МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»

Обнинский институт атомной энергетики –

филиал федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский ядерный университет «МИФИ»

(ИАТЭ НИЯУ МИФИ)

ОТДЕЛЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

Одобрено на заседании Ученого совета ИАТЭ НИЯУ МИФИ, протокол №25.8 от 25.08.2025

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ПРОГРАММИРОВАНИЕ

название дисциплины

для студентов направления подготовки

38.03.05 Бизнес-информатика

образовательная программа:

Цифровая трансформация компании

Форма обучения: очная

г. Обнинск 2025 г.

1. Цели и задачи освоения дисциплины

Цель дисциплины – обучение студентов основам программирования и практическим навыкам их применения в рамках бизнес-информатики.

Задачи дисциплины –

- формирование четкого представления о базовых понятиях программирования.
- развитие представлений о сферах применения программ: обработка текстовой, числовой, особенности моделирования предметной области;
- знакомство с профессиональным средствами программирования для разработки программного обеспечения;
- приобретение практических навыков по обработке текстовой и числовой информации, с помощью разработанного программного обеспечения.

2. Место дисциплины в структуре образовательной программы (далее – ОП) бакалавриата

Дисциплина реализуется в рамках обязательной части.

Для освоения дисциплины необходимы компетенции, сформированные в рамках изучения следующих дисциплин:

- математика (школьный уровень);
- информатика и ИКТ (школьный уровень).

Дисциплины и/или практики, для которых освоение данной дисциплины необходимо как предшествующее:

- объектно-ориентированное программирование;
- базы данных;
- веб-программирование;
- основы разработки мобильных приложений.

Дисциплина изучается на 1 курсе в 1 и 2 семестрах.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения ОПОП бакалавриата обучающийся должен овладеть следующими результатами обучения по дисциплине:

Код компетенции	Наименование компетенции	Код и наименование индикатора достижения компетенции
ОПК-3	Способен управлять процессами создания и использования продуктов и услуг в сфере ИКТ, в том числе разрабатывать алгоритмы и программы для их практической реализации	З-ОПК-3 Знать: Методы оценки объемов и сроков выполнения работ Технологии выполнения работ в организации Технологии межличностной и групповой коммуникации в деловом взаимодействии, основы конфликтологии Архитектура, устройство и функционирование вычислительных систем Коммуникационное оборудование Сетевые протоколы Основы современных операционных систем Основы современных систем управления базами данных Устройство и функционирование современных ИС Теория баз данных Системы хранения и анализа баз данных Основы программирования Современные объектно-ориентированные языки программирования Современные структурные языки программирования Языки современных бизнес-приложений Современные методики тестирования разрабатываемых ИС Современные стандарты информационного взаимодействия систем Программные средства и платформы инфраструктуры информационных технологий организаций Современные подходы и стандарты автоматизации организации (например, Web-системы, CRM, MRP, ERP..., ITIL, ITSM) Системы классификации и кодирования информации, в том числе присвоение кодов документам и элементам справочников Отраслевая нормативная техническая документация Источники информации, необходимой для профессиональной деятельности У-ОПК-3 Уметь: Разрабатывать документы Оценивать объемы работ и сроки их выполнения Проводить переговоры В-ОПК-3 Владеть навыками: Подготовка частей коммерческого предложения заказчику касательно объема и сроков выполнения работ по созданию (модификации) и вводу в эксплуатацию

		типовой ИС Осуществление инженерно-технологической поддержки в ходе согласования коммерческого предложения с заказчиком

4. Воспитательный потенциал дисциплины

Направления / цели воспитания	Задачи воспитания (код)	Воспитательный потенциал дисциплин
Профессиональное и трудовое воспитание	Формирование культуры исследовательской и инженерной деятельности (В-16)	Использование воспитательного потенциала дисциплины "Программирование" для формирования навыков владения эвристическими методами поиска и выбора технических решений в условиях неопределенности через специальные задания с использованием программных пакетов.

Организация интерактивных мероприятий и реализация специализированных заданий с воспитательным и социальным акцентом:

1. Организация научно-практических конференций, круглых столов, встреч с выдающимися учеными и ведущими специалистами отраслей.
2. Организация и проведение предметных олимпиад и конкурсов профессионального мастерства.

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

Вид работы	**Форма обучения** (вносятся данные по реализуемым формам)		
	Очная		
	Семестр		
	№ 1	**№ 2**	**Всего**
	Количество часов на вид работы:		
Контактная работа обучающихся с преподавателем			
Аудиторные занятия *(всего)*	64	64	128
В том числе:			
лекции (лекции в интерактивной форме)	-	-	-

практические занятия (практические занятия в интерактивной форме)	32	32	64
лабораторные занятия	32	32	64
Промежуточная аттестация			
В том числе:			
Зачет с оценкой			
экзамен		54	54
Самостоятельная работа обучающихся			
Самостоятельная работа обучающихся *(всего)*	80	62	142
В том числе:			
проработка учебного (теоретического) материала	36	30	66
подготовка к выполнению лабораторных работ	22	16	38
подготовка ко всем видам контрольных испытаний текущего контроля успеваемости (в течение семестра)	22	16	38
Всего (часы):	**144**	**180**	**324**
Всего (зачетные единицы):	**4**	**5**	**9**

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

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

№ п/п	Наименование раздела /темы дисциплины	**Виды учебной работы** в часах (вносятся данные по реализуемым формам)				
		Очная форма обучения				
		Лек	**Пр**	**Лаб**	**Внеауд**	**СРО**
1.	**Основы программирования**					
1.1.	Введение	-	2	2	-	3
1.2.	Примитивные типы данных	-	2	2	-	3
1.3.	Ветвление	-	2	2	-	3
1.4.	Методы	-	4	4	-	3
1.5.	Итерации	-	4	4	-	3
1.6.	Обработка исключений	-	2	2	-	3
1.7.	Массивы	-	4	4	-	3
1.8.	Символьные типы данных	-	4	4	-	3
1.9.	Составные типы данных	-	6	6	-	5
1.10.	Файлы	-	4	4	-	7
	Итого за 1 семестр:		32	32		36
2.	**Программирование на платформе .NET**					
2.1.	Универсальные шаблоны	-	4	4	-	6
2.2.	Делегаты, События	-	6	6	-	6
2.3.	Коллекции	-	8	8	-	6
2.4.	Асинхронность и параллелизм	-	6	6	-	6
2.5.	Настольные приложения с графическим интерфейсом	-	8	8	-	6
	Итого за 2 семестр:		32	32	-	30

