессе изучения дисциплины, в дальнейшем используются во время изучения дисциплин:

Интеллектуальные системы и технологии

Современные технологии интернет-программирования

Современные технологии управления базами данных

Современные технологии объектно-ориентированного программирования

Объектно-ориентированное программирование

Архитектура информационных систем

Инструментальные средства информационных систем

Технологии программирования

Экономика

Интерактивное программирование web-приложений

Создание интерактивных приложений для интернет

Государственная итоговая аттестация

## Компетенции обучающегося, формируемые в результате освоения дисциплины

В процессе освоения данной дисциплины у студента формируются следующие компетенции:

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	З-ОПК-6 Знать: языки и среды программирования; библиотеки программных модулей; шаблоны, классы объектов, используемые при разработке программного обеспечения У-ОПК-6 Уметь: создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов; использовать выбранную среду программирования для написания программного кода

		В-ОПК-6 Владеть: языками и средами программирования для разработки алгоритмов и программ
УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	3-УК-2 Знать: виды ресурсов и ограничений для решения профессиональных задач; основные методы оценки разных способов решения задач; действующее законодательство и правовые нормы, регулирующие профессиональную деятельность У-УК-2 Уметь: проводить анализ поставленной цели и формулировать задачи, которые необходимо решить для ее анализировать альтернативные решений для достижения результатов; использовать нормативно-правовую документацию в сфере профессиональной деятельности В-УК-2 Владеть: методиками разработки цели и задач проекта; методами оценки потребности в ресурсах, продолжительности и стоимости проекта, навыками работы с нормативно-правовой документацией

### Задачи воспитания, реализуемые в рамках освоения дисциплины

Направление/цели	Создание условий, обеспечивающих	Использование воспитательного потенциала учебной дисциплины	Вовлечение в разноплановую внеучебную деятельность
Профессиональное воспитание	формирование культуры исследовательской и инженерной деятельности <b>(В16)</b>	Использование воспитательного потенциала дисциплин профессионального модуля для формирования культуры исследовательской и инженерной деятельности за счёт практических студенческих исследований современных производственных систем; проектной деятельности студентов по разработке и оптимизации технологических систем, связанной с решением реальных производственных задач; прохождения через разнообразные игровые, творческие, фестивальные формы, требующие анализа сложного объекта, постановки относительно него преобразовательных задач для их оптимального решения.	1. Организация научно-практических конференций и встреч с ведущими специалистами предприятий города и ветеранами атомной отрасли. 2. Организация и проведение предметных олимпиад и участие в конкурсах профессионального мастерства.

### Структура и содержание учебной дисциплины

Дисциплина преподается студентам в 3-ем семестре. Общая трудоемкость дисциплины составляет 3 зачетных единицы, 108 ак. часа.

## Календарный план

№ Р а з д е л а	№ Т е м ы	Наименование раздела (темы) дисциплины	Виды учебной деятельности (в часах)					Аттестац ия раздела (форма)	Макси маль ный балл за раздел
			Всего	Лекции	Лабораторные	Практические	СРС		
1	<b>Основные понятия среды визуального программирования</b>							КИ	25
	1	Введение. Основные понятия визуального программирования	16	2	-	4	10		
	2	Стандартные элементы управления	18/2	2	-	6/2	10		
	3	Меню в панели инструментов	18/4	4/2	-	4/2	10		
2	<b>Элементы Forms</b>							КИ	25
	4	Диалоговые окна и компоненты для вывода на печать	18/4	4/2	-	6/2	10		
	5	Контейнеры и дополнительные компоненты	18/4	2/2	-	6/2	10		
	6	Компоненты, предназначенные для работы с базами данных	20/2	2	-	6/2	10		
<b>Вид промежуточной аттестации</b>								<b>3</b>	<b>50</b>
<b>Итого</b>			<b>108/ 16</b>	<b>16/6</b>	<b>-</b>	<b>32/10</b>	<b>60</b>		<b>100</b>

КИ- контроль по итогам

\* - занятия в интерактивной форме

Сокращенное наименование форм текущего контроля и аттестации разделов:

Обозначение	Полное наименование
КИ	Контроль по итогам
3	Зачет

## Содержание лекционного курса

Темы лекции. Вопросы, отрабатываемые на лекции	Всего часов	Учебно-методическое обеспечение
1	2	3
Лекция 1. Введение. Основные понятия визуального программирования. 1. Определение СВП. 2. Достоинства и недостатки СВП. 3. Понятие объекта, свойств, методов и событий объектов. 4. Понятие компонентов СВП 5. Способы изменения свойств объекта.	2	1-3
Лекция 2. Стандартные элементы управления. Форма (Form) 1. Основные свойства, методы и события. Надпись (Label): виды, основные свойства, методы и события. Поле ввода (TextBox): основные свойства, методы и события. Управляющая кнопка (Button): основные свойства, методы и события. Флажок (CheckBox): основные свойства, методы и события. Список флажков (CheckedListBox): основные свойства, методы и события. Поле со списком (ComboBox): основные свойства, методы и события. Компонент для выбора даты и времени (DateTimePicker): основные свойства, методы и события. 2. Гиперссылка (LinkLabel): основные свойства, методы и события.	2	1-3

