Структурное программирование

При структурном подходе составление алгоритмов и программ основано на трех базовых структурах: *линейные*, *разветвляющиеся* и *циклические* алгоритмы.

Основная характеристика структурного программирования – это ***модульность***.

Программа разбивается на множество ***подпрограмм-модулей***, каждая из которых выполняет одно или несколько действий.

Комбинируя эти подпрограммы, формируется итоговый алгоритм, состоящий уже не из простых операторов, а из законченных блоков, имеющих определенную смысловую нагрузку.

Нисходящее проектирование

Основным принципом технологии структурного программирования является ***нисходящее проектирование***, которое позволяет вести разработку приложения «***сверху вниз***», от общих задач к частным, образуя, тем самым иерархическую структуру программы.

При программировании сначала выделяются подпрограммы, решающие глобальные задачи. На следующем шаге происходит уточнение, при котором каждый из модулей разбивается на небольшое число других подпрограмм. Процесс разбиения модулей происходит до тех пор, пока вся задача не окажется реализованной.

Достоинство такого подхода заключается в том, что программа становится более надежной, ее легче отлаживать, а подпрограммы можно использовать

Объектно-ориентированное программирование

Объектно-ориентированное программирование – это метод программирования, который в качестве основных элементов конструкции использует *классы* и *объекты*, а не алгоритмы.

Объекты представляют собой программные модули. Каждый программный объект обладает ***свойствами***, описывающими структуру его данных, использует ***методы*** (средства обработки данных) и может реагировать на ***события***, которые приводят, как правило, к изменению свойств объекта.

Методы включают в себя набор процедур и функций, определяющих алгоритм работы объекта.

Однотипные объекты объединяются в классы.

Классы объектов

Класс объектов – это *шаблон*, определяющий набор свойств, методов и событий, специфический для данных объектов . Класс можно понимать как описание множества объектов, имеющих общую структуру и поведение.

Экземпляр класса – это объект, созданный по шаблону класса объектов. Каждый экземпляр имеет структуру данного класса и уникальное имя.

Экземпляры классов взаимодействуют между собой, посылая и получая сообщения.

Сообщение – это запрос на выполнение действия, содержащий набор необходимых параметров.

Механизм сообщений реализуется с помощью вызова функций или методов класса.

Принципы объектно-ориентированного программирования

Инкапсуляция – объединение в одном объекте данных и свойственных им процедур обработки. Согласно этому принципу, пользователь для данного класса должен видеть и использовать только список свойств и методов объекта и не вникать в его внутреннюю реализацию.

Наследование предусматривает создание новых классов на базе существующих и позволяет классу-потомку унаследовать все или некоторые свойства и методы класса-родителя (базового класса).

Полиморфизм представляет собой свойство различных объектов выполнять одно и то же действие разными способами.

Модульность – означает, что объекты заключают в себе полное определение их характеристик. Никакие определения методов и свойств не должны располагаться вне объекта. Это делает возможным свободное копирование и внедрение одного объекта в другие.

Интегрированные системы программирования

Интегрированная система программирования – это комплекс программ, предназначенных для разработки и эксплуатации программ на конкретном алгоритмическом языке программирования высокого уровня.

Для создания программы система программирования содержит следующие компоненты:

- ***текстовый редактор*** для ввода и редактирования исходного кода программы;
- ***транслятор*** – программа, обеспечивающая перевод исходного кода на машинный язык;
- ***редактор связей*** (компоновщик) – программа, которая объединяет программные модули в одну программу, готовую к исполнению. При этом создается исполняемый файл с расширением ***.exe*** (загрузочный модуль).
- ***библиотека*** стандартных функций и процедур;
- ***интегрированный отладчик***, позволяющий анализировать работу программы в режиме пошагового выполнения, отслеживая при этом значения переменных, с

Трансляция программы

Для перевода исходного текста (исходного кода) программы на машинный язык (машинный код) используются два вида трансляции: *интерпретация* и *компиляция*.

Интерпретатор – это программа, которая выбирает поочередно операторы исходного кода, анализирует их и сразу выполняет. Недостатком интерпретаторов является то , что программы с большим числом повторяющихся вычислений работают медленно. Кроме того, для выполнения такой программы на другом компьютере требуется снова установить интерпретатор.

Компилятор полностью обрабатывает весь текст программы и генерирует машинный код, который называется *объектным кодом*. В процессе компиляции осуществляется поиск синтаксических ошибок, а также выполняется оптимизация исходного кода, позволяющая повысить быстродействие программы. В результате полученная программа является компактной, эффективной и может быть

Среды быстрого проектирования

При объектно-ориентированном программировании широко используется ***визуальный подход***.

Для автоматизации визуального программирования разработаны специальные ***среды быстрого проектирования*** (*RAD*-среды).

Все необходимые элементы управления создаются не путем ручного программирования, а с помощью готовых *визуальных компонентов,* которые перетаскиваются в проектируемое *окно-форму*. Затем их свойства настраиваются с помощью простых *редакторов*. При этом вспомогательный исходный текст программы, ответственный за создание и работу элементов, генерируется *RAD*-средой автоматически.

В результате программирование во многом заменяется на *проектирование,* что позволяет сосредоточиться на логике решения задачи.

Визуальные среды разработаны для многих объектных языков программирования. Наиболее популярны в настоящее

Языки программирования

Машинный язык – запись программы в двоичном коде. ЭВМ понимает только машинный язык.

Машинно-ориентированный язык – язык символического кодирования. Является *языком низкого уровня*. Программирование на этом языке заключается в замене машинных кодов на их буквенные обозначения. Пример такого языка – **Ассемблер**. Требуется транслятор, который называется «ассемблер». Зависит от типа конкретной ЭВМ.

Процедурно-ориентированные языки относятся к языкам программирования *высокого уровня*. Не зависят от типа ЭВМ. Запись программы достаточна близка к естественной записи, удобна для восприятия. Требуется транслятор. Примеры языков высокого уровня:

- *Фортран, Бейсик, Алгол, Паскаль* – для инженерно-технических расчетов;
- *Пролог* – язык логического программирования;
- *Лисп* – для решения задач функционального программирования, ориентированный на обработку списков.

В настоящее время широко используются и другие группы языков: ***объектно-ориентированные языки*** (*C++, Delphi*, *Visual Basic*, *Java*, *Javascript*), ***языки программирования баз данных*** (*SQL*); ***языки программирования для компьютерных сетей*** (*HTML*, *VRML*, *XML*,