	Всего:		**64**	**64**	**-**	**66**

Прим.: Лек – лекции, Пр – практические занятия / семинары, Лаб – лабораторные занятия, Внеауд – внеаудиторная работа, СРО – самостоятельная работа обучающихся

6.2. Содержание дисциплины, структурированное по разделам (темам)

Практические/семинарские занятия

№	Наименование раздела /темы дисциплины	Содержание
1.	**Основы программирования**	
1.1.	Введение	Применение языков программирования и компьютерных программ. Язык программирования C#/C++ и платформа .NET. Программные средства для работы с кодом. Первая программы на языке C#/C++. Понятие ключевых слов. Структура программы. Пространства имен. Подсветка синтаксиса. Работа со спецификацией языка C#/C++ для разработчиков.
1.2.	Примитивные типы данных	Представление объектов реального мира в виде цифровых моделей. Отличие математической модели от цифровой модели. Проблемы с бесконечностью, деление на нуль, область определения. Числовые типы данных: целочисленные, числа с плавающей запятой. Операции над числовыми типами данных. Приведение типов. Класс Convert. Класс Math.
1.3.	Ветвление	Ветвление. Конструкция if/else. Конструкция switch/case. Тип данных bool. Логические операторы. Логические выражения. Тернарный оператор.
1.4.	Методы	Процедура. Функция. Сигнатура метода. Передача параметров по значению. Передача параметров по ссылке. Перегрузка методов. Рекурсия. Методы. Определение метода. Вызов метода. Параметры метода. Перегрузка метода. Параметры по умолчанию. Возврат значения.
1.5.	Итерации	Операторы итерации. Цикл for. Цикл while. Цикл do/while. Операторы break, continue.
1.6.	Обработка исключений	Исключения. Виды ошибок. Методы обработки ошибок. Блок try/catch. Блок try/catch/finally. Блок try/catch/catch/catch. Оператор throw.
1.7.	Массивы	Массивы. Обход массива. Обращение к элементам массива. Передача массива в качестве аргумента. Многомерные массивы. Поиск. Сортировка. Класс Random.
1.8.	Символьные типы данных	Символьный тип данных. Литералы. Преобразование типов. Методы char. Получение символа из строки. Строковой тип данных.

		Конкатенация строк. Экранирование символов. Сравнение строк. Методы string. Класс StringBuilder
1.9.	Составные типы данных	Структура. Класс. Конструктор. Поля, методы. Модификаторы readonly, static, const. Свойства. Ключевое слово this. Инкапсуляция. Наследование. Ключевое слово null. Интерфейс. Перечисления. Стандарт оформления кода. Язык UML: диаграмма классов.
1.10.	Файлы	Запись в файл. Чтение файла. Класс File. Классы потоков StreamWriter, StreamReader. Сериализация. Json сериализация.
2.	**Программирование на платформе .NET**	
2.1.	Универсальные шаблоны	Универсальные шаблоны. Преимущества и недостатки универсальных шаблонов Универсальные коллекции. Использование универсальных шаблонов.
2.2.	Делегаты, события	Делегаты. Определение делегатов. Присвоение ссылки на метод. Соответствие методов делегату. Добавление методов в делегат. Объединение делегатов. Вызов делегата. События. Определение и вызов события. Добавление обработчика события. Удаление обработчика события. Управление обработчиками.
2.3.	Коллекции	Списки. Словари. Стек. Очередь. Интерфейс ICollection<T>. Интерфейс IEnumerable<T>. Инициализация коллекций. Обращение к элементам коллекции. Методы коллекций. Language Integrated Query (LINQ). LINQ to Objects. Синтаксис выражений LINQ. Методы расширения LINQ.
2.4.	Асинхронность и параллелизм	Асинхронные методы, async и await. Возвращение результата из асинхронного метода. Последовательный и параллельный вызов асинхронных операций. Обработка ошибок в асинхронных методах. Отмена асинхронных операций. Параллельное программирование и библиотека TPL. Задачи продолжения. Класс Parallel. Класс CancellationToken**.**
2.5.	Настольные приложения с графическим интерфейсом	Обзор платформ для создания настольных графических приложений. Примеры создания и настройка проектов настольных приложений. Пример создания настольного приложения.

Лабораторные занятия

№	Наименование раздела /темы дисциплины	Название лабораторной работы
1.	**Основы программирования**	
1.1.	Введение	*Лабораторная работа №1.1*
1.2.	Примитивные типы данных	*Лабораторная работа №1.2*
1.3.	Ветвление	*Лабораторная работа №1.3*
1.4.	Методы	*Лабораторная работа №1.4*
1.5.	Итерации	*Лабораторная работа №1.5*
1.6.	Обработка исключений	*Лабораторная работа №1.6*
1.7.	Массивы	*Лабораторная работа №1.6*
1.8.	Символьные типы данных	*Лабораторная работа №1.7*
1.9.	Составные типы данных	*Лабораторная работа №1.8*
1.10.	Файлы	*Лабораторная работа №1.9*
2.	**Программирование на платформе .NET**	
2.1.	Универсальные шаблоны	*Лабораторная работа №2.1*
2.2.	Делегаты, события	*Лабораторная работа №2.2*
2.3.	Коллекции	*Лабораторная работа №2.3*
2.4.	Асинхронность и параллелизм	*Лабораторная работа №2.4*
2.5.	Настольные приложения с графическим интерфейсом	*Лабораторная работа №2.5*

7. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

В качестве учебно-методических материалов используется рекомендованная литература и рекомендованные ресурсы сети Интернет (разделы 7 и 8).

8. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

8.1. Связь между формируемыми компетенциями и формами контроля их освоения

№ п/п	Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части) / и ее формулировка	Наименование оценочного средства
Текущий контроль, 1 семестр			
1.	Введение	З-ОПК-3; У-ОПК-3; В-ОПК-3	Лабораторная работа №1.1
2.	Примитивные типы данных	З-ОПК-3; У-ОПК-3; В-ОПК-3	Лабораторная работа №1.2
3.	Ветвление	З-ОПК-3; У-ОПК-3; В-ОПК-3	Лабораторная работа №1.3
4.	Методы	З-ОПК-3; У-ОПК-3; В-ОПК-3	Лабораторная работа №1.4
5.	Итерации	З-ОПК-3; У-ОПК-3; В-ОПК-3	Лабораторная работа №1.5
6.	Обработка исключений	З-ОПК-3; У-ОПК-3; В-ОПК-3	Лабораторная работа №1.6
7.	Массивы	З-ОПК-3; У-ОПК-3; В-ОПК-3	Лабораторная работа №1.6
8.	Символьные типы данных	З-ОПК-3; У-ОПК-3; В-ОПК-3	Лабораторная работа №1.7
9.	Составные типы данных	З-ОПК-3; У-ОПК-3; В-ОПК-3	Лабораторная работа №1.8
10.	Файлы	З-ОПК-3; У-ОПК-3; В-ОПК-3	Лабораторная работа №1.9
Промежуточный контроль, 1 семестр			
	Зачет с оценкой	З-ОПК-3; У-ОПК-3; В-ОПК-3	Зачет
Всего:			
Текущий контроль, 2 семестр			
1.	Универсальные шаблоны	З-ОПК-3; У-ОПК-3; В-ОПК-3	Лабораторная работа №2.1
2.	Делегаты, события	З-ОПК-3; У-ОПК-3; В-ОПК-3	Лабораторная работа №2.2
3.	Коллекции	З-ОПК-3; У-ОПК-3; В-ОПК-3	Лабораторная работа №2.3
4.	Асинхронность и параллелизм	З-ОПК-3; У-ОПК-3; В-ОПК-3	Лабораторная работа №2.4
5.	Настольные приложения с графическим интерфейсом	З-ОПК-3; У-ОПК-3; В-ОПК-3	Лабораторная работа №2.5
Промежуточный контроль, 2 семестр			
	Экзамен	З-ОПК-3; У-ОПК-3; В-ОПК-3	Экзамен
Всего:			

8.2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций в процессе освоения образовательной программы

8.2.1. Экзамен

Экзамен проводится в виде письменных ответов на 1 вопрос и 1 задание, с последующим устным собеседованием.

а) типовые вопросы (задания):

1. Применение языков программирования и компьютерных программ. Язык программирования C#/C++ и платформа .NET. Программные средства для работы с кодом. Первая программы на языке C#/C++. Понятие ключевых слов. Структура программы. Пространства имен.
2. Представление объектов реального мира в виде цифровых моделей. Отличие математической модели от цифровой модели. Проблемы с бесконечностью, деление на нуль, область определения. Числовые типы данных: целочисленные, числа с плавающей запятой.

3. Операции над числовыми типами данных.
4. Приведение типов. Класс Convert.
5. Класс Math.
6. Ветвление. Конструкция if/else.
7. Конструкция switch/case.
8. Тип данных bool. Логические операторы. Логические выражения. Тернарный оператор.
9. Методы. Процедура. Функция. Сигнатура метода. Передача параметров по значению. Передача параметров по ссылке.
10. Рекурсия.
11. Перегрузка методов. Параметры по умолчанию. Именованные аргументы. Возврат значения.
12. Операторы итерации. Цикл for. Цикл while. Цикл do/while. Операторы break, continue.
13. Исключения. Виды ошибок. Методы обработки ошибок. Блок try/catch. Блок try/catch/finally. Блок try/catch/catch/catch. Оператор throw.
14. Одномерные массивы. Обход массива. Обращение к элементам массива. Передача массива в качестве аргумента.
15. Многомерные массивы.
16. Поиск в массиве.
17. Сортировка массива.
18. Класс Random.
19. Нерегулярные массивы.
20. Символьный тип данных. Литералы. Преобразование типов. Методы char. Получение символа из строки.
21. Строковой тип данных. Конкатенация строк. Экранирование символов. Сравнение строк. Методы string. Класс StringBuilder. Форматирование строк.
22. Структура. Класс. Конструктор. Поля, методы. Ключевое слово null. Перечисления. Стандарт оформления кода.
23. Наследование.
24. Язык UML: диаграмма классов.
25. Модификаторы readonly, static, const.
26. Свойства.
27. Ключевое слово this. Инкапсуляция. Интерфейс.
28. Запись в файл. Чтение файла. Класс File.
29. Классы потоков StreamWriter, StreamReader.
30. Сериализация.
31. Универсальные шаблоны. Преимущества и недостатки универсальных шаблонов.
32. Универсальные коллекции.
33. Делегаты. Определение делегатов. Присвоение ссылки на метод. Соответствие методов делегату. Добавление методов в делегат. Объединение делегатов. Вызов делегата.
34. События. Определение и вызов события. Добавление обработчика события. Удаление обработчика события. Управление обработчиками.
35. Списки.

36. Словари.
37. Стек.
38. Очередь.
39. Интерфейс ICollection<T>.
40. Интерфейс IEnumerable<T>.
41. Обращение к элементам коллекции. Методы коллекций. Language Integrated Query (LINQ). LINQ to Objects. Синтаксис выражений LINQ. Методы расширения LINQ.
42. Асинхронные методы, async и await.
43. Возвращение результата из асинхронного метода.
44. Последовательный и параллельный вызов асинхронных операций. Обработка ошибок в асинхронных методах.
45. Отмена асинхронных операций.
46. Параллельное программирование и библиотека TPL.
47. Задачи продолжения.
48. Класс Parallel.
49. Класс CancellationToken**.**
50. Обзор платформ для создания настольных графических приложений.

Типовое задание:

1. Создайте программу обработки набора данных с использованием LINQ. Дан набор данных о компаниях:

Название	Дата регистрации	ИНН	Доход за год
ООО «Ромашка»	23.08.1994	950201086898	30050000
ООО «Клюква»	25.06.2000	484803888888	23000000
ПАО «Трейд»	01.03.2005	323302562858	35000000
ПАО «Капитал»	28.01.1993	978543123256	31200000
ООО «Инвест»	02.06.2010	255656841552	82003000

Программа должна вывести на экран:

a. Вывести информацию о среднем доходе по компаниям за год.
b. Вывести информацию о компаниях младше 20 лет.
c. Произвести проверку на уникальность ИНН.