<p>Список элементов (ListBox): основные свойства, методы и события.  Список окон (ListView): основные свойства, методы и события. Поле ввода с маской ввода (MaskedTextBox): основные свойства, методы и события. Календарь (MonthCalendar): основные свойства, методы и события. Значок в области уведомлений (NotifyIcon): основные свойства, методы и события. Значок (NumericUpDown): основные свойства, методы и события. Окно изображения (PictureBox): основные свойства, методы и события.</p> <p>3. Индикатор выполнения (ProgressBar): основные свойства, методы и события. Компонент выбора параметра из группы элементов (RadioButton): основные свойства, методы и события. Текстовое поле в формате rtf (RichTextBox): основные свойства, методы и события. Всплывающее окно (ToolTip): основные свойства, методы и события. Компонент для перемещения по страницам внутри формы (WebBrowser): основные свойства, методы и события. Редактирование списков. Сортировка в списках на этапе 1-4 проектирования и во время выполнения приложения. Обработка событий, связанных с выбором пользователем строки списка.</p> <p>4. Обработка множественного выбора в списке. Обработка событий, связанных с вводом пользователем строки в поле комбинированного списка. Свойства списка. Стилль списка. Методы очистки списка, добавления и удаления строк. События, связанные с выбором строк списка и редактированием поля комбинированного списка. Элементы управления – флажки и переключатели (радиокнопки). Состояние флажков и переключателей. Состояние переключателей в группе. Обработка события - выбора переключателя в группе. Программное изменение состояния переключателя. Программное изменение состояния группы переключателей. Контейнеры – формы, рисунки, рамки. Элементы управления – рисунки, изображения, рамки.</p>		
<p>Лекция 3. Меню и панели инструментов.</p> <p>1. Контекстное меню (ContextMenu): основные свойства, методы и события. Главное меню (MenuStrip): основные свойства, методы и события. Строка состояния (StatusStrip): основные свойства, методы и события. Контейнер для объектов панели инструментов (ToolStrip): основные свойства, методы и события.</p> <p>2. Панели (ToolStripContainer): основные свойства, методы и события. Клавиши быстрого вызова. Комбинации клавиш – «горячие клавиши» (акселераторы). Управление доступом к пунктам меню. Группировка команд меню. Процедуры обработки команд меню. Создание и применение контекстного всплывающего меню.</p> <p>3. Визуальный компонент – панель инструментов. Создание панелей инструментов. Размещение панели инструментов в окне. Простые кнопки, кнопки с фиксацией. Группы кнопок: кнопки – переключатели и кнопки – списки. Программирование функций кнопок панели инструментов</p>	4	1-3
<p>Лекция 4. Диалоговые окна и компоненты для вывода на печать.</p> <p>1. Диалоговое окно выбора цвета (ColorDialog): основные свойства, методы и события. Диалоговое окно выбора каталога (FolderBrowserDialog): основные свойства, методы и события. Диалоговое окно выбора шрифта текста (FontDialog): основные свойства, методы и события. Диалоговое окно для открытия файла (OpenFileDialog): основные свойства, методы и события.</p> <p>2. Диалоговое окно для сохранения файла (SaveFileDialog): основные свойства, методы и события. Параметры печати (PageSetupDialog): основные свойства, методы и события. Диалоговое окно печати документов (PrintDialog): основные свойства, методы и события.</p>	4	1-3

<p>Компонент для предварительного просмотра документа перед печатью (PrintPreviewControl): основные свойства, методы и события.</p> <p>3. Создание и использование дополнительных диалоговых окон. Активное окно, скрытое окно, свернутое окно. Модальные и немодальные окна. Встроенные диалоговые окна сообщений. Параметры и возвращаемые значения.</p> <p>4. Использование встроенных диалоговых окон для организации ветвления в программе. Стандартные диалоги. Диалог открытия и сохранения файлов. Диалоги выбора шрифтов и цвета. Вкладки и страницы свойств</p>		
<p>Лекция 5. Контейнеры и дополнительные компоненты.</p> <p>1. Визуальные компоненты, предназначенные для отображения графических примитивов. Визуальные компоненты, предназначенные для просмотра и редактирования графических файлов. Пиксели. Кисть, цвет и стили кисти. Режимы рисования. Графические методы. Изображение примитивов с помощью графических методов.</p> <p>2. Построение графиков и диаграмм. Строка состояния. Панели строки состояния. Расположение панелей состояния. Стили панелей. Визуализация длительных процессов. Компоненты для работы с датой и временем. Всплывающие подсказки.</p> <p>3. Справочная система приложения. Создание справки на основе файлов в формате HTML. Создание и компиляция проекта справки. Разделы справки. Создание содержания справочной системы. Подключение справки к приложению. Вызов справочной системы и контекстной справки.</p>	2	1-3
<p>Лекция 6. Компоненты, предназначенные для работы с базами данных.</p> <p>1. Подключение к базам данных. Выбор таблицы. Компоненты, предназначенные для отображения значения поля и навигации по базе данных. Отображение информации в табличном виде. Фильтрация и упорядочивание записей базы данных.</p> <p>2. Подключение к базе данных и настройка. Построение и редактирование запроса по требованиям пользователя приложения. Запросы с параметром.</p>	2	1-3

### Перечень практических занятий

Тема практического занятия. Вопросы, отрабатываемые на практическом занятии	Всего часов	Учебно-методическое обеспечение
1	2	3
Основы работы в визуальной среде программирования	4	1-3
Создание тестирующих программ с использованием элементов управления	6	1-3
Создание программ с использованием компонентов для работы с датой и временем	4	1-3
Создание текстового редактора с использованием различных меню	6	1-3
Создание программы для отображения информации из базы данных SQL	6	1-3
Создание программы для редактирования реляционной базы данных SQL	6	1-3

**Перечень лабораторных работ – не предусмотрен учебным планом**

## Задания для самостоятельной работы студентов

Вопросы для самостоятельного изучения (задания)	Всего часов	Учебно-методическое обеспечение
1	2	3
Перспективы развития современных сред программирования	10	1-3
Функции преобразования типов (Convert). Операции поиска в списках и комбинированных списках. Работа с массивами элементов управления.	10	1-3
Активизация контекстного меню. Использование изображений для кнопок панели инструментов.	10	1-3
Связывание диалогов с элементами управления – выбор диска, выбор папки, список. Синхронизация элементов управления навигацией.	10	1-3
Перо, цвет и стили пера. Создание справки на основе файлов в формате RTF	10	1-3

### Расчетно-графическая работа не предусмотрена учебным планом

### Курсовая работа не предусмотрена учебным планом

#### Образовательные технологии

При реализации учебного материала курса используются различные образовательные технологии, способствующие созданию атмосферы свободной и творческой дискуссии как между преподавателем и студентами, так и в студенческой группе. Целью при этом является выработка у студентов навыков и компетенций, позволяющих самостоятельно вести исследовательскую и научно-педагогическую работу.

