

Аннотация адаптированной рабочей программы дисциплины «Комплексы проблемно-ориентированных программ»

Целью освоения дисциплины «Комплексы проблемно-ориентированных программ» является развитие у обучающихся умений и навыков в области выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях. В результате изучения дисциплины обучающийся должен освоить основные понятия, методы и технологии используемые при построения проблемно-ориентированных программных комплексов, уметь применять на практике методы и подходы информационных технологий.

Задачи дисциплины:

- обучение основным подходам к проектированию, разработке и использованию проблемно ориентированных программных комплексов;
- дать обучающимся знание технологий создания проблемно-ориентированных комплексов с использованием универсальных языков программирования;
- ознакомить обучающихся с использованием технологий объектно-ориентированного программирования при разработке проблемно-ориентированных комплексов;
- привитие обучающимся практических навыков создания программного обеспечения с использованием стандартных библиотек классов и шаблонов.

В результате освоения дисциплины обучающиеся изучат теоретический и практический материал по следующим темам:

Технические знания до V в. н. э. и в Средние века (V–XIV вв.)

1. Технические знания древности и античности до V в. н. э.

1.1 Технические знания как часть мифологии. Храмы и знания (Египет и Месопотамия).

Понятие и виды проблемно-ориентированных программных комплексов. Изучение типовых задач решаемых проблемно-ориентированными программными комплексами, различных подходов к их классификации. Рассмотрение примеров программных комплексов и основных областей их применения. Использование специализированных инструментальные средства и универсальных языков программирования для разработки проблемно-ориентированных программных комплексов. Обзор основных технологий создания проблемно-ориентированных программных комплексов. Рассмотрение специализированных инструментальные средства и универсальных языков программирования используемых для разработки проблемно-ориентированных программных комплексов.

Типы данных, операторы и управляющие конструкции. Основные встроенные типы данных языка C++. Операторы их назначение, свойства, приоритет и примеры использования. Особенности управляющих конструкций языка C++. Массивы, указатели и структуры. Одномерные и двумерные массивы. Указатели, адресная арифметика, операции * и &. Связь указателей и массивов, массивы указателей и указатели на массивы. Структуры, описание и обращение к полям, указатели на структуры. Использование функций стандартной библиотеки. Классификация функций стандартной библиотеки. Функции ввода-вывода, потоки, форматированный ввод-вывод. Работа с текстовыми и бинарными файлами. Функции работа с динамической памятью. Обработка строк и блоков памяти.

Инкапсуляция. Принцип и назначение и роль инкапсуляции при разработке программного обеспечения. Особенности реализации инкапсуляции в C++, права доступа к членам, друзья классов. Примеры практического использования инкапсуляции в практике программирования. Наследование. Назначение и область применения наследования при разработке программного обеспечения. Особенности реализации механизмов наследования в C++. Управление правами доступа при наследовании. Множественное и виртуальное наследование. Примеры использования различных аспектов механизма наследования в практике программирования. Полиморфизм и виртуальные функции. Понятие, назначения и область применения полиморфизма. Особенности реализации полиморфизма в C++. Виртуальные функции, определение, назначение особенности реализации. Таблица виртуальных методов. Чисто виртуальные методы и абстрактные классы. Примеры использования полиморфизма в практике программирования.

Понятие шаблонов и область их применения. Понятие, назначения и область применения шаблонов. Синтаксис описания шаблонов, особенности реализации шаблонов в C++. Шаблоны типов и шаблоны классов. Аргументы шаблонов, использование различных видов аргументов. Механизмы реализации шаблонов компилятором. Особенности организации исходного программного кода и использовании шаблонов. Разработка собственных шаблонов функций и типов. Назначение и особенности использования шаблонов функций. Примеры использования шаблонов функций. Назначение и особенности использования шаблонов типов. Примеры использования шаблонов типов. Применение шаблонов при создании различного рода контейнеров, включая списки, множества, стек и др.

Назначение и область применения библиотеки. Основные особенности и характерные черты библиотеки STL. Обзор функциональных возможностей и основных областей применения библиотеки. Рассмотрение групп шаблонов библиотеки с краткой характеристикой назначения и возможностей каждой из групп. Шаблоны потоков в STL. Характеристика возможностей и назначения потоков ввода-вывода библиотеки STL. Шаблоны используемые для консольного ввода-вывода. Шаблоны используемые для организации работы с файлами. Форматированный ввод-вывод. Буферизация при работе с потоками. Применение потоков библиотеки для работы с объектами пользовательских

классов. Шаблоны контейнеров в STL. Назначение и область применения контейнеров библиотеки STL. Принципы построения шаблонов контейнеров. Особенности применения итераторов и распределения памяти.

Методы и примеры использования шаблонов потоков. Методы использования потоков при создании проблемно-ориентированных программных комплексов. Основные преимущества использования потоков. Практические приемы и рекомендации использования потоков. Примеры практического использования шаблонов потоков при разработке прикладного программного обеспечения. Методы и примеры использования шаблонов контейнеров. Методы использования шаблонов контейнеров при создании проблемно-ориентированных программных комплексов. Основные преимущества использования шаблонов контейнеров. Практические приемы и рекомендации использования контейнеров. Примеры практического использования шаблонов контейнеров при разработке прикладного программного обеспечения.

Объем дисциплины 3 з.е.

Форма промежуточного контроля – *зачет с оценкой*.