2. Написать программу для json сериализации класса Pet. Класс Pet содержит свойства кличка, имя хозяина, возраст. Возраст задается целочисленным типом данных. Данные о Pet вводятся с клавиатуры. Программа должна выдавать пользователю пояснения о производимых действиях. Значение возраста должно корректно обрабатываться. Ограничений на имя и фамилию нет. В результате работы программы должен быть создан файл json формата.

3. Написать программу для вычисления значения функции. Переменные типа double.

$$y = \begin{cases} x - 2, & x > 15 \\ 1 + x, & -2 \le x \le 15 \\ x + 120, & x < -2 \end{cases}$$

4. Сгенерировать целочисленный массив из N элементов. N вводится с клавиатуры пользователем. Массив должен содержать отрицательные элементы. Найти сумму и среднее арифметическое элементов массива. Вывести результаты работы программы на экран. Вывести исходный массив на экран. Записать результаты в файл. (нельзя использовать Linq).

5. Сгенерировать массив типа double из N элементов. N вводится с клавиатуры пользователем. Найти минимальный и максимальный элемент массива. Вывести результаты работы программы на экран. Вывести исходный массив на экран. Записать результаты в файл. (нельзя использовать Linq).

6. Программа принимает на вход текст от пользователя. Программа должна удалить из текста все символы(*,!;%№:?.). Убрать пробелы. Вывести символы текста в обратном порядке Результат записать в файл.

7. Программа принимает на вход текст от пользователя. Программа должна удалить из текста все цифры. Разделить текст на предложения. Результат записать в файл.

8. Рассчитать значение функции от одной переменной. Все переменные типа double. Результат работы программы вывести на консоль.
f(x) = 1/3 + ln(|3x/5|) + min(10, x) + x^(x/4)

9. Рассчитать значение функции от нескольких переменных. Все переменные типа double. Результат работы программы вывести на консоль.
f(x, y, z) = 1/3 + ln(|y3/9|) - 7z/2 + min(20, x)

10. Рассчитать значения функции с определенным шагом. Реализовать с помощью цикла *for*.

$$f(x,k) = \frac{10k + x}{2}$$
$$x \in [0;10], \Delta x = 1,$$
$$k \in [0;10], \Delta k = 1$$

11. Рассчитать значение функции, используйте рекурсию.

$$f(x) = \sum_{k=1}^{n} \frac{1}{e^k}$$

12. Рассчитать значение функции, используйте рекурсию.

$$f(x) = \prod_{k=1}^{n} \frac{1}{e^k}$$

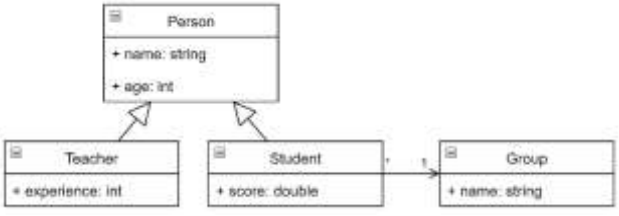
13. Рассчитать значение функции, используйте цикл.

$$f(x) = \sum_{k=1}^{n} \frac{1}{e^k}$$

14. Рассчитать значение функции, используйте цикл.

$$f(x) = \sum_{k=1}^{n} \frac{1}{e^k}$$

15. Реализовать библиотеку классов.



Для каждого класса должен быть конструктор. Для каждого класса должен быть определен метод ToString(), который выводит основные поля класса.

16. Дан массив, содержащий n элементов. Все четные элементы сложить, вывести массив и результат.

17. Вычислить и вывести сумму элементов матрицы A(n,n), расположенных на главной диагонали матрицы.

б) критерии оценивания компетенций (результатов):

Критерий оценки – правильность и полнота ответа на вопросы и решения задания.

в) описание шкалы оценивания:

Оценка за экзамен выставляется в баллах от 0 до 40 в равных долях за вопрос и задание. Экзамен считается сданным при оценке не ниже 60% от максимального балла.

8.2.2 Зачет

Зачет проводится в виде письменного ответа на 1 задание, с последующим устным собеседованием.

а) Типовые вопросы (задания):

1. Написать программу для вычисления значения функции. Переменные типа double.

$$y = \begin{cases} x + 5,\ x > 10 \\ 2x,\ 0 \le x \le 10 \\ x - 5,\ x < 0 \end{cases}$$

2. Рассчитать значение функции от одной переменной. Все переменные типа double. Результат работы программы вывести на консоль.
f(x) = 1/3 + 3x/7 + max(x, 3) + x^(x/3)

3. Рассчитать значение функции от нескольких переменных. Все переменные типа double. Результат работы программы вывести на консоль.
f(x, y, z) = 1/3 + 2y/7 - z/2 + min(z, x)

4. Рассчитать значения функции с определенным шагом. Реализовать с помощью цикла *for*.

$$f(x,k) = \frac{k-x}{3}$$
$$x \in [0;10], \Delta x = 1,$$
$$k \in [0;10], \Delta k = 1$$

5. Рассчитать значение функции, используйте рекурсию.

$$f(x) = \sum_{k=1}^{n} \frac{1}{e^k}$$

6. Рассчитать значение функции, используйте рекурсию.

$$f(x) = \prod_{k=1}^{n} \frac{1}{e^k}$$

7. Рассчитать значение функции, используйте цикл.

$$f(x) = \sum_{k=1}^{n} \frac{1}{e^k}$$

8. Рассчитать значение функции, используйте цикл.

$$f(x) = \sum_{k=1}^{n} \frac{1}{e^k}$$

9. Дана строка символов, состоящая и произвольного текста на английском языке, слова разделены пробелами. Поменять местами первую и последнюю буквы каждого слова.

10. В массиве из n целых чисел найти наибольший элемент и поменять его местами с первым элементом.

11. В массиве из n целых чисел найти наименьший элемент и поменять его местами с первым элементом.

12. Найти наибольший элемент главной диагонали матрицы A(n,n) и вывести всю строку, в которой он находится.

б) критерии оценивания компетенций (результатов):

Критерий оценки – правильность и полнота решения задания.

в) описание шкалы оценивания:

Оценка за зачет выставляется в баллах от 0 до 40.
Ответ на вопрос оценивается от 0 до 20 баллов.
Решение задания оценивается от 0 до 20 баллов.
Зачет считается сданным при оценке не ниже 60% от максимального балла.