Аудиторные занятия проводятся в виде лекций, практических занятий с использованием ПК и компьютерного проектора. Самостоятельная работа студентов проводится под руководством преподавателей, с оказанием консультаций и помощи при подготовке к практическим занятиям.

В рамках самостоятельной работы студенты изучают электронные образовательные курсы в он-лайн формате.

#### Фонд оценочных средств

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий текущего, рубежного и промежуточного контроля по дисциплине.

Связь между формируемыми компетенциями и формами контроля их освоения представлена в следующей таблице:

№ п/п	Наименование контролируемых разделов (темы)	Код и наименование индикатора достижения компетенций	Наименование оценочного средства
<b>Входной контроль</b>			
1	Входной контроль		Вопросы входного контроля (письменно)
<b>Аттестация разделов, текущий контроль успеваемости</b>			
2	Основные понятия среды визуального программирования	3-ОПК-6, У-ОПК-6, В-ОПК-6, 3-УК-2, 3-УК-2, В-УК-2	Контроль по итогам (в форме теста)

3	Элементы Forms	З-ОПК-6, У-ОПК-6, В-ОПК-6, З-УК-2, З-УК-2, В-УК-2	Контроль по итогам (в форме теста)
Промежуточная аттестация			
4	Зачет	З-ОПК-6, У-ОПК-6, В-ОПК-6, З-УК-2, З-УК-2, В-УК-2	Вопросы к зачету (письменно)

Входной контроль предназначен для выявления пробелов в знаниях студентов и готовности их к получению новых знаний. Оценочные средства для входного контроля представляют собой вопросы, которые задаются студентам в письменной форме.

#### ***Перечень вопросов входного контроля***

1. Охарактеризуйте процессы обработки информации.
2. Понятие переменной и константы.
3. Требования к идентификаторам.
4. Основные типы данных.
5. Типы и характеристика операторов языка СИ.
6. Преобразование типов данных в СИ.
7. Синтаксис и использования операторов условия.
8. Синтаксис и использования операторов циклов.
9. Операторы инкремента и декремента в языке СИ.
10. Работа с массивами в языке СИ.
11. Строки и символьные массивы.
12. Предназначение технологии программирования.
13. Представление алгоритма в графическом виде.
14. Основные принципы объектно-ориентированного программирования.
15. Понятие класса и объекта в ООП.
16. Понятие ассоциации.
17. Понятие процедурного программирования.
18. Свойства алгоритмов.
19. Представление информации в компьютере.
20. Целые и вещественные числа.
21. Понятие декомпозиция.
22. Понятие препроцессор.
23. Понятие компиляция и компоновка.
24. Выражения и основные конструкции в СИ.
25. Приведение типов, адреса переменных и указатели.

Текущий контроль – это непрерывно осуществляемый мониторинг уровня усвоения знаний и формирования умений и навыков в течение семестра. Текущий контроль знаний, умений и навыков студентов осуществляется в ходе учебных (аудиторных) занятий, проводимых по расписанию. Формами текущего контроля выступают используются коллоквиум, выполнение практических контрольных заданий.

#### ***Перечень вопросов вопросов коллоквиума***

1. Общая характеристика платформы MSDN. Структура программы на языке программирования С#.
2. Классификация типов данных. Встроенные типы данных. Преобразование типов.
3. Основные операторы. Чтение данных с клавиатуры. Вывод данных на экран.

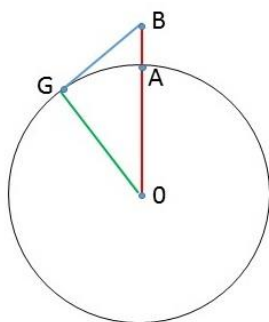


4. Понятие исключительной ситуации.
5. Массивы. Размерность массива. Методы и свойства класса Array.
6. Многомерные массивы. Рваные массивы.
7. Регулярные и буквальные строковые литералы. Ссылки типа string. Операции над строками.
8. Методы и свойства класса String. Форматирование строк. Неизменяемость объектов класса String. Тип System.Text.StringBuilder.
9. Функции. Параметры функции. Функции с переменным числом параметров.
10. Функции. Функции с необязательными параметрами.
11. Функции с именованными параметрами. Перегрузка методов.
12. Понятие потока. Классификация потоков. Основные классы для работы с файловыми потоками.
13. Режимы доступа к файлу, режимы открытия файла, режимы совместного использования файла.
14. Потоки байтов FileStream.
15. Потоки символов StreamWriter и StreamReader.
16. Двоичные файлы BinaryWriter, BinaryReader.
17. Определение класса. Поля. Методы. Объекты.
18. Конструкторы. Деструктор. Сбор мусора.
19. Ключевое слово this. Ключевое слово static.
20. Статические конструкторы. Статические классы.
21. Перегрузка операций (унарные и бинарные операции, приведение типов).
22. Основные свойства ООП. Инкапсуляция. Спецификаторы доступа.
23. Свойства. Автоматические свойства. Индексаторы.
24. Основные свойства ООП. Иерархии классов. Наследование.
25. Виртуальные методы. Механизм позднего связывания.
26. Абстрактные классы. Класс object. Интерфейсы.
27. Работа с объектами через стандартные интерфейсы .NET
28. Понятие делегата. Описание и использование. Многоадресатная передача.
29. События. События для многоадресатной передачи.
30. Механизм обработки исключений. Системные исключения и их обработка.
31. Свойства и методы класса Exception. Исключения, создаваемые программистом.
32. Основные абстрактные структуры данных. Физическое представление данных.
33. Пространство имен System.Collections. Интерфейсы пространства имен System.Collections.
34. Коллекции общего назначения. Основные элементы класса ArrayList.
35. Коллекции общего назначения. Основные элементы класса Hashtable.
36. Коллекции общего назначения. Основные элементы класса SortedList.
37. Коллекции общего назначения. Основные элементы класса Stack.
38. Коллекции общего назначения. Основные элементы класса Queue.
39. Коллекции общего назначения. Доступ к коллекциям с помощью нумератора.
40. Классы-прототипы. Основные преимущества использования обобщений. Ограничения.
41. Параметризованные коллекции библиотеки .NET.
42. Создание собственных классов-прототипов. Обобщенные методы.

