

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Бойко Валерий Леонидович
Должность: Ректор
Дата подписания: 16.01.2025 19:33:48
Уникальный программный ключ:
1ae60504b2c916e8fb686192f29d3bf1653db777



**Высшая Школа
Управления**

Негосударственное образовательное частное учреждение высшего
образования «Высшая школа управления» (ЦКО)
(НОЧУ ВО «Высшая школа управления» (ЦКО))

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.О.12 Программирование

Направление подготовки

38.03.05

«Бизнес-информатика»

Направленность (профиль) подготовки

Информационные системы в бизнесе

Квалификация выпускника

«Бакалавр»

Форма обучения

заочная

Рабочая программа рассмотрена
на заседании кафедры
цифровой экономики и управления и
государственного администрирования
«28» августа 2024, протокол №1

Заведующий кафедрой д.э.н., доцент
Н.Р. Куркина

г. Москва, 2024

Рабочая программа дисциплины «Программирование» составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 38.03.05 Бизнес-информатика, утвержденного приказом Министерства образования и науки Российской Федерации № 838 от 20 июля 2020 года (зарегистрирован в Минюсте России 19 августа 2020 г. № 59325).

Организация-разработчик: НОЧУ ВО «Высшая школа управления» (ЦКО)

Разработчик: _____

Содержание

1. Цели и задачи освоения дисциплины	5
2. Место дисциплины в структуре образовательной программы	5
3. Планируемые результаты обучения	6
4. Структура и содержание дисциплины (модуля)	8
4.1 Объем дисциплины и виды учебной работы	8
4.2 Тематический план дисциплины	9
4.3 Содержание дисциплины	11
4.4. Практическая подготовка	15
5. Учебно-методическое, информационное и материально-техническое обеспечение дисциплины	15
5.1 Основная литература	15
5.2 Дополнительная литература	16
5.3 Профессиональные базы данных и информационные справочные системы	16
5.4 Материально-техническое и программное обеспечение (лицензионное и свободно распространяемое)	16
6. Методические указания для обучающихся по освоению дисциплины	17
6.1 Занятия лекционного и семинарского (практического) типов	17
6.2. Самостоятельная работа студентов	18
7. Методические рекомендации по обучению лиц с ограниченными возможностями здоровья и инвалидов	19
Приложение 1. Фонд оценочных средств	22
1. Паспорт фонда оценочных средств	23
2. Оценочные средства	24
2.1 Текущий контроль	24
2.2 Промежуточная аттестация	27

1. Цели и задачи освоения дисциплины

Цель дисциплины: сформировать у обучающихся базовые знания по теории программирования и технологии разработки программ для ЭВМ; сформировать умение применять инструмент, позволяющий разрабатывать программное обеспечение для ЭВМ в выбранной предметной области, соответствующей объекту профессиональной деятельности по направлению подготовки и моделировать процессы и компоненты вычислительных систем и сетей.

Задачи дисциплины:

- сформировать понимание методологии и основных принципов процедурного и объектно-ориентированного подходов к программированию;
- сформировать навыки разработки программ в интегрированных инструментальных средах.

2. Место дисциплины в структуре образовательной программы

Дисциплина «Программирование» относится к дисциплинам обязательной части блока Б1 «Дисциплины (модули)» учебного плана, согласно ФГОС ВО для направления подготовки 38.03.05 Бизнес-информатика.

3. Планируемые результаты обучения

Код и наименование компетенции	Код и наименование индикатора достижения компетенции (ИДК)	Планируемые результаты обучения
ОПК-1 Способен проводить моделирование, анализ и совершенствование бизнес-процессов и информационно-технологической инфраструктуры предприятия в интересах достижения его стратегических целей с использованием современных методов и программного инструментария	ИОПК-1.1 Знать: основы моделирования, анализа и совершенствования бизнес-процессов и информационно-технологической инфраструктуры предприятия в интересах достижения его стратегических целей с использованием современных методов и программного инструментария ИОПК-1.2 Уметь: проводить моделирование, анализ и совершенствование бизнес-процессов и информационно-технологической инфраструктуры предприятия в интересах достижения его стратегических целей с использованием современных методов и программного инструментария ИОПК-1.3 Владеть: навыками моделирования, анализа и совершенствования бизнес-процессов и информационно-технологической инфраструктуры предприятия в интересах достижения его стратегических целей с использованием современных методов и программного инструментария	Знать: • Основы моделирования, анализа и совершенствования бизнес-процессов и информационно-технологической инфраструктуры предприятия для достижения стратегических целей с использованием современных методов и программных инструментов • Методы исследования и анализа рынка информационных систем и информационно-коммуникационных технологий, а также принципы выбора рациональных решений для управления бизнесом • Основы управления процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, включая разработку алгоритмов и программ для их практической реализации Уметь: • Проводить моделирование, анализ и совершенствование бизнес-процессов и информационно-технологической инфраструктуры предприятия для достижения стратегических целей с использованием современных методов и программных инструментов
ОПК-2 Способен проводить исследование и анализ рынка информационных систем и информационно-коммуникационных технологий, выбирать рациональные решения для управления бизнесом	ИОПК-2.1 Знать: методы исследования и анализа рынка информационных систем и информационно-коммуникационных технологий, выбирать рациональные решения для управления бизнесом. ИОПК-2.2 Уметь: проводить исследование и анализ рынка информационных систем и информационно-коммуникационных технологий, выбирать рациональные решения для управления бизнесом. ИОПК-2.3 Владеть: навыками исследования и анализа рынка информационных систем и информационно-коммуникационных технологий, выбирать рациональные решения для управления бизнесом.	• Проводить исследование и анализ рынка информационных систем и информационно-коммуникационных технологий, выбирать рациональные решения для управления бизнесом • Управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, разрабатывать алгоритмы и программы для их практической реализации Владеть: • Навыками моделирования, анализа и