6.2.3. Лабораторные работы

а) типовые задания - образец:

Лабораторная работа №1.1
Лабораторная работа включает изучение интерфейса и встроенных средств среды разработки Visual Studio или QT Creator, а также основы программирования на языке C#/C++. Студент получает практические навыки создания и конфигурирования проекта в среде Visual Studio или QT Creator (или Online Compiler языка Си#/СИ++). Загрузка исходного кода в репозиторий git.

Лабораторная работа №1.2
Лабораторная работа содержит 4 задания.
Пример:

Задание 1. Создать цифровую модель геометрической фигуры. Результат работы программы вывести на консоль.
Даны три стороны треугольника, найти углы треугольника. Даны три стороны треугольника, найти углы треугольника.
Задание 2. Рассчитать значение функции от одной переменной. Все переменные типа double. Результат работы программы вывести на консоль.
*f(x) = 1/21 + |3x/4| + 7*x/2 + min(2, x) + xcos(x/3) + x^(x/3)/3*
Задание 3. Рассчитать значение функции от нескольких переменных. Все переменные типа double. Результат
работы программы вывести на консоль.
f(x, y, z) = 1/21 + y/4 + 7z/2 + min(2, x)
Задание 4. Создать цифрового двойника. Определить атрибуты. Для каждого атрибута выбрать тип данных. Обосновать тип данных. Создать программу, в которой определены атрибуты объекта. Вывести на экран атрибуты объекта.
Комната

Лабораторная работа №1.3
содержит 4 задания. Задание №4 допускается выполнять в паре.
Пример:
Задание 1.
Написать программу для вычисления значения функции. Переменные типа double.

$$y = \begin{cases} x - 2,\ x > 2.5 \\ 1 + x,\ 0 \le x \le 2.5 \\ x + 120,\ x < 0 \end{cases}$$

Задание 2.
Написать программу для решения задачи. Проверить существуют ли такие x, y, z, для которых равенство будет верным. Все переменные типа bool.
f(x, y, z) = !x&y&z&z = true
Задание 3.
Написать программу для решения задачи, используя switch/case.
Пользователь задаёт число от 1 до 5 (присваивает его объявленной переменной). Программа должна вывести значение числа словом на экран.
Задание 4.
Написать программу для решения задачи.
Поле шахматной доски определяется парой натуральных чисел, каждое из которых не превосходит восьми: первое число — номер вертикали (при счете слева направо), второе — номер горизонтали (при счете снизу вверх). Даны натуральные числа a, b, c, d, каждое из которых не превосходит восьми. Определить, являются ли поля (a, b) и (c, d) полями одного цвета. Ответ проверить на шахматной доске или на клетчатой бумаге.

Лабораторная работа №1.4
Лабораторная работа содержит 3 задания.
Пример:
Задание 1.
Реализовать метод, который производит операцию над числами x и y по формуле

(a) и (b), после чего возвращает наибольшее из полученных значений. Входные параметры метода – два целых числа. Сделать перегрузку метода для параметров типа double. Сделать перегрузку метода для параметров типа char.

a. $5x + 3$

b. $-2y + 13/3$

Задание 2.

Реализовать метод для вычисления значения функции с помощью рекурсивного подхода.

$$f(n) = \sum_{k=1}^{n} \frac{k+1}{3}$$

Задание 3.

Реализовать метод для вычисления значения функции с помощью рекурсивного подхода.

$$f(n) = \prod_{k=1}^{n} \frac{k+1}{3}$$

Лабораторная работа №1.5

Лабораторная работа содержит 3 задания. Задание №3 допускается выполнять в паре.

Пример:

Задание 1.

Рассчитать значения функции с определенным шагом. Реализовать три варианта алгоритма для циклов: for, while и do/while.

$$f(x,k) = \frac{10x}{e^k}$$
$$x \in [0;10], \Delta x = 1,$$
$$k \in [0;1], \Delta k = 0.1$$

Задание 2.

Найти сумму и произведение ряда, используйте цикл.

$$f(x) = \sum_{k=1}^{n} \frac{10x}{e^k}$$
$$f(x) = \prod_{k=1}^{n} \frac{10x}{e^k}$$

Задание 3.

Из натурального числа удалить заданную цифру. Число и цифру вводить с клавиатуры. Например, задано число 5683. Требуется удалить из него цифру 8. Получится число 563.

Лабораторная работа №1.6

Лабораторная работа содержит 2 задания.

Пример:

Задание 1
В массиве из 20 целых чисел найти наибольший элемент и поменять его местами с первым элементом. Вывести результаты работы на экран.
Задание 2
Дана матрица A(n, m). Найти наименьший элемент в каждой строке матрицы. Вывести результаты работы на экран. n, m – натуральные числа.

Лабораторная работа №1.7

Лабораторная работа содержит 2 задания. В рамках лабораторной работы студент реализует решение одной и той же задачи двумя различными подходами. В первом случае входные данные представлены массивом символов char[]. Во втором случае входные данные представлены строкой string.
Пример:
Задание 1.
Посчитать в строке количество слов.
Задание 2.
Дана строка символов, содержащая некоторый текст на русском языке. Заменить все большие буквы «А» на символ «*».

Лабораторная работа №1.8

Лабораторная работа содержит 1 задание. Необходимо создать библиотеку классов и консольное приложение. Библиотека классов должна содержать классы предметной области. Предметная область выбирается студентом по согласованию с преподавателем. Библиотека классов должна содержать классы, удовлетворяющие минимальным требованиям.

1) Библиотека классов должна включать: класс, статический класс, структура.
2) Код должен содержать комментарии.
3) Хотя бы один класс должен содержать 3 поля, 3 свойства, 3 метода, перегрузку методов, иметь интерфейс, содержать конструкторы с параметрами, определены модификаторы доступа и их обоснования, константные поля, переопределить метод ToString().

В консольном приложении демонстрируется возможности библиотеки классов.
При выполнении лабораторной работы в группе, на каждого студента приходится минимум три класса.