#### ***Перечень практических заданий***

1. Вывести на экран число с точностью до сотых.
2. Вывести на экран число  $e$  (основание натурального логарифма) с точностью до десятых.
3. Составить программу вывода на экран числа, вводимого с клавиатуры. Выводимому числу должно предшествовать сообщение «Вы ввели число».

4. Составить программу вывода на экран числа, вводимого с клавиатуры. После выводимого числа должно следовать сообщение « — вот какое число Вы ввели».
5. Вывести на одной строке числа 1, 13 и 49 с одним пробелом между ними.
6. Вывести на одной строке числа 7, 15 и 100 с двумя пробелами между ними.
7. Составить программу вывода на экран в одну строку трех любых чисел с двумя пробелами между ними.
8. Составить программу вывода на экран в одну строку четырех любых чисел с одним пробелом между ними.
9. Вывести на экран числа 50 и 10 одно под другим.
10. Вывести на экран числа 5, 10 и 21 одно под другим.
11. Составить программу вывода на экран «столбиком» четырех любых чисел.
12. Составить программу вывода на экран следующей информации:  
5 10  
7 см
13. Составить программу вывода на экран следующей информации:  
2 кг  
13 17
14. Составить программу:
  - а) вычисления значения функции  $y=7x^2+3x+6$  при любом значении  $x$ ;
  - б) вычисления значения функции  $x=12a^2+7a+12$  при любом значении  $a$ .
15. Дана сторона квадрата. Найти его периметр.
16. Дан радиус окружности. Найти ее диаметр.
17. Считая, что Земля — идеальная сфера с радиусом  $R=6350$  км, определить расстояние до линии горизонта от точки В с заданной высотой  $h=AB$  над Землей.



$h=AB$  – высота,  
 $OA=OG=R$  – радиус Земли,  
 $BG$  – расстояние до горизонта,  
 $BG$  перпендикулярна  $OG$  как  
 касательная к окружности.

18. Дана длина ребра куба. Найти объем куба и площадь его боковой поверхности.
19. Дан радиус окружности. Найти длину окружности и площадь круга.
20. Даны два целых числа. Найти: а) их среднее арифметическое; б) их среднее геометрическое.
21. Известны объем и масса тела. Определить плотность материала этого тела.
22. Известны количество жителей в государстве и площадь его территории. Определить плотность населения в этом государстве.
23. Составить программу решения линейного уравнения  $ax+b=0$  ( $a \neq 0$ ).
24. Даны катеты прямоугольного треугольника. Найти его гипотенузу.
25. Найти площадь кольца по заданным внешнему и внутреннему радиусам.
26. Даны катеты прямоугольного треугольника. Найти его периметр.

27. Даны два числа. Найти среднее арифметическое и среднее геометрическое их модулей.
28. Даны основания и высота равнобедренной трапеции. Найти ее периметр.
29. Даны стороны прямоугольника. Найти его периметр и длину диагонали.
30. Даны два числа. Найти их сумму, разность, произведение, а также частное от деления первого числа на второе.
31. Даны длины сторон прямоугольного параллелепипеда. Найти его объем и площадь боковой поверхности.
32. Известны координаты на плоскости двух точек. Составить программу вычисления расстояния между ними.
33. Даны основания и высота равнобедренной трапеции. Найти периметр трапеции.
34. Даны основания равнобедренной трапеции и угол при большем основании. Найти площадь трапеции.
35. Треугольник задан координатами своих вершин. Найти периметр и площадь треугольника.
36. Выпуклый четырехугольник задан координатами своих вершин. Найти площадь этого четырехугольника как сумму площадей треугольников.
37. Известна стоимость 1 кг конфет, печенья и яблок. Найти стоимость всей покупки, если купили  $x$  кг конфет,  $y$  кг печенья и  $z$  кг яблок.
38. Известна стоимость монитора, системного блока, клавиатуры и мыши. Сколько будут стоить 3 компьютера из этих элементов?  $N$  компьютеров?
39. Возраст Тани —  $X$  лет, а возраст Мити —  $Y$  лет. Найти их средний возраст, а также определить, на сколько отличается возраст каждого ребенка от среднего значения.
40. Два автомобиля едут навстречу друг другу с постоянными скоростями  $V_1$  и  $V_2$  км/час. Определить, через какое время автомобили встретятся, если расстояние между ними было  $S$  км.
41. Два автомобиля едут друг за другом с постоянными скоростями  $V_1$  и  $V_2$  км/час ( $V_1 > V_2$ ). Определить, какое расстояние будет между ними через 30 минут после того, как первый автомобиль опередил второй на  $S$  км.
42. Известно значение температуры по шкале Цельсия. Найти соответствующее значение температуры по шкале: а) Фаренгейта; б) Кельвина. Для пересчета по шкале Фаренгейта необходимо исходное значение температуры умножить на 1,8 и к результату прибавить 32, а по шкале Кельвина абсолютное значение нуля соответствует 273,15 градуса по шкале Цельсия.
43. У американского писателя-фантаста Рэя Бредбери есть роман «450 градусов по Фаренгейту». Какой температуре по шкале Цельсия соответствует указанное в названии значение?

