1

1) Win32 API. #include <window.h>. Создание потоков, критической секции, событий. (CreateThread(), EnterCriticalSection(), CreateEvent())
2) Это основная единица, которой операционная система выделяет время процессора. Каждый поток имеет приоритет планирования и набор структур, в которых система сохраняет контекст потока, когда выполнение потока приостановлено. Потоки используются для многопоточного программирования, что позволяет программам увеличить пропускную способность.
3) Объект синхронизации, который используется для координации доступа к общим ресурсам несколькими потоками. Название "мьютекс" происходит от сокращения от английского термина "mutual exclusion" (взаимное исключение). Мьютексы предотвращают конфликты доступа к разделяемым ресурсам, гарантируя, что только один поток имеет доступ к защищенному участку кода или данным в определенный момент времени.
4) объект синхронизации, который используется для уведомления одного или нескольких потоков о том, что произошло какое-то событие. События могут быть использованы для сигнализации о завершении операции, изменении состояния, ожидании ввода и других сценариях. События бывают двух типов: ручные (manual-reset) и автоматически сбрасываемые (auto-reset). Ручные события остаются в состоянии "сигнал" (выставленным) после того, как один из потоков их установит, в то время как автоматические сами сбрасываются в состояние "несигнал" после того, как один поток был уведомлен.
5) Сравнение будет происходить между 98 и 11. При сравнении можно выделить несколько факторов: в 11 стандарте нет как таковой возможности создавать приостановленные потоки, так как CREATE_SUSPENDED не работает в силу изменения работы функций создания потоков, изменение работы функций по созданию потоков, изменение критических секций, вернее их отсутствие, наличие condition_variable.

3

1) Это процесс разделения сложной системы на более мелкие и управляемые компоненты, которые называются объектами. Этот принцип ООП помогает в управлении сложностью программного обеспечения и обеспечивает легкость поддержки и модификации кода. Декомпозиция включает в себя создание объектов, которые абстрагируют различные аспекты системы и связывают их взаимодействие через интерфейсы и отношения.

2) Форма полиморфизма, при которой вызов конкретной версии функции или оператора определяется во время компиляции, а не во время выполнения программы. Это достигается за счет перегрузки функций или операторов с разными параметрами или типами данных.
3) Это один из основных принципов объектно-ориентированного программирования, который заключается в объединении данных и методов, которые работают с этими данными, в единый объект или класс. Основная идея инкапсуляции заключается в том, чтобы сокрыть детали реализации и предоставить интерфейс, через который можно взаимодействовать с объектом. Данные внутри объекта скрыты от прямого доступа извне, и доступ к ним осуществляется только через методы класса.

4

1) Builder - это порождающий паттерн проектирования, который позволяет создавать сложные объекты пошагово. Центральной идеей этого паттерна является разделение процесса конструирования объекта от его представления. Каждый шаг конструирования выполняется отдельным объектом, называемым builder-ом, который инкапсулирует детали создания объекта.
2) Decorator - это структурный паттерн проектирования, который позволяет динамически добавлять объектам новые функциональные возможности, оборачивая их в специальные классы-декораторы. Паттерн использует композицию объектов для расширения функциональности и обеспечивает гибкую альтернативу наследованию.