Лабораторная работа №1.9

Лабораторная работа содержит 3 задания. В каждом задании необходимо реализовать загрузку и сохранение информации о работе программы с помощью определенного подхода. В качестве программы рекомендуется использовать кодовую базу предыдущих лабораторных работ.
Задание 1.
С помощью класса File.
Задание 2.
С помощью классов потока StreamReader, StreamWriter.
Задание 3.
С помощью сериализации.

Лабораторная работа №2.1

В данной работе требуется реализовать универсальный класс предметной области. Класс должен отвечать следующим характеристикам.

1. Класс должен содержать, описывающие его, свойства (не менее 5).
2. Класс должен содержать ограничения на тип универсального шаблона.
3. Класс должен содержать три метода, которые принимают в качестве аргумента тип универсального шаблона.

Продемонстрировать работу класса с помощью консольного приложения.

Лабораторная работа №2.2

В лабораторной работе требуется реализовать конструкции делегатов и событий для вашего приложения.

1. Реализовать функцию сохранения данных при закрытии программы. Используйте событие Console.CancelKeyPress.
2. Реализовать класс предметной области, который выдает событие. Для удобной передачи аргументов создайте класс NameEventArgs.
3. Реализовать класс, реагирующий на событие.

Продемонстрировать работу класса с помощью консольного приложения.

Лабораторная работа №2.3

Лабораторная работа содержит 2 задания. В первом задании требуется по пунктам произвести требуемые операции. Во втором задании на основе таблицы создать класс и реализовать требуемый функционал.

Пример:

Задание 1.

1. Создать *список A* экземпляров класса *автомобиль*.
2. Инициализировать *список* случайными значениями класса *автомобиль* (10 экземпляров).
3. Вывести *список A* на экран.
4. Добавить в *список* случайный экземпляр класса *автомобиль*.
5. Создать *список B* класса *автомобиль* (5 экземпляров).
6. Вывести *список B* на экран.
7. Добавить к элементам первого списка элементы второго.
8. Вывести элементы *списка* на экран.
9. Создать случайный экземпляр класса *автомобиль*, записать ссылку экземпляр, поместить экземпляр в случайное место списка.
10. Создать случайный экземпляр класса *автомобиль* и вставить его в начало списка.
11. Вывести список AB на экран.
12. Создать *список C* класса *автомобиль* (5 экземпляров).
13. Вставить *список C* в *список AB* в случайный диапазон значений.
14. Удалить экземпляр, созданный на шаге 9.
15. Вывести *список ABC* на экран.
16. Удалить диапазон значений из списка с 3 элемента до 6.
17. Вывести *список ABC* на экран.

18. Удалить все элементы из *списка*.
19. Вывести *список* на экран.

Задание 2
Класс: Car

Name	Code	Owner
Nissan	vbUHVjazD6	Эдуард Устинов
Fiat	MmSKTicFnu	Лаврентий Евсеев
KIA	2KUTI	Лука Баранов
Audi	e8aBaPZvyG	Оксана Абрамова
BMW	5wWIeyeUD	Владимир Кононов
Lada	mTyJIaYFTk	Валерий Зайцев
Lada	WaKUoI8chO	Ольга Тарасова
Toyota	EbUdr58oCv	Лариса Шарапова
KIA	vMVV	Антон Цветков
KIA	HQAjigXhoq	Никита Павлов
Nissan	210FjKzS3	Кира Захарова
Nissan	tW9JcKoFZB	Даниил Матвеев
Opel	r9s9Stwc1c	Андрей Богданов
Opel	7xqI8O7i0a	Владислав Беляев
УАЗ	J2OoCfJqXt	Михаил Юдин
Lada	XfQyST2fgB	Георгий Кудрявцев
BMW	O1SFHhkGNN	Валерий Алексеев
Audi	KUClJvp3vz	Эдуард Дементьев
Ford	cd6ZjjJHs	Роман Борисов
Ford	wuVUlQicr	Степан Пестов

1. Вывести на экран все уникальные марки в наборе данных.
2. Пользователь задает свойство Code. Проверить существует ли запись в наборе данных.
3. Пользователь задает свойство Name. Вывести на экран количество записей с заданным свойством.
4. Пользователь задает свойство Name. Вывести на экран записи с выбранным свойством.
5. Пользователь задает последовательность символов Code. Вывести на экран записи, Code которых содержит последовательность.

Лабораторная работа №2.4
Лабораторная работа содержит 2 задания.
Задание 1.
Реализовать программу для обработки элементов вектора. В качестве обработки выбирается математическое преобразование элементов вектора, например – возведение в степень.
Задание 2.
Реализовать программу с вызовом асинхронных методов. Реализовать три асинхронных метода, например, расчет функции, вывод системного времени. Пользователь вызывает методы из консоли. Работа методов выводится в консоль.

Лабораторная работа №2.5
В лабораторной работе требуется создать настольное приложение. Тематика приложение согласуется с преподавателем. В приложении должен быть реализован функционал ввода и вывода информации. Изменение графического интерфейса при взаимодействии с пользователем. Прикладной функционал для решения задачи в рамках выбранной тематики. Студент может по согласованию с преподавателем выбрать фреймворк для разработки настольного приложения.

б) критерии оценивания компетенций (результатов):

По завершению лабораторной работы в ходе устного опроса перед компьютером студент демонстрирует: код программы, объясняет основные проблемы, с которыми пришлось столкнуться и методы их решения, показывает результаты работы программы, отвечает на вопросы преподавателя.

в) описание шкалы оценивания:

Выполненная лабораторная работа оценивается согласно пункту (6.3).

8.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Рейтинговая оценка знаний является интегральным показателем качества теоретических и практических знаний и навыков студентов по дисциплине и складывается из оценок, полученных в ходе текущего контроля и промежуточной аттестации.

Текущий контроль в семестре проводится с целью обеспечения своевременной обратной связи, для коррекции обучения, активизации самостоятельной работы студентов.

Промежуточная аттестация предназначена для объективного подтверждения и оценивания достигнутых результатов обучения после завершения изучения дисциплины.

Текущий контроль осуществляется два раза в семестр: контрольная точка № 1 (КТ № 1) и контрольная точка № 2 (КТ № 2).

Результаты текущего контроля и промежуточной аттестации подводятся по шкале балльно-рейтинговой системы.