Аттестация раздела по дисциплине проводится в форме тестирования. Тест содержит от 15 вопросов. На выполнение задания отводится от 30 минут. Тест— это форма контроля, направленная на проверку уровня освоения контролируемого теоретического и практического материала по дидактическим единицам дисциплины (терминологический аппарат, основные методы).

Примерный перечень тестовых заданий:

### Тестовые задания 1. (Т1)

1. Что будет содержать строка `name` после выполнения данного кода?

```
1 | string name = "Alexey";
2 | name[1] = ".";
```

- a) Возникнет ошибка
  - b) "Alexey"
  - c) "A.ехey"
  - d) ".lexey"
2. Что делает модификатор `sealed`?
- a) Разрешает наследование только один раз

- b) Ограничивает видимость текущей сборкой
  - c) Добавляется просто для наглядности, но ничего не изменяет
  - d) Запрещает наследование
3. Что будет выведено на консоль?

```
1 | bool isValid = 1;  
2 | Console.WriteLine(isValid);
```

- a) true
  - b) true
  - c) 1
  - d) Возникнет ошибка
4. Что будет содержать массив models после выполнения данного фрагмента кода?

```
1 | object[] models = { 1, "300", 100, "5" };  
2 | Array.Sort(models);
```

- a) 1, "5", 100, "300"
  - b) 1, "300", 100, "5"
  - c) Возникнет ошибка
  - d) 1, 100, "300", "5"
5. Можно ли сделать перегрузку оператора присваивания?
- a) Да
  - b) Нет
6. Структура — это ...
- a) Значимый тип
  - b) Ссылочный тип
7. Что будет выведено на консоль?

```
1 | int firstNumber = 5;  
2 | double secondNumber = 5.0;  
3 | Console.WriteLine(firstNumber == secondNumber);
```

- a) True
  - b) False
  - c) Возникнет ошибка
8. Возникнет ли ошибка при выполнении данного фрагмента кода?

```
1 | object model = 156;  
2 | model = "157";  
3 | Console.WriteLine(model);
```

- a) Да
  - b) Нет
9. Верно ли, что тип string хранит набор символов Unicode?
- a) Да
  - b) Нет
10. Можно ли наследоваться от нескольких интерфейсов?
- a) Да
  - b) Нет
11. C# код компилируется в код ...
- a) CLI
  - b) MASM
  - c) CIL
  - d) FCL

12. Какой способ вызова исключения является корректным?

- a) call new Exception();
- b) throw new Exception();
- c) new Exception();
- d) throw Exception();
- e) Exception();
- f) call Exception();

13. Что будет выведено на консоль?

```
1 | byte firstValue = 200;  
2 | byte secondValue = 100;  
3 | Console.WriteLine(firstValue + secondValue);
```

- a) -44
- b) 300
- c) -45
- d) 44
- e) 45

Возникнет ошибка

14. Что будет выведено на консоль?

```
1 | public class Person  
2 | {  
3 |     public readonly string name = "Artur";  
4 |     public Person(string name)  
5 |     {  
6 |         this.name = name;  
7 |     }  
8 | }  
9 |  
10 |  
11 | class Program  
12 | {  
13 |     static void Main(string[] args)  
14 |     {  
15 |         Person person = new Person("John");  
16 |         Console.WriteLine(person.name);  
17 |  
18 |         Console.ReadKey();  
19 |     }  
20 | }
```

- a) "John"
- b) Возникнет ошибка
- c) "Artur"

15. Выполнится ли блок finally, если исключения не было?

- a) Нет
- b) Да
- c) Зависит от кода
- d) Зависит от режима работы

## Тестовые задания 2 (Т2)

1. Что будет выведено на консоль?

```
1 | dynamic number = 3002;  
2 | Console.WriteLine(number.GetType().Name);
```

- a) Int
- b) Single
- c) Dynamic
- d) Object
- e) Int32

2. Что будет выведено на консоль?

```
1 | int number = -3;
2 | Console.WriteLine((uint)number);
```

- a) -3
  - b) 3
  - c) Возникнет ошибка
  - d) 4294967293
  - e) 0
3. Что будет выведено на консоль?

```
1 | Console.WriteLine(11 / 2);
```

- a) 6
  - b) 5
  - c) 5,5
4. Можно ли сделать перегрузку операторов true и false?
- a) Да
  - b) Нет
5. Тип string является значимым или ссылочным типом?
- a) Значимым
  - b) Ссылочным
6. Что будет выведено на консоль?

```
1 | float digit = 0 / 0f;
2 | Console.WriteLine(digit == digit);
```

- a) True
  - b) NaN
  - c) False
  - d) Возникнет ошибка
7. Что будет выведено на консоль?

```
1 | class Person
2 | {
3 |     public string Name { get; set; }
4 |
5 |     public Person(string name)
6 |     {
7 |         this.Name = name;
8 |     }
9 | }
10 |
11 | class Program
12 | {
13 |     static void Main(string[] args)
14 |     {
15 |         const string personName = "Anatoliy";
16 |
17 |         Person firstPerson = new Person(personName);
18 |         Person secondPerson = new Person(personName);
19 |
20 |         Console.WriteLine(firstPerson == secondPerson);
21 |     }
22 | }
```

- a) False
  - b) Возникнет ошибка
  - c) True
8. Что будет выведено на консоль?

```
1 | public static void Main(string[] args)
2 | {
3 |     {
4 |         int digit = 5;
5 |     }
6 |
7 |     Console.WriteLine(digit);
8 | }
```

- a) 5
- b) 0
- c) Null
- d) Возникнет ошибка
- e) «»

9. Можно ли наследоваться от нескольких классов?

- a) Нет
- b) Да

10. Что делает модификатор partial?