ОПК-3 Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы программы для их практической реализации	ИОПК-3.1 Знать: основы управления процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации ИОПК-3.2 Уметь: управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации ИОПК-3.3 Владеть: навыками управления процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	совершенствования бизнес-процессов и информационно-технологической инфраструктуры предприятия для достижения стратегических целей с использованием современных методов и программных инструментов • Навыками исследования и анализа рынка информационных систем и информационно-коммуникационных технологий, способностью выбирать рациональные решения для управления бизнесом • Навыками управления процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, включая разработку алгоритмов и программ для их практической реализации
---	--	--

4. Структура и содержание дисциплины (модуля)

4.1 Объем дисциплины и виды учебной работы

Виды учебной работы	Объем в часах
Общая трудоемкость дисциплины	180 (5 зачетных единицы)
Контактная работа обучающихся с преподавателем (всего)	26
Аудиторная работа (всего), в том числе:	26
Лекции	10
Семинары, практические занятия	16
Лабораторные работы	
Внеаудиторная работа (всего):	154
в том числе: консультация по дисциплине	
Самостоятельная работа обучающихся (всего)	154
Вид промежуточной аттестации обучающегося	Экзамен

4.2 Тематический план дисциплины

Наименование разделов и тем	С е м е с т р	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)							Компетенции
		Всего	Из них аудиторные занятия			Самостоятельная работа	Курсовая работа	Контрольная работа	
			Ле кц ии	Лаб ора тор ные раб оты	Пра кти чес кие /сем ина рск ие зан яти я				
Тема 1. Введение в язык C++	3	12	2		2	8			ОПК-1, ОПК-2, ОПК-3
Тема 2. Массивы, указатели и ссылки	3	12	2		2	8			ОПК-1, ОПК-2, ОПК-3
Тема 3. Операторы управления потоком	3	12	2		2	8			ОПК-1, ОПК-2, ОПК-3
Тема 4. Символьные массивы и строки	3	12	2		2	8			ОПК-1, ОПК-2, ОПК-3
Тема 5. Типы, задаваемые пользователем	3	12	2		2	8			ОПК-1, ОПК-2, ОПК-3
Тема 6. Функции C/C++	3	12			2	10			ОПК-1, ОПК-2, ОПК-3
Тема 7. Классы	3	12			2	10			ОПК-1, ОПК-2, ОПК-3
Тема 8. Перегрузка операций	3	12			2	10			ОПК-1, ОПК-2, ОПК-3

Тема 9. Наследование функциональности	3	12				12			ОПК-1, ОПК-2, ОПК-3
Тема 10. Виртуальные функции и полиморфизм	3	12				12			ОПК-1, ОПК-2, ОПК-3
Тема 11. Шаблоны и библиотека стандартных шаблонов	3	10				10			ОПК-1, ОПК-2, ОПК-3
Тема 12. Основы языка C#	3	10				10			ОПК-1, ОПК-2, ОПК-3
Тема 13. Классы и структуры C#	3	10				10			ОПК-1, ОПК-2, ОПК-3
Тема 14. Наследование C#	3	10				10			ОПК-1, ОПК-2, ОПК-3
Тема 15. Делегаты и события	3	10				10			ОПК-1, ОПК-2, ОПК-3
Тема 16. Коллекции. Основы разработки интерфейса приложений	3	10				10			ОПК-1, ОПК-2, ОПК-3
Итого		180	10		16	154			

4.3 Содержание дисциплины

Тема 1. Введение в язык C++

Язык программирования C++, характеристика, назначение. Классификация языков программирования: машинный язык, язык ассемблера, языки высокого уровня. Понятие программы, исходная, объектная и исполняемая программы. Трансляторы: ассемблеры и компиляторы. Элементы языка C++: алфавит, константы, идентификаторы, ключевые слова, комментарии. Основные этапы разработки программы. Принципы структурного программирования. Возможности, предоставляемые интегрированной средой разработки программ (IDE) – Visual Studio. Директивы препроцессора: включение файлов (`#include`) и макроподстановки (`#define`). Базовые (простые) типы данных C++. Приведение типов данных. Определение (создание) переменных различных типов. Константы (литералы): десятичные, шестнадцатеричные, символьные, с плавающей точкой, строковые. Ввод / вывод содержимого переменных на консоль. Консольный ввод / вывод с использованием библиотеки ввода-вывода (функции `printf` и `scanf`) и высокоуровневый ввод-вывод с использованием объектов `cin` и `cout`. Операции арифметические, отношения, логические, побитовые. Алгоритмические конструкции ветвления и цикла на примере операторов `if` и `while`. Отладка программ в интегрированной среде Visual Studio: точки останова, просмотр промежуточного содержимого переменных программы, пошаговый режим отладки. Область видимости и время жизни переменных, понятие блока. Понятие функции в языке C++. Правила работы с функциями: прототип, определение, вызов. Создание простейших функций. Главная функция. Передача аргументов главной функции.