Семестр 1:

Вид контроля	**Этап рейтинговой системы Оценочное средство**	**Балл**	
		Минимум	Максимум
Текущий	**Контрольная точка № 1**	18	32
	Лабораторная работа №1.1	4	6
	Лабораторная работа №1.2	4	6
	Лабораторная работа №1.3	4	6
	Лабораторная работа №1.4	4	7
	Лабораторная работа №1.5	4	7
	Контрольная точка № 2	16	28
	Лабораторная работа №1.6	4	7
	Лабораторная работа №1.7	4	7
	Лабораторная работа №1.8	4	7
	Лабораторная работа №1.9	4	7
Промежуточный	Зачет с оценкой	24	40
	Зачет с оценкой	24	40
ИТОГО по дисциплине		60	100

Семестр 2:

Вид контроля	**Этап рейтинговой системы Оценочное средство**	**Балл**	
		Минимум	Максимум
Текущий	**Контрольная точка № 1**	21	36
	Лабораторная работа №2.1	7	12
	Лабораторная работа №2.2	7	12
	Лабораторная работа №2.3	7	12
	Контрольная точка № 2	15	24
	Лабораторная работа №2.4	6	10
	Лабораторная работа №2.5	9	14
Промежуточный	**Экзамен**	24	40
	Экзамен	24	40
ИТОГО по дисциплине		60	100

Бонусы: поощрительные баллы студент получает к своему рейтингу в конце семестра за активную и регулярную работу на занятиях, за активную работу на практических занятиях.

По Положению бонус (премиальные баллы) не может превышать **5 баллов**.

Штрафы: за несвоевременную сдачу лабораторных работ максимальная оценка может быть снижена на 40%.

Методические материалы, определяющие процедуры оценивания:

Процедура оценивания знаний, умений, владений по дисциплине включает учет успешности по всем видам заявленных оценочных средств.

Проверка освоения материала практических занятий проводится на лабораторных работах в процессе защиты лабораторной работы. Студент отвечает на вопросы преподавателя по предыдущим разделам в устной или письменной форме. Результат ответов на вопросы влияет на общую оценку за защиту лабораторной работы.

Варианты лабораторных работ распределяются на первом занятии, вариант лабораторной работы может быть изменен по согласованию с преподавателем. Лабораторные работы защищаются в установленные преподавателем сроки.

Устный опрос проводится на каждом практическом занятии и затрагивает как тематику прошедшего занятия, так и лекционный материал. Применяется групповое оценивание ответа или оценивание преподавателем.

По окончании освоения дисциплины в первом семестре проводится промежуточная аттестация в виде зачета с оценкой, что позволяет оценить совокупность приобретенных в процессе обучения компетенций. При выставлении итоговой оценки применяется балльно-рейтинговая система оценки результатов обучения.

По окончании освоения дисциплины проводится промежуточная аттестация в виде экзамена, что позволяет оценить совокупность приобретенных в процессе обучения компетенций. При выставлении итоговой оценки применяется балльно-рейтинговая система оценки результатов обучения.

Экзамен предназначен для оценки работы обучающегося в течение всего срока изучения дисциплины и призван выявить уровень и систематичность полученных обучающимся теоретических знаний, приобретенных навыков самостоятельной работы.

Оценка сформированных компетенций на экзамене для тех обучающихся, которые пропускали занятия и не участвовали в проверке компетенций во время изучения дисциплины, проводится после индивидуального собеседования с преподавателем по пропущенным или не усвоенным обучающимся темам с последующей оценкой самостоятельно усвоенных знаний на экзамене.

9. Перечень учебной литературы, необходимой для освоения дисциплины

1. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 285 с. — (Высшее образование). — ISBN 978-5-534-16031-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/560815
2. Казанский, А. А. Программирование на C# : учебное пособие для вузов / А. А. Казанский. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 181 с. — (Высшее образование). — ISBN 978-5-534-21381-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/569864
3. Огнева, М. В. Программирование на языке C++: практический курс : учебник

для вузов / М. В. Огнева, Е. В. Кудрина, А. А. Казачкова. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 342 с. — (Высшее образование). — ISBN 978-5-534-18949-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/563618

4. Подбельский, В. В. Программирование. Базовый курс C# : учебник для вузов / В. В. Подбельский. — Москва : Издательство Юрайт, 2025. — 369 с. — (Высшее образование). — ISBN 978-5-534-10616-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/560848
5. Программирование: математическая логика : учебник для вузов / М. В. Швецкий, М. В. Демидов, А. В. Голанова, И. А. Кудрявцева. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 675 с. — (Высшее образование). — ISBN 978-5-534-11009-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/565921

10. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее - сеть «Интернет»), необходимых для освоения дисциплины

1. Документация по C#/C++: [Официальные сайты] — https://docs.microsoft.com/ru-ru/dotnet/csharp/ и https://docs.microsoft.com/ru-ru/dotnet/c++/ .
2. http://elibrary.ru/ – Научная электронная библиотека (свободный доступ).

11. Методические указания для обучающихся по освоению дисциплины

Вид учебного занятия	Организация деятельности студента
Практические занятия	Перед практическими занятиями необходимо ознакомиться с основной литературой. На практическом занятие активно участвовать в решении задач, предлагать альтернативные варианты решения, задавать вопросы по теме.
Лабораторная работа	При выполнении лабораторных работ необходимо ориентироваться на практические занятия и рекомендуемую литературу. Лабораторная работа считается выполненной после ее успешной защиты, включающей: – демонстрацию на компьютере решаемой задачи с разъяснением разработанного программного кода и демонстрацией выполнения;

	– собеседование с преподавателем для выявления уровня освоения теоретических основ.
Подготовка к зачету	При подготовке к зачету необходимо ориентироваться на материалы практических занятий и лабораторные работы, а также рекомендуемую литературу.
Подготовка к экзамену	При подготовке к экзамену необходимо ориентироваться на материалы практических занятий и лабораторные работы, а также рекомендуемую литературу.

12. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)

12.1. Перечень информационных технологий

– Компьютерное тестирование по итогам изучения разделов дисциплины.
– Проверка домашних заданий и консультирование посредством электронной почты.
– Использование электронных презентаций при проведении практических занятий.

12.2. Перечень программного обеспечения

1. Kaspersky Endpoint Security для образовательных учреждений, договор №1322эа от 27.10.2020.
2. Web-браузер (Яндекс, Internet Explorer, Google Chrome, Mozilla Firefox, Opera или любой другой) (Свободно распространяемое ПО).
3. Среда программирования Visual Studio Code [свободный доступ: https://code.visualstudio.com/]/ Visual Studio 2019 / Visual Studio 2013 / Visual Studio 2022 [свободный доступ: https://visualstudio.microsoft.com/ru/].
4. Среда программирования QT creartor.
5. Online Compiler языка Си#/СИ++ (свободный доступ).
6. Программное обеспечение для работы с git SourceTree [свободный доступ: https://www.sourcetreeapp.com/]/ [свободный доступ: https://desktop.github.com/] / расширение Visual Studio для git .

12.3. Перечень информационных справочных систем

Доступ к электронным библиотечным ресурсам и электронной библиотечной системе (ЭБС) осуществляется посредством специальных разделов на официальном сайте ИАТЭ НИЯУ МИФИ. Обеспечен доступ к электронным каталогам библиотеки ИАТЭ НИЯУ МИФИ, а также электронным

образовательным ресурсам (ЭИОС), сформированным на основании прямых договоров с правообладателями учебной и учебно-методической литературы, методических пособий:

1. ЭБС «Айбукс», Договор № 05-25-910 от 2.06.2025 на оказание услуг по предоставлению доступа к электронно-библиотечной системе «Айбукс.ру/ibooks.ru». Срок действия: 1.09.2025г. по 31.08.2026г.
2. ЭБС «ЛАНЬ», Договор № 08-25-910 от 30.05.2025 на оказание услуг по предоставлению доступа к базе данных ЭБС «Лань». Срок действия: с 1.09.2025г. по 31.08.2026г.
3. ЭБС «ЮРАЙТ», Договор № 09-25-910 от 3.06.2025 на оказание услуг по предоставлению доступа к «Образовательной платформе ЮРАЙТ». Срок действия: с 1.09.2025г. по 31.08.2026г.
4. ЭБС «Консультант студента», Договор № 06-25-910 от 3.06.2025 на оказание услуг по предоставлению доступа к комплектам «Медицина, Здравоохранение», «Книги издательства «Феникс», «Издательский дом МЭИ», «Книги издательства «Проспект»: «Иностранные языки», «Естественные науки», «Экономика и управление», «Гуманитарные науки», «Юридические науки», входящим в базу данных «Электронная библиотека технического вуза». (ЭБС «Консультант студента»). Срок действия: 1.09.2025г. по 31.08.2026г.
5. Научная электронная библиотека E-library.ru, Договор № SU-353/2025 от 2.11.2024 на доступ к электронным изданиям в составе базы данных «НАУЧНАЯ ЭЛЕКТРОННАЯ БИБЛИОТЕКА eLIBRARY.RU». Срок действия: с 1.01.2025г. по 31.12.2025г.

13. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

• Компьютерный класс сетевых технологий. Класс оснащен 10 компьютерами (Intel Core i5/8GB/1 TB) и 1 компьютером (Intel Celeron 1.6 GHz, 2 GB RAM, 250 GB) с операционной системой Linux, а также мультимедийным проектором. Есть доступ к Wi-Fi.
• Аудиторный класс, оборудованный проекционным экраном, мультимедийным проектором и персональным компьютером (AMD, ATHLON64, 2.7 GHz, 4 GB RAM, 250 GB). Есть доступ к Wi-Fi.

14. Иные сведения и (или) материалы

14.1. Перечень образовательных технологий, используемых при осуществлении образовательного процесса по дисциплине

Практические занятия проходят с обсуждением учебного материала, демонстрируемого в форме презентаций на экране с использованием мультимедиа-проектора. Все лабораторные занятия проводятся в интерактивной форме при тесном контакте студентов с преподавателем.

В рамках лабораторных работ студенты выполняют лабораторные работы, призванные дать представление о возможностях применения программирования, как инструментария для решения самых разнообразных практических задач. Лабораторные работы проводятся при активном взаимодействии студентов и преподавателя, в ходе которого обсуждаются детали создания проекта задачи, проблемы и ошибки, возникающие на всех этапах их разработки, проводится проверка корректности полученных результатов.

14.2. Формы организации самостоятельной работы обучающихся (темы, выносимые для самостоятельного изучения; вопросы для самоконтроля; типовые задания для самопроверки

На самостоятельное изучение студентам предлагается более глубоко рассмотреть темы, кратко затрагиваемые в лекционных курсах. Контроль освоения материала осуществляется в ходе приема лабораторных работ и в рамках экзамена по дисциплине.

№	Тема	Часть, осваиваемая самостоятельно
1.1.	Введение	Репозиторий git. Сервисы Gitlab, Github, Bitbucket. Язык разметки Markdown.
1.2.	Примитивные типы данных	Типы данных: byte, sbyte, short, ushort, uint, long, ulong, float, decimal. Тип DateTime.
1.3.	Ветвление	switch/case охранные условия. Возвращение значения из switch Оператор go to.
1.4.	Методы	Параметры out. Именованные аргументы.
1.5.	Итерации	Оператор foreach.
1.6.	Символьные типы данных	Форматирование строк
1.7.	Массивы	Нерегулярные массивы

№	Тема	Часть, осваиваемая самостоятельно
1.8.	Работа с символьными типами данных	Синтаксический анализ и форматирование. Нормализация.
1.10.	Файлы	Xml сериализация. Бинарная сериализация.
2.1.	Универсальные шаблоны	Ковариантность и контрвариантность в универсальных шаблонах
2.2.	Делегаты, события	Ковариантность и контравариантность делегатов
2.3.	Коллекции	Классы System.Collections Сортировка коллекции
2.4	Асинхронность и параллелизм	Асинхронные потоки с IAsyncEnumerable<T>, Задачи параллельного программирования
2.5	Настольные приложения с графическим интерфейсом	Платформы для создания настольных графических приложений.

Контроль освоения самостоятельно изученного теоретического материала осуществляется в виде собеседования во время защиты лабораторных, в виде устного опроса на экзамене.

Кроме этого, студенты также самостоятельно выполняют большую часть предусмотренных практических работ, промежуточный результат которых представляется на лабораторных занятиях, а конечный результат - на защите лабораторных работ.

Программу составила:

О.А.Цветкова, старший преподаватель отделения интеллектуальных кибернетических систем