- a) Запрещает наследование
- b) Позволяет использовать неуправляемый код
- c) Ограничивает видимость текущим классом
- d) Позволяет разделять сущность на несколько частей

11. Что будет находиться в переменной name

```
1 | int? a = null;  
2 | string name = a == null ? "Artur" : "Vasya";
```

- a) "Artur"
- b) String.Empty
- c) null
- d) "Vasya"
- e) Возникнет ошибка

12. Что будет выведено на консоль?

```
1 | Console.WriteLine("Text" + " test");
```

- a) Возникнет ошибка
- b) "Text test"
- c) String.Empty
- d) "Text"
- e) " test"

13. Что будет выведено на консоль?

```
1 | int number = 2;  
2 | Console.WriteLine(++number - number-- * ++number);
```

- a) -6
- b) 1
- c) -1
- d) 0
- e) Возникает ошибка

14. Что будет выведено на консоль?

```
1 | enum Names  
2 | {  
3 |     Viktor  
4 | }  
5 |  
6 | static void Main(string[] args)  
7 | {  
8 |     Console.WriteLine(Names.Viktor.GetType().Name);  
9 | }
```

- a) Name
- b) Enum
- c) EnumValue
- d) Object
- e) Value

15. Что будет выведено на консоль?

```
1 | Console.WriteLine(Int16.MaxValue);
```

- a) 32767
- b) 255
- c) 9223372036854775807
- d) 2147483647

### Критерии оценки тестовых заданий, устных опросов:

1. Полнота знаний теоретического контролируемого материала.
2. Количество правильных ответов.

Тестовое задание / опрос считается сданным, если студент правильно ответил на 60 процентов от общего числа вопросов.

Критерии оценивания	Оценка
Студент ответил на 90 % (и более) вопросов	Отлично
Студент ответил на 70-89 % вопросов	Хорошо
Студент ответил на 60-69 % вопросов	Удовлетворительно
Студент ответил менее чем на 59 % вопросов	Неудовлетворительно

Сумма баллов по разделам дисциплины складывается из оценок, полученных обучающимся в течение семестра по всем формам текущего контроля. Каждая форма контроля оценивается баллом в интервале от 0 до 10.

Промежуточная аттестация осуществляется в форме зачета. Зачет проводится в форме тестирования. Тест содержит от 20 вопросов. Тест – это форма контроля, направленная на проверку уровня освоения контролируемого теоретического и практического материала по дидактическим единицам дисциплины (терминологический аппарат, основные методы).

### Примерный перечень тестовых заданий:

1. Какой тип переменной используется в коде: `int a = 5`
  - a) Знаковое 8-бит целое
  - b) Знаковое 64-бит целое
  - c) Знаковое 32-бит целое
  - d) 1 байт
2. Что делает оператор «%»
  - a) Возвращает процент от суммы
  - b) Возвращает остаток от деления (+)
  - c) Возвращает тригонометрическую функцию
  - d) Ни чего из выше перечисленного.
3. Что сделает программа выполнив следующий код: `Console.WriteLine(«Hello, World!»);`
  - a) Напишет на новой строке Hello, World!
  - b) Напишет Hello, World!
  - c) Удалит все значения с Hello, World!
  - d) Вырежет слово Hello, World! из всего текста
4. Как сделать инкрементацию числа?
  - a) ++
  - b) —
  - c) %%
  - d) !=
5. Как сделать декрементация числа?
  - a) %%
  - b) —
  - c) !=
  - d) ++
6. Как найти квадратный корень из числа x
  - a) `Sqrt(x)`
  - b) `Summ.Koren(x);`
  - c) `Arifmetic.sqrt(x);`
  - d) `Math.Sqrt(x);`
7. Обозначения оператора «НЕ»
  - a) Not
  - b) No
  - c) !



d) !=

8. Обозначение оператора «ИЛИ»

a) !

b) !=

c) ||

d) Or

9. Обозначение оператора «И»

a) and

b) &&

c) &

d) Все выше перечисленные.

10. Чему будет равен c, если `int a = 10; int b = 4; int c = a % b;`

a) 11

b) 2

c) 3

d) 1

11. Чему будет равен c, если `int a = 10; int b = 4; bool c = (a == 10 && b == 4);`

a) True

b) False

c) Null

d) 14

12. Чему будет равен c, если `int a = 0; int c = a—;`

a) Null

b) -1

c) 0

d) 1

13. Чему будет равен c, если `int a = 0; int c = —a;`

a) Null

b) -1

c) 1

14. Чему равен d, если `int a = 0; int b = a++; int c = 0; int d = a + b + c + 3;`

a) 3

b) True

c) False

d) 4

15. Для чего нужны условные операторы

a) Чтобы устанавливать условия пользователю

b) Для ветвления программы

c) Для оптимизации программы

d) Чтобы были.

16. Что вернет функция `Termin` после выполнения. Код:

```
int Termin()
{
int a = 1;
int b = 3;
if (a != 5) return a + b;
else return 0;
}
```

a) 5

b) 3

c) 4

d) 0

17. Как называется оператор «?:»

a) Вопросительный

b) Прямой оператор

c) Тернарный оператор

d) Территориальный оператор

18. Что такое массив

a) Набор однотипных данных, которые располагаются в памяти последовательно друг за другом

b) Набор текстовых значений в формате Unicode, которые расположены в случайном порядке.

c) Набор данных типа int (32-бит целое)

d) Переменная

19. Какие бывают массивы ?

a) Разнообразные

b) Сложные и простые

c) Одномерные и многомерные

d) Резиновые и статичные

20. Что такое цикл и для чего они нужны

a) Циклы нужны для многократного запуска программы

b) Циклы нужны для многократного выполнения кода.

c) Циклы нужны для многократного размещения данных

d) Циклы нужны чтобы выполнить код без ошибок

### Тестовое задание (Т2).

1. Какие бывают циклы?

a) Большие и маленькие

b) Цикл, Форич, Двойной цикл, Многократный

c) for, while, do-while, foreach

d) ref, out, static, root

2. Какой оператор возвращает значение из метода ?

a) veni

b) return

c) out

d) end

3. Что такое константа ?

a) Переменная типа string

b) Переменная которая может быть изменена в любое время.

c) Глобальная переменная

d) Переменная значение которой нельзя изменить.

4. Что обозначает ключевое слово var ?

a) Устраивает войну между программами

b) Обозначает что переменная без явного типа данных

c) Обозначает что переменная имеет явный тип данных

d) Такого слова нет в C#

5. Что такое Куча ?

a) Это структура данных

b) Именованная область памяти

c) Область динамической памяти

d) Куча переменных

6. Сколько родительских классов может иметь производный класс?

a) Всегда один

b) Не больше двух

c) Не больше одного

d) Любое количество

7. Пространство имен – это:

a) множество имен, включающих имена переменных проекта

b) множество имен, включающих имена переменных класса

c) модуль, содержащий встроенные пространства имен и классы, имена которых уникальны в пространстве имен

8. Проект – это:

a) модуль, содержащий классы и являющийся частью решения

- b) спецификация, отвечающая требованиям CLS
- c) компонент пространства имен
- d) набор классов, используемых при построении самой Visual Studio .Net

9. К значимым типам языка C# относятся:

- a) все арифметические типы
- b) массивы
- c) строки
- d) все арифметические типы, кроме типа double

10. К ссылочным типам языка C# относятся:

- a) тип double
- b) массивы
- c) строки
- d) все арифметические типы, кроме типа double
- e) структуры

11. В результате объявления: `int x=2,y=5,z=((x|y +x++)>=9) ? x|y : ++x +2 ;` переменные x, y, z получают значения

- a) x=3; y=5; z=7
- b) x=4; y=5; z=6
- c) x=2; y=5; z=7
- d) возникнет ошибка на этапе трансляции
- e) возникнет ошибка на этапе выполнения
- f) x=3; y=5; z=8

12. Под перегруженной операцией понимается операция:

- a) для которой существует несколько реализаций
- b) сложность которой превышает сложность операции сложения

13. Даны объявления:

```
char ch = 'A';
```

```
char[] ar = {'B', 'C', 'D'}, ar1 = new char[3];
```

Отметьте синтаксически корректные операторы:

- a) `ar1 = char.Parse("BCD")`
- b) `ar.CopyTo(ar1,0)`
- c) `ch = char.Copy("B")`
- d) `ar1 = Array.Copy(ar)`

14. Для строки s типа string запись s[i]

- a) означает символ строки типа char
- b) означает символ строки типа string
- c) не определена

15. Отметьте правильное высказывание:

- a) эквивалентность строк (`s == s1`) означает совпадение ссылок s и s1
- b) операция (`s+s1`) определена только над строками, содержащими числа
- c) получить i-й символ строки, используя операцию взятия индекса s[i], недопустимо
- d) аргументы в методе `Format` позволяют задать ссылку и способ форматирования объектов,

вставляемых в строку

e) у класса string нет статических методов

16. Отметьте правильное высказывание:

- a) присваивание `ch = s[i]`; где s класса string, а ch – переменная класса char, недопустимо
- b) константы `"\c\x58"` и `@"\cX"` эквивалентны
- c) метод `Join` всегда позволяет восстановить исходную строку, расщепленную методом `Split`
- d) аргументы в методе `Format` позволяют задать ширину области для вставляемого объекта

17. Изменить значение строки класса string

- a) можно только используя статические методы класса
- b) можно только используя динамические методы класса
- c) вообще невозможно

18. Для классов родителей и потомков справедливо следующее утверждение:

- a) у родительского класса может быть несколько непосредственных потомков
- b) у класса потомка может быть несколько непосредственных родительских классов

- c) класс родитель может быть сам себе родитель
  - d) класс потомок может быть собственным потомком
19. Ключевое слово `interface` в языке `C#` задает описание:
- a) пользовательского интерфейса класса
  - b) открытой части класса
  - c) частного случая класса
  - d) абстрактного класса
20. Как правильно написать команду вывода текста в консоли?
- a) `Console.Read();`
  - b) `Console.WriteLine();`
  - c) `Console.ReadLine();`

### Критерии оценки тестовых заданий, устных опросов:

1. Полнота знаний теоретического контролируемого материала.
2. Количество правильных ответов.

Тестовое задание / опрос считается сданным, если студент правильно ответил на 60 процентов от общего числа вопросов.

Критерии оценивания	Оценка
Студент ответил на 90 % (и более) вопросов	Отлично
Студент ответил на 70-89 % вопросов	Хорошо
Студент ответил на 60-69 % вопросов	Удовлетворительно
Студент ответил менее чем на 59 % вопросов	Неудовлетворительно

Сумма баллов по разделам дисциплины складывается из оценок, полученных обучающимся в течение семестра по всем формам текущего контроля. Каждая форма контроля оценивается баллом в интервале от 0 до 10.

### Учебно-методическое и информационное обеспечение учебной дисциплины

Основная литература:

1. Тюкачев, Н. А. `C#`. Алгоритмы и структуры данных / Н. А. Тюкачев, В. Г. Хлебостроев. — 6-е изд., стер. — Санкт-Петербург : Лань, 2023. — 232 с. — ISBN 978-5-507-47248-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/346067>
2. Тюкачев, Н. А. `C#`. Основы программирования : учебное пособие для вузов / Н. А. Тюкачев, В. Г. Хлебостроев. — 4-е изд., стер. — Санкт-Петербург : Лань, 2021. — 272 с. — ISBN 978-5-8114-7266-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/158960>

Дополнительная литература:

3. Заборовский, Г. А. Программирование на языке `C#` : учебно-методическое пособие / Г. А. Заборовский, В. В. Сидорик. — Минск : БНТУ, 2020. — 84 с. — ISBN 978-985-583-074-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/248405>

Программное обеспечение и Интернет-ресурсы:

1. Электронно-библиотечная система «Лань» - <http://e.lanbook.com>.
2. Электронно-библиотечная система «Юрайт» - <http://www.biblio-onlain.ru>.
3. Российская национальная библиотека - <http://www.nlr.ru/>
4. Поисковая система - <http://www.rambler.ru>.
5. Поисковая система - <http://www.yandex.ru>.
6. Гарант - <http://base.garant.ru/>.
7. Интернет-Университет Информационных Технологий - <http://www.intuit.ru>

Для проведения практических занятий и выполнения самостоятельной работы используются учебные компьютерные классы с выходом в Интернет и лицензионным программным обеспечением.

## **Материально-техническое обеспечение учебной дисциплины**

Лекционные занятия проводятся в аудиториях: учебная мебель, учебная доска, комплект мультимедийного оборудования

Практические занятия проводятся в компьютерных классах: учебная мебель, учебная доска, комплект мультимедийного оборудования, персональные компьютеры.

Для самостоятельной работы обучающихся имеется: читальный зал с выходом в сеть Интернет: Учебная мебель, комплект мультимедийного оборудования, персональные компьютеры, МФУ.

## **Учебно-методические рекомендации для студентов**

### **1. Указания для прослушивания лекций**

Перед началом занятий внимательно ознакомиться с учебным планом проведения лекций и списком рекомендованной литературы.

Перед посещением очередной лекции освежить в памяти основные концепции пройденного ранее материала. Подготовить при необходимости вопросы преподавателю. Не надо опасаться, что вопросы могут быть простыми.

На лекции основное внимание следует уделять не формулам и математическим выкладкам, а содержанию изучаемых вопросов, определениям и постановкам задач.

В процессе изучения лекционного курса необходимо по возможности часто возвращаться к основным понятиям и методам решения задач (здесь возможен выборочный контроль знаний студентов).

Желательно использовать конспекты лекций, в которых используется принятая преподавателем система обозначений.

Для более подробного изучения курса следует работать с рекомендованными литературными источниками и вновь появляющимися источниками.

### **2. Указания для участия в практических занятиях**

Перед посещением уяснить тему практического занятия и самостоятельно изучить теоретические вопросы.

В конце семинара при необходимости выяснить у преподавателя неясные вопросы.

Основные результаты выполнения работы необходимо распечатать.

### **3. Самостоятельная работа студентов обычно складывается из нескольких составляющих:**

- работа с текстами: учебниками, историческими первоисточниками, дополнительной литературой, в том числе материалами интернета, а также проработка конспектов лекций;
- написание докладов, рефератов;
- подготовка к практическим занятиям;
- подготовка к зачету непосредственно перед ним.

Таким образом, самостоятельная работа студентов является необходимым компонентом получения полноценного высшего образования.

## **Методические рекомендации для преподавателей**

### **1. Указания для проведения лекций**

На первой вводной лекции сделать общий обзор содержания курса и отметить новые методы и подходы к решению задач, рассматриваемых в курсе, довести до студентов требования кафедры, ответить на вопросы.

При подготовке к лекционным занятиям необходимо продумать план его проведения, содержание вступительной, основной и заключительной части лекции, ознакомиться с новинками учебной и методической литературы, публикациями периодической печати по теме лекционного занятия. Уточнить план проведения семинарского занятия по теме лекции. Перед изложением текущего лекционного материала напомнить об основных итогах, достигнутых на предыдущих лекциях. С этой целью задать несколько вопросов аудитории и осуществить выборочный контроль

знания студентов.

В ходе лекционного занятия преподаватель должен назвать тему, учебные вопросы, ознакомить студентов с перечнем основной и дополнительной литературы по теме занятия. Раскрывая содержание учебных вопросов, акцентировать внимание студентов на основных категориях, явлениях и процессах, особенностях их протекания. Раскрывать сущность и содержание различных точек зрения и научных подходов к объяснению тех или иных явлений и процессов.

Следует аргументировано обосновать собственную позицию по спорным теоретическим вопросам. Приводить примеры. Задавать по ходу изложения лекционного материала риторические вопросы и самому давать на них ответ. Это способствует активизации мыслительной деятельности студентов, повышению их внимания и интереса к материалу лекции, ее содержанию. Преподаватель должен руководить работой студентов по конспектированию лекционного материала, подчеркивать необходимость отражения в конспектах основных положений изучаемой темы, особо выделяя категориальный аппарат. В заключительной части лекции необходимо сформулировать общие выводы по теме, раскрывающие содержание всех вопросов, поставленных в лекции. Объявить план очередного практического занятия, дать краткие рекомендации по подготовке студентов к семинару. Определить место и время консультации студентам, пожелавшим выступить на семинаре с докладами и рефератами.

На последней лекции уделить время для обзора наиболее важных положений, рассмотренных в курсе.

## 2. Указания для проведения семинарских занятий

Четко обозначить тему семинара.

Обсудить основные понятия, связанные с темой семинара.

В процессе решения задач вести дискуссию со студентами о правильности применения теоретических знаний.

Отмечать студентов, наиболее активно участвующих в решении задач и дискуссиях.

В конце семинара задать аудитории несколько контрольных вопросов.

## 3. Указания по контролю самостоятельной работы студентов

По усмотрению преподавателя задание на самостоятельную работу может быть индивидуальным или фронтальным.

При использовании индивидуальных заданий требовать от студента письменный отчет о проделанной работе.

При применении фронтальных заданий вести коллективные обсуждения со студентами основных теоретических положений.

С целью контроля качества выполнения самостоятельной работы требовать индивидуальные отчеты (допустимо вместо письменного отчета применять индивидуальные контрольные вопросы).

Программа составлена в соответствии с требованиями ОС НИЯУ МИФИ по направлению 09.03.02 Информационные системы и технологии

Рабочую программу составил ст. пр. А.Г. Мотков

Рецензент: доцент И.А. Штырова

Программа одобрена на заседании УМКН «Информационные системы и технологии».

Председатель учебно-методической комиссии О.В. Виштак