Тема 2. Массивы, указатели и ссылки

Понятие массива, возможности, определяемые этим типом данных, при обработке информации Синтаксис объявления массива в языке C++. Инициализация массива при его создании. Понятие индекса - последовательного номера элемента массива. Доступ к элементам массива через операцию индексации. Многомерные массивы и их размещение в памяти компьютера. Структурная и бесструктурная инициализация многомерных массивов. Назначение типа данных указатель. Синтаксис объявления переменной типа указатель. Использование операции адресации для инициализации указателя. Доступ к содержимому переменной, адрес которой хранится в указателе. Операции над указателями: присваивание адреса, доступ по указателю (разыменование), увеличение/уменьшение указателя, сравнение указателей, индексация указателей. Связь массивов и указателей, имя массива – указатель на его первый элемент. Массивы указателей. Указатели и динамическое распределение памяти, операторы `new` и `delete`. Тип данных ссылка. Создание переменной типа ссылка. Три способа использования ссылки: - передача, как параметра функции, -

ссылка, в качестве возвращаемого значения функции, - независимая ссылка.

Тема 3. Операторы управления потоком

Оператор - единица выполнения программы. Простые и составные операторы (блоки). Условный оператор if, синтаксис и логика работы. Вложенные операторы if. Выбор одного из двух - конструкция if – else. Множественный выбор – конструкция if – else if - ... - else. Организация множественного выбора при помощи оператора switch. Синтаксис конструкции switch и порядок выполнения операторов. Операторы организации циклов. Оператор цикла for. Заголовок for, логика работы оператора. Логика работы оператора for при отсутствии любого или всех выражений в его заголовке. Операторы цикла с предусловием while и постусловием do-while, синтаксис и логика работы. Варианты предпочтительного использования операторов цикла в зависимости от решаемой задачи. Вложенные циклы. Операторы прерывания блока: break, continue, return. Использование этих операторов для досрочного завершения циклов. Оператор безусловного перехода goto.

Тема 4. Символьные массивы и строки

Определение строки символов. Способы инициализации строк. Представление строки в памяти, символ конца строки. Строковые литералы, обработка компилятором строкового литерала. Использование признака завершения строки для определения длины строки, копирования, присваивания, объединения строк. Считывание строк с клавиатуры. Библиотечные функции для работы со строками, примеры использования функций для обработки строк. Массивы строк. Хранения символьных строк различного размера в массиве указателей.

Тема 5. Типы, задаваемые пользователем

Структуры: назначение, шаблон и объявление. Создание структурных переменных. Инициализация структур. Доступ к элементам структуры через операцию точка и через указатель. Определение в качестве элементов структуры битовых полей. Передача структурных переменных, как параметров функциям. Объединения, синтаксис, специфика использования подобных объектов. Доступ к элементам объединения. Перечисляемый тип данных, определение и использование перечисляемого типа Переименование типов – инструкция typedef.

Тема 6. Функции C/C++

Выделение группы инструкций из основной программы в функцию. Синтаксис создания функций в языке C++ - определение функции. Прототип функции. Передача параметров функции и возврат функцией значения. Перегрузка функций. Рекурсивный вызов функции. Механизм вызова функции и возврат из нее через стек. Особенности передачи параметров функциям по значению и по адресу (указателю, ссылке). Передача в

функцию массивов. Указатели на функции. Косвенный вызов функции через указатель. Возможности языка C++, поддерживающего указатели на функции (передача функции в качестве аргумента в другие функции, хранение функций в массивах и структурах). Использование указателей на функции в практических примерах

Тема 7. Классы

Классы, как типы, определяемые пользователем Синтаксис объявления класса, члены класса, модификаторы доступа к членам класса. Создание объектов класса. Область видимости членов класса. Указатель `this`. Инкапсуляция в контексте понятия класса. Инициализация объектов, конструкторы класса. Уничтожение объектов при выходе их из области видимости, деструкторы. Перегрузка конструкторов: конструкторы без параметров, конструкторы с параметрами, конструкторы копии. Конструктор копирования по умолчанию. Особенности использования конструкторов копии. Присваивание объектов. Передача объектов функциям. Объекты, в качестве возвращаемых значений функций. Создание массивов объектов. Использование указателей на объекты. Динамическое размещение объектов в памяти.

Тема 8. Перегрузка операций

Перегрузка операций как форма полиморфизма. Назначение перегрузки операций. Синтаксис функций-членов класса, осуществляющих перегрузку операторов. Перегрузка бинарных операций: арифметических операций, операций отношения, логических операций. Перегрузка унарных операций. Особенность оператора присваивания, случаи необходимости перегрузки оператора присваивания. Перегрузка операции индексации.

Тема 9. Наследование функциональности

Механизм наследования поведения класса. Базовый и производный классы. Создание производного класса. Доступ из производного класса к членам базового класса. Защищенные члены класса. Конструкторы производных классов, передача аргументов базовому классу. Иерархия наследования. Сравнение концепции наследования и концепции контейнерных классов (наследование и оболочка объекта).

Тема 10. Виртуальные функции и полиморфизм

Особенность указателей на базовый класс в языке C++. Создание виртуальных функций, ключевое слово `virtual`. Виртуальные функции, полиморфические кластеры и позднее связывание. Чистые виртуальные функции и абстрактные классы. Техническая реализация виртуальных функций. Преимущества позднего связывания.

