ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Елсакова
Алёна Владимировна

Эволюция языков программирования

- Первым программистом была ***женщина – леди Ада Лавлейс***, дочь поэта лорда Байрона.

Она разрабатывала программы для одного из первых механических компьютеров, созданного в начале XIX века английским учёным Чарльзом Беббиджом.

Однако настоящее программирование в современном понимании началось