1. Цели и задачи дисциплины

Целью курса является ознакомление с технологиями анализа и синтеза сложных программных систем при объектно-ориентированном подходе к анализу и проектированию приложений.

Задачи дисциплины: сформировать у студента представление о современных технологиях анализа и синтеза сложных программных систем, овладение методами объектно-ориентированного программирования и проектирования в реальной программной среде.

2. Место дисциплины в учебном процессе и требования к знаниям и умениям специалиста

Изучение дисциплины основано на изучении дисциплин “Алгоритмические языки”, ”Технологии программирования ”, ”Системное программное обеспечение”. Дисциплина является основой для изучения дисциплин “Диалоговые средства”, дипломное проектирование.

В результате изучения дисциплины студенты должны:

Знать концепцию объектно-ориентированного программирования, основные ее понятия (класс, объект), свойства (инкапсуляция, наследование, полиморфизм), методику анализа и проектирования объектно-ориентированных программ;

Уметь проектировать, программировать и отлаживать объектно-ориентированные программы;

Иметь представление об объектно-ориентированных расширениях современных языков программирования, инструментальных программных средствах ООП; тенденциях развития и областях применения ООП.

Иметь навыки проектирования классов и иерархий классов, использования стандартных систем классов в прикладных задачах

3. Содержание дисциплины.

Тема 1. Эволюция технологий проектирования и программирования. История развития объектно-ориентированных языков программирования. ADA, SmallTalk, C++, ObjectPascal и т.д. Реализация основных принципов ООП в различных средах программирования.

Тема 2. Сложность, присущая программному обеспечению. Структура сложных программ. Признаки сложной системы. Организованная и неорганизованная сложность. Декомпозиция. Абстракция. Иерархия.

Тема 3. Объектный подход. Тенденции в методологии проектирования программных средств. Основные положения объектного подхода. Компоненты объектного подхода. Применение объектного подхода.

Тема 4. Классы и объекты: Данные, методы, наследование свойств, доступ. Состояние, поведение и индивидуальность объекта. Отношения между объектами. Внешние и внутренние проявления класса. Отношения между классами. Отношения между классами и объектами.

Тема 5. Диаграмма классов. Диаграммы перехода состояний. Диаграмма объектов. Диаграммы процессов. Исключительные ситуации.

Тема 6. Объектно-ориентированное проектирование. Идентификация классов и объектов. Идентификация семантики классов и объектов. Идентификация связей между классами и объектами. Реализация классов и объектов. Достоинства и недостатки объектно-ориентированного проектирования.

Тема 7. Технология визуального объектно-ориентированного программирования и RAD технология. Предпосылки развития, основные положения, современное

состояние. Классификация языков визуального программирования. Принципы групповой разработки программного обеспечения.

Тема 8. Borland Delphi. MS Visual Studio(C++, C#). Системы классов Visual Component Library(VCL), Microsoft Foundation Classes(MFC).

Тема 9. Интерфейс: правила организации, методы и средства программирования. Абстрактные классы. Реализация наследования через механизм интерфейсов.

Тема 10. Динамические объекты. Разработка и использование. Динамическое связывание. Работа с динамическими объектами.

Тема 11. Проектирование, разработка и отладка компонентов. Визуальные и невизуальные компоненты. Встраивание собственных компонентов в стандартную иерархию классов.

Тема 12. Шаблоны, как средство автоматизации программирования. Шаблоны классов и функций. Наследование шаблонных классов. Правила отождествления параметров шаблона. Применение шаблонных классов для создания контейнерных классов. Стандартная библиотека шаблонов(STL).

Тема 13. Объектно-ориентированные системы: методы, языки и способы программирования. Объектно-ориентированные базы данных. Принципы проектирования ООБД. Примеры ООСУБД. Среды разработки.

6. Задания для выполнения контрольной работы

В среде MS VISUAL STUDIO C++ реализовать в соответствии с вариантом иерархию наследования. В реализуемые классы в обязательном порядке должны быть включены виртуальные функции, конструкторы, деструкторы.

№ Вариант задания.

3 рабочий, кадры, инженер, администрация;