Тема 11. Шаблоны и библиотека стандартных шаблонов

Функция шаблон (родовая функция), назначение. Определение функции-шаблона, примеры. Различие родовых и перегруженных функций. Родовые классы – шаблоны

классов. Объявление родового класса. Случаи применения шаблонов класса. Создание различных шаблонов классов. Определение понятия контейнер, типы контейнеров (классы-контейнеры). Наиболее востребованные классы-контейнеры: векторы, списки, ассоциативные списки (назначение и возможности, наиболее важные функции-члены, доступ к элементам контейнеров, создание объектов). Строковый класс и возможности работы со строками.

Тема 12. Основы языка C#

Типы данных C#: типы значений и ссылочные типы, концептуальная разница этих типов. Создание и инициализация переменных типа значений (простые типы) и ссылочных типов (массивов и строк). Предопределенные (встроенные) типы значений. Встроенные ссылочные типы: `object` и `string`, краткая характеристика. Консольный ввод-вывод. Операторы управления потоком исполнения программы: условные операторы, операторы цикла: `while`, `for`, `do-while`, `foreach`. Операторы перехода: `goto`, `break`, `continue`, `return`. Перечисления. Массивы. Организация взаимосвязанных классов (пространства имен и оператор `using`).

Тема 13. Классы и структуры C#

Объявление класса. Данные-члены и функции-члены (конструкторы, методы, операции, свойства, индексы). Создание экземпляров класса. Объявление методов. Доступ к членам класса. Модификаторы доступа к членам класса. Передача параметров в методы по значению и по ссылке. Объявление конструкторов, перегрузка конструкторов. Понятие свойство в языке C#. Определение свойства. Модификаторы доступа для свойств. Свойства, доступные только для чтения и только для записи. Свойства и потеря производительности. Структуры, объявление структуры. Отличия структур от классов. Случаи предпочтительного использования структур. Специфика работы операции `new` при выделении памяти для структурных объектов. Конструкторы структур, особенность конструктора копирования. Класс `Object` и метод `ToString()`. Назначение механизма перегрузки операций. Логика перегрузки операции сложения для класса, реализующего математический вектор. Перегрузка арифметических операций. Особенность в C# перегрузки операций отношения (сравнения). Ограничения на перегрузку операций в C#.

Тема 14. Наследование C#

Сравнение типов наследования – наследование функциональности (реализации) и наследование интерфейса. Сущность наследования функциональности. Примеры с использованием наследования функциональности. Конструкторы производных классов. Переопределение методов базового класса виртуальные методы. Особенность синтаксиса переопределения метода в языке C#. Реализация виртуальными функциями динамического

полиморфизма. Наследование интерфейса - абстрактные классы и абстрактные методы. Интерфейсы C#, синтаксис объявления. Сходство и различие абстрактных классов и интерфейсов. Создание и использование интерфейсов.

Тема 15. Делегаты и события

Концепция указателей на функции, передача метода, как параметра другому методу. Назначение делегатов. Объявление и примеры использования делегатов. Групповые делегаты, назначение, особенности синтаксиса. Модель событий C#. Объявление события. Регистрация обработчиков события. Уведомление зарегистрированных обработчиков о наступлении события (генерация события).

Тема 16. Коллекции. Основы разработки интерфейса приложений

Концепция коллекции в C#. Массивы списки – класс ArrayList. Создание экземпляра класса. Сохранение объектов в ArrayList. Добавление и удаление элементов. Методы класса. Класс Stack (стек). Хранение временных элементов. Реализует алгоритм Last In First Out. Помещение и извлечение элементов из стека. Методы класса. Класс Queue (очередь). Принцип действия очереди – First In First Out. Методы класса. Пространство имен System.Windows.Forms. Класс формы (class Form), основные свойства, события и методы. Создание и уничтожение экземпляра формы, внешний вид формы. Стандартные элементы управления (Button, ComboBox, Label, TextBox, ListBox, MenuStrip, DataGridView), основные события и свойства. Многодокументный интерфейс (MDI).

4.4. Практическая подготовка

Практическая подготовка реализуется путем проведения практических занятий, предусматривающих участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

Объем занятий в форме практической подготовки составляет 16 часов.

5. Учебно-методическое, информационное и материально-техническое обеспечение дисциплины

5.1 Основная литература

1. Рихтер, Д. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Д. Рихтер; пер. с англ. — М.: Вильямс, 2020. — 896 с.
2. Страуструп, Б. Язык программирования C++ / Б. Страуструп; пер. с англ. — 4-е изд., перераб. и доп. — М.: Вильямс, 2022. — 1344 с.
3. Троелсен, Э., Джепсон, Ф. Язык программирования C#. Основы разработки приложений на платформе .NET / Э. Троелсен, Ф. Джепсон; пер. с англ. — 9-е изд. — СПб.: Питер, 2021. — 1120 с.

4. Шилдт, Г. Полный справочник по C++ / Г. Шилдт; пер. с англ. — 5-е изд., перераб. и доп. — М.: Вильямс, 2021. — 1088 с.

5.2 Дополнительная литература

1. Албахари, Д., Албахари, Б. C# 10 и платформа .NET 6. Карманный справочник / Д. Албахари, Б. Албахари; пер. с англ. — М.: Вильямс, 2022. — 400 с.
2. Глинский, Я. Н. Программирование на C++: от новичка до профессионала / Я. Н. Глинский. — СПб.: БХВ-Петербург, 2019. — 640 с.
3. Липпман, С., Лажуа, Ж., Му, Б. C++ для профессионалов / С. Липпман, Ж. Лажуа, Б. Му; пер. с англ. — М.: БХВ-Петербург, 2020. — 944 с.
4. Макдональд, М. Pro C# 10 с .NET 6 / М. Макдональд. — М.: АСТ, 2022. — 1256 с.
5. Хорстманн, К., Уэсли, Г. Современный C++ / К. Хорстманн, Г. Уэсли; пер. с англ. — 2-е изд. — СПб.: Питер, 2021. — 768 с.

5.3 Профессиональные базы данных и информационные справочные системы

1. <https://elibrary.ru> - Научная электронная библиотека eLIBRARY.RU (ресурсы открытого доступа)
2. <https://www.rsl.ru> - Российская Государственная Библиотека (ресурсы открытого доступа)
3. <https://link.springer.com> - Международная реферативная база данных научных изданий Springerlink (ресурсы открытого доступа)
4. <https://zbmath.org> - Международная реферативная база данных научных изданий zbMATH (ресурсы открытого доступа)

5.4 Материально-техническое и программное обеспечение (лицензионное и свободно распространяемое)

Наименование дисциплины (модуля), практик в соответствии с учебным планом	Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения.
Б1.О.12 Программирование	Кабинет информатики	Учебные места, оборудованные блочной мебелью, компьютерами с выходом в сеть интернет, рабочее место преподавателя в составе стол, стул, тумба, компьютер преподавателя с	Microsoft Windows XP Professional Microsoft Office 2010 Kaspersky Endpoint для бизнеса КонсультантПлюс AdobeReader Cisco WebEx Информационно-

		выходом в сеть интернет, экран, мультимедийный проектор, телевизор, тематические стенды, презентационный материал	коммуникационная платформа «Сферум»
	Аудитория для самостоятельной работы	Учебные места, оборудованные блочной мебелью, компьютерами с выходом в сеть интернет, многофункциональное устройство	

6. Методические указания для обучающихся по освоению дисциплины

6.1 Занятия лекционного и семинарского (практического) типов

Методические указания для занятий лекционного типа. В ходе лекционных занятий обучающемуся необходимо вести конспектирование учебного материала, обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Необходимо задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

Целесообразно дорабатывать свой конспект лекции, делая в нем соответствующие записи из основной и дополнительной литературы, рекомендованной преподавателем и предусмотренной учебной программой дисциплины.

Методические указания для занятий семинарского (практического) типа. Практические занятия позволяют развивать у обучающегося творческое теоретическое мышление, умение самостоятельно изучать литературу, анализировать практику; учат четко формулировать мысль, вести дискуссию, то есть имеют исключительно важное значение в развитии самостоятельного мышления.

Подготовка к практическому занятию включает два этапа. На первом этапе обучающийся планирует свою самостоятельную работу, которая включает: уяснение задания на самостоятельную работу; подбор основной и дополнительной литературы; составление плана работы, в котором определяются основные пункты предстоящей подготовки. Составление плана дисциплинирует и повышает организованность в работе. Второй этап включает непосредственную подготовку к занятию, которая начинается с

изучения основной и дополнительной литературы. Особое внимание при этом необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов. Далее следует подготовить тезисы для выступлений по всем учебным вопросам, выносимым на практическое занятие или по теме, вынесенной на дискуссию (круглый стол), продумать примеры с целью обеспечения тесной связи изучаемой темы с реальной жизнью. Готовясь к докладу или выступлению в рамках интерактивной формы (дискуссия, круглый стол), при необходимости следует обратиться за помощью к преподавателю.

6.2. Самостоятельная работа студентов

Самостоятельная работа студентов предусмотрена учебным планом по дисциплине в объеме 154 часов. Самостоятельная работа реализуется в рамках программы освоения дисциплины в следующих формах:

- работа с конспектом занятия (обработка текста);
- проработка тематики самостоятельной работы;
- написание контрольной работы;
- поиск информации в сети «Интернет» и литературе;
- выполнение индивидуальных заданий;
- подготовка к сдаче экзамена.

Самостоятельная работа проводится с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений обучающихся;
- углубления и расширения теоретических знаний студентов;
- формирования умений использовать нормативную, правовую, справочную документацию, учебную и специальную литературу;
- развития познавательных способностей и активности обучающихся: творческой инициативы, самостоятельности, ответственности, организованности; формирование самостоятельности мышления, способностей к саморазвитию, совершенствованию и самоорганизации;
- развитию исследовательских умений студентов.

Технология организации самостоятельной работы обучающихся включает использование информационных и материально-технических ресурсов: библиотека с читальным залом, компьютерные классы с возможностью работы в Интернет, аудитории для самостоятельной работы.

Перед выполнением обучающимися внеаудиторной самостоятельной работы преподаватель проводит консультирование по выполнению задания, который включает

цель задания, его содержания, сроки выполнения, ориентировочный объем работы, основные требования к результатам работы, критерии оценки.

Самостоятельная работа может осуществляться индивидуально или группами обучающихся в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности, уровня умений обучающихся.

Контроль самостоятельной работы студентов предусматривает:

- соотнесение содержания контроля с целями обучения;
- объективность контроля;
- валидность контроля (соответствие предъявляемых заданий тому, что предполагается проверить);
- дифференциацию контрольно-измерительных материалов.

Формы контроля самостоятельной работы:

- просмотр и проверка выполнения самостоятельной работы преподавателем;
- организация самопроверки, взаимопроверки выполненного задания в группе;
- обсуждение результатов выполненной работы на занятии;
- проведение письменного опроса;
- проведение устного опроса; организация и проведение индивидуального собеседования;
- организация и проведение собеседования с группой.

7. Методические рекомендации по обучению лиц с ограниченными возможностями здоровья и инвалидов

Обучение по дисциплине обучающихся с ограниченными возможностями здоровья (далее – ОВЗ) осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

Содержание образования и условия организации обучения, обучающихся с ОВЗ определяются адаптированной образовательной программой, а для инвалидов также в соответствии с индивидуальной программой реабилитации инвалида.

Освоение дисциплины обучающимися с ОВЗ может быть организовано как совместно с другими обучающимися, так и в отдельных группах. Предполагаются специальные условия для получения образования обучающимися с ОВЗ.

Профессорско-педагогический состав знакомится с психолого-физиологическими особенностями обучающихся инвалидов и лиц с ОВЗ, индивидуальными программами реабилитации инвалидов (при наличии).

В курсе предполагается использовать социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе. Подбор и разработка учебных материалов производятся с учетом предоставления материала в различных формах: аудиальной, визуальной, с использованием специальных технических средств и информационных систем.

Освоение дисциплины лицами с ОВЗ осуществляется с использованием средств обучения общего и специального назначения (персонального и коллективного использования). Материально-техническое обеспечение предусматривает приспособление аудиторий к нуждам лиц с ОВЗ.

Форма проведения аттестации для студентов-инвалидов устанавливается с учетом индивидуальных психофизических особенностей. Для студентов с ОВЗ предусматривается доступная форма предоставления заданий как оценочных средств, а именно:

- в печатной или электронной форме (для лиц с нарушениями опорно-двигательного аппарата);
- в печатной форме или электронной форме с увеличенным шрифтом и контрастностью (для лиц с нарушениями слуха, речи, зрения);
- методом чтения ассистентом задания вслух (для лиц с нарушениями зрения).

Студентам с инвалидностью увеличивается время на подготовку ответов на контрольные вопросы. Для таких студентов предусматривается доступная форма предоставления ответов на задания, а именно:

- письменно на бумаге или набором ответов на компьютере (для лиц с нарушениями слуха, речи);
- выбором ответа из возможных вариантов с использованием услуг ассистента (для лиц с нарушениями опорно-двигательного аппарата);
- устно (для лиц с нарушениями зрения, опорно-двигательного аппарата).
- при необходимости для обучающихся с инвалидностью процедура оценивания результатов обучения может проводиться в несколько этапов.

**Фонд оценочных средств
для текущего контроля и промежуточной аттестации
при изучении дисциплины
Б1.О.12 Программирование**

Москва 2024

1. Паспорт фонда оценочных средств

Код и наименование компетенции	Индикатор достижения компетенции	Наименование оценочного средства
ОПК-1 Способен проводить моделирование, анализ и совершенствование бизнес-процессов и информационно-технологической инфраструктуры предприятия в интересах достижения его стратегических целей с использованием современных методов и программного инструментария	ИОПК-1.1 Знать: основы моделирования, анализа и совершенствования бизнес-процессов и информационно-технологической инфраструктуры предприятия в интересах достижения его стратегических целей с использованием современных методов и программного инструментария ИОПК-1.2 Уметь: проводить моделирование, анализ и совершенствование бизнес-процессов и информационно-технологической инфраструктуры предприятия в интересах достижения его стратегических целей с использованием современных методов и программного инструментария ИОПК-1.3 Владеть: навыками моделирования, анализа и совершенствования бизнес-процессов и информационно-технологической инфраструктуры предприятия в интересах достижения его стратегических целей с использованием современных методов и программного инструментария	Текущий контроль: контрольная работа, тестовое задание Промежуточная аттестация: экзамен
ОПК-2 Способен проводить исследование и анализ рынка информационных систем и информационно-коммуникационных технологий, выбирать рациональные решения для управления бизнесом	ИОПК-2.1 Знать: методы исследования и анализа рынка информационных систем и информационно-коммуникационных технологий, выбирать рациональные решения для управления бизнесом. ИОПК-2.2 Уметь: проводить исследование и анализ рынка информационных систем и информационно-коммуникационных технологий, выбирать рациональные решения для управления бизнесом. ИОПК-2.3 Владеть: навыками исследования и анализа рынка информационных систем и информационно-коммуникационных технологий, выбирать рациональные	

	решения для управления бизнесом.	
ОПК-3 Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы программы для их практической реализации	ИОПК-3.1 Знать: основы управления процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации ИОПК-3.2 Уметь: управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации ИОПК-3.3 Владеть: навыками управления процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	

Этапы формирования компетенций в процессе освоения ОПОП прямо связаны с местом дисциплин в образовательной программе. Каждый этап формирования компетенций, характеризуется определенными знаниями, умениями и навыками и (или) опытом профессиональной деятельности, которые оцениваются в процессе текущего контроля успеваемости, промежуточной аттестации по дисциплине (практике) и в процессе итоговой аттестации. Дисциплина «Программирование» является промежуточным этапом формирования компетенций ОПК-1, ОПК-2, ОПК-3 в процессе освоения ОПОП.

Для оценки уровня сформированности компетенций в процессе изучения дисциплины предусмотрено проведение текущего контроля успеваемости по темам (разделам) дисциплины и промежуточной аттестации по дисциплине – экзамен.

2. Оценочные средства

2.1 Текущий контроль

Примерные задания для контрольных работ

1. Написать функцию, определяющую содержимое заданного бита целого числа.
Функции передаются, как параметры целое число и номер проверяемого бита.
2. Реализовать функцию для нахождения факториала числа: $f(n) = n!$
3. Написать функцию, реализующую заданную логическую операцию языка C++.
4. Написать функцию, выполняющую сортировку переданного ей массива.

5. Написать функции для работы со стеком: инициализация стека, запись в стек и чтение из стека. Стек имитируется как массив заданного размера с первоначальным значением указателя вершины стека, равным размеру массива.
6. Вывести на консоль линейку меню: 1, 2, 3, 4 с приглашением выбора. Через массив указателей на функции организовать реакцию на выбор пользователя (вывод на консоль соответствующей строки).
7. Создать класс, имитирующий работу программного стека. Конструктор класса выделяет в управляемой куче память под стек, размер которой определяется параметром конструктора. Класс должен содержать функции, имитирующие машинные команды – push, pop. Перегрузить для класса операцию присваивания =. Обосновать необходимость перегрузки этой операции для данного класса.
8. Осуществить перегрузку операции индексации [] для класса, имитирующего массив целых чисел с контролем границ массива. Конструктор класса выделяет в управляемой куче память, размер которой определяется параметром конструктора
9. Создать класс, имитирующий работу очереди (дисциплина «first in first out»).

Шкала и критерии оценивания контрольных работ

Шкала оценивания	Критерии оценивания
«отлично»	Обучающийся глубоко и содержательно раскрывает тему контрольной работы, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер.
«хорошо»	Обучающийся в целом раскрывает тему контрольной работы, однако ответ не носит развернутого и исчерпывающего характера.
«удовлетворительно»	Обучающийся в целом раскрывает тему контрольной работы и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«неудовлетворительно»	Обучающийся не владеет выбранной темой контрольной работы. Тема контрольной работы не раскрыта

Типовое тестовое задание

Вопрос №1 .

Что вернет функция Termin после выполнения. Код:

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

Варианты ответов:

1. 5
2. 3
3. 4

Вопрос №2 .

Какие бывают массивы?

Варианты ответов:

1. Разнообразные
2. Сложные и простые
3. Одномерные и многомерные
4. Резиновые и статичные

Вопрос №3 .

Какой оператор возвращает значение из метода?

Варианты ответов:

1. Veni
2. return
3. out
4. end

Вопрос №4 .

Выберите правильное описание переменных: переменная x для сохранения действительного типа, переменная a для сохранения целого числа, символьная переменная c; переменную с инициализировать значением 'n', переменную x инициализировать значением 18.21

Тип ответа: Многие из многих

Варианты ответов:

1. Float x, a; char c; c = 'n';
2. Float x = 18.21; a; char c = 'n';
3. Float x = 18.21; int a; char c = 'n';
4. Float x; x = 18.21; int a; char c = 'n';

Вопрос №5 .

Укажите, что может быть индексом элемента массива

Варианты ответов:

1. Символ
2. Дробное число
3. Целое положительное число
4. Математическое выражение, результат вычисления которого является целое положительное число

Шкала оценивания тестового задания

% верных решений (ответов)	Шкала оценивания
85-100%	«отлично»
70-84%	«хорошо»
51-69%	«удовлетворительно»
50% и менее	«неудовлетворительно»

2.2 Промежуточная аттестация

Примерный перечень вопросов к экзамену

Тема 1. Введение в язык C++

1. Особенность языков программирования: машинного, ассемблера, высокого уровня.
2. Базовые элементы языка C++.
3. Основные этапы разработки программы.
4. Базовые типы данных языка C++.
5. Консольный ввод/вывод.
6. Основные операции языка (арифметические, отношения, логические, побитовые), примеры.
7. Блок-схемы алгоритмических конструкций ветвления и цикла.
8. Область видимости и время жизни переменных, понятие блока.
9. Функции C/C++, определение, прототип, вызов

Тема 2. Массивы, указатели и ссылки

10. Понятие массива в языке C/C++, размещение элементов массива в памяти, объявление и инициализация массивов.
11. Доступ к элементам массива, понятие индекса, перебор элементов массива в цикле.
12. Многомерные массивы, структурная и бесструктурная инициализация массивов. Размещение многомерных массивов в памяти.
13. Указатели C/C++, назначение, примеры создания.
14. Операции над указателями.
15. Массивы указателей.
16. Связь между массивами и указателями.
17. Понятие ссылки в языке C/C++, допустимые способы использования.
18. Передача ссылки в качестве параметра в функцию.
19. Ссылка в качестве возвращаемого значения функции.

Тема 3. Операторы управления потоком

20. Понятие оператора, простые и составные операторы (блоки).
21. Оператор условия (if), логика работы, примеры.
22. Вложенные операторы if.
23. Множественный выбор с использованием конструкции if-else if -...-else.
24. Множественный выбор, оператор switch.
25. Оператор цикла while, примеры, варианты предпочтительного использования.
26. Оператор цикла for, примеры, варианты предпочтительного использования.
27. Оператор цикла do while, примеры, варианты предпочтительного использования.
28. Вложенные циклы, примеры.
29. Оператор break, назначение, примеры использования.
30. Оператор continue, назначение, пример использования.
31. Операторы return и goto.

Тема 4. Символьные массивы и строки

32. Строка – определение, инициализация, представление в памяти.
33. Считывание строк с клавиатуры.
34. Обработка строк, при использовании признака конца строки.
35. Библиотечные функции для работы со строками.
36. Массивы строк.

Тема 5. Типы, задаваемые пользователем

37. Создание структурных переменных.
38. Доступ к элементам структуры.
39. Передача структурных переменных функциям.
40. Объединения.
41. Перечисления.

Тема 6. Функции C/C++

42. Работа с функциями в языке C/C++: объявление, определение, вызов.
43. Перегрузка функций
44. Рекурсивный вызов функций
45. Механизм использования стека при вызове функций и возврате из них.
46. Передача параметров функции по значению и по адресу.
47. Указатели на функции, косвенный вызов функции.
48. Передача функции в качестве аргумента в другие функции.
49. Массивы указателей на функции.

Тема 7. Классы

50. Определение класса, данные-члены и функции члены, спецификаторы доступа.
51. Поведение (функциональность) класса.
52. Конструкторы и деструкторы (создание и удаление объектов).
53. Механизм инкапсуляции.
54. Перегрузка конструкторов.
55. Конструктор копирования.
56. Передача объектов функциям, объекты в качестве возвращаемых значений функций, примеры.
57. Массивы объектов.
58. Размещение объектов в динамической памяти.

Тема 8. Перегрузка операций

59. Назначение перегрузки операций, определение функции, реализующей перегрузку.
60. Перегрузка арифметических операций, пример.
61. Перегрузка операций отношения, пример.
62. Перегрузка логических операций, пример.
63. Перегрузка операции индексации, пример.
64. Перегрузка операции присваивания, пример.

Тема 9. Наследование функциональности

65. Концепция наследования в объектно-ориентированном подходе разработки программ.
66. Создание производного класса.
67. Возможности доступа из производного класса к членам базового класса, защищенные члены класса.
68. Конструкторы и наследование, передача аргументов конструктору базового класса.
69. Иерархия наследования.
70. Контейнерные классы.
71. Сравнение механизма наследования с контейнерным классом.

Тема 10. Виртуальные функции и полиморфизм

72. Указатели на базовый и производный класс и доступ к объектам.

73. Виртуальные функции и реализация принципа динамического полиморфизма, пример.
74. Понятие полиморфического кластера.
75. Чистые виртуальные функции и абстрактные классы.
76. Техническая реализация виртуальных функций.
77. Раннее и позднее связывание, преимущества и недостатки.

Тема 11. Шаблоны и библиотека стандартных шаблонов

78. Шаблоны в языке C++, назначение.
79. Родовая функция.
80. Родовой класс.
81. Понятие контейнер в языке C++.
82. Библиотека стандартных шаблонов: векторы.
83. Библиотека стандартных шаблонов: списки.
84. Библиотека стандартных шаблонов: ассоциативные списки.
85. Строковый класс в языке C++.

Тема 12. Основы языка C#

86. Типы данных в языке C#: типы значений и ссылочные типы.
87. Встроенные простые типы данных C# (типы значений)
88. Встроенные ссылочные типы C#: object и string.
89. Класс Console и ввод/вывод.
90. Условные операторы C#.
91. Операторы цикла C#.
92. Операторы перехода C#.
93. Массивы C# , создание и инициализация.
94. Пространства имен и оператор using.

Тема 13. Классы и структуры C#

95. Объявление класса, данные-члены и функции-члены, модификаторы доступа.
96. Передача параметров в метод.
97. Перегрузка конструкторов.
98. Свойства C#.
99. Структуры.
100. Механизм перегрузки операций.
101. Перегрузка в C# операций отношения.
102. Ограничения на перегрузку операций.

Тема 14. Наследование C#

103. Сравнение типов наследования (реализации и интерфейса).
104. Пример использования наследования реализации.
105. Конструкторы производных классов и передача параметров конструктору базового класса.
106. Виртуальные методы.
107. Абстрактные классы и абстрактные методы.
108. Интерфейсы C#.

Тема 15. Делегаты и события

109. Концепция указателей на функции.
110. Делегаты: назначение, объявление, пример использования.

111. Групповые делегаты.
112. Модель событий C#.

Тема 16. Коллекции. Основы разработки интерфейса приложений

113. Определение коллекции C#.
114. Массивы-списки (класс ArrayList), назначение, основные методы.
115. Стек (класс Stack), назначение, основные методы.
116. Очередь (класс Queue), назначение, основные методы.
117. Windows Forms. Класс формы.
118. Windows Forms. Стандартные элементы управления.
119. Windows Forms. Элемент управления DataGridView.
120. Windows Forms. Многодокументный интерфейс.

Шкала и критерии оценивания экзамена

Шкала оценивания	Критерии оценивания
«отлично»	оценка соответствует повышенному уровню и выставляется обучающемуся, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал монографической литературы, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
«хорошо»	оценка соответствует повышенному уровню и выставляется обучающемуся, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос или выполнении заданий, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.
«удовлетворительно»	оценка соответствует пороговому уровню и выставляется обучающемуся, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, демонстрирует недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.
«неудовлетворительно»	оценка выставляется обучающемуся, который не достигает порогового уровня, демонстрирует непонимание проблемы, не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы.