

Б1.В.ОД.9 «Объектно-ориентированное программирование»

Цель изучения дисциплины:

Целями преподавания дисциплины являются:

- освоение студентами языка C++ и на его основе овладение основными приёмами и методами программирования и алгоритмизации;
- приобретение навыков самостоятельного изучения отдельных тем дисциплины и решения типовых задач;
- приобретение навыков работы в современных интегрированных системах программирования;
- приобретение навыков разработки программных комплексов;
- усвоение полученных знаний студентами, а также формирование у них мотивации к самообразованию за счет активизации самостоятельной познавательной деятельности.

Место дисциплины в структуре ОП

Дисциплина «Объектно-ориентированное программирование» является обязательной дисциплиной вариативной части по направлению 09.03.01 «Информатика и вычислительная техника».

Содержание дисциплины:

1. Основы языка

1.1. Основы алгоритмизации. Модели решения функциональных и вычислительных задач. Основные этапы решения задач на ЭВМ; алгоритмы и их свойства. Структура программы. Алфавит языка. Имена. Целые типы.

1.2. Плавающие типы данных. Директивы препроцессора. Именованные константы.

1.3. Операции C++. Операции присваивания, сравнения, арифметические, логические и т.д.

1.4. Операторы C++. Условные операторы. Ветвление и циклы. Переключатели.

2. Составные типы

2.1. Указатели и массивы. Адресная арифметика.

2.2. Многомерные массивы и символьные строки.

2.3. Динамическое распределение памяти.

3. Функции

3.1. Функции. Передача аргументов по значению.

3.2. Объявления и определения. Область существования и видимости имен. Классы памяти.

3.3. Объявления объектов и типов. Правила преобразования стандартных типов.

3.4. Ссылки. Функции. Передача аргументов по ссылке. Аргументы по умолчанию. Передача указателя на функцию в качестве аргумента.

4. Методология объектно-ориентированного программирования

4.1. Классы. Понятие абстрактных типов данных. Скрытые, общие и защищенные данные. Операции ".", "->". Область видимости класс.

4.2. Функции-члены классов. Инлайн-функции. Инициализация данных. Конструкторы и деструкторы, их перегрузка. Порядок выполнения конструкторов и деструкторов.

4.3. Статические члены класса. Указатель this. Статические функции-члены. Указатели на члены класса.

4.4. Дружественные функции и дружественные классы. Конструктор и операция NEW.

4.5. Наследование. Построение производного класса. Защищенные члены.

4.6. Раннее и позднее (динамическое) связывание. Виртуальные функции. Чисто абстрактные классы.

4.7. Переопределение стандартных операций. Особенности переопределения операций new, delete, =, [], (), ->.

4.8. Преобразования абстрактных типов. Оператор-функция преобразования типов.

5. Динамические структуры данных.

5.1. Способы эффективного хранения и обработки данных. Основные структуры данных и их обработка с точки зрения объектно-ориентированного программирования. Списки. Двухнаправленные и кольцевые списки. Их построение и реализация.

5.2. Очереди и стеки. Их построение и реализация в виде массива и списка.

5.3. Деревья. Двоичное дерево поиска. Их построение и реализация.

5.4. Таблицы. Их применение в поисковых алгоритмах построения баз данных.

6. Общие вопросы проектирования программ

6.1. Оценка качества программных продуктов. Методы защиты программ и данных; проектирование интерфейса с пользователем.

6.2. Структуры диалога; поддержка пользователя; многооконные интерфейсы.

6.3. Объектное и событийное программирование, сравнительный анализ; Сравнение C++ с языками Java и C#.

Требования к результатам освоения дисциплины

Студент в соответствии с задачами профессиональной деятельности и целями основной образовательной программы после изучения дисциплины должен обладать следующими общекультурными, общепрофессиональными и профессиональными компетенциями:

способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования (ПК-2)

способностью участвовать в настройке и наладке программно-аппаратных комплексов (ОПК-4)

способностью обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности (ПК-3)

способностью осваивать методики использования программных средств для решения практических задач (ОПК-2)

В результате освоения дисциплины студенты должны:

Знать: инструментальные средства языков
программирования

Уметь: строить рациональные алгоритмы и на их основе создавать программные продукты, как для решения сравнительно простых задач, так и для задач с функциями, структурами данных, указателями, строками и классами;

Владеть: технологией структурного и объектно-ориентированного программирования, а также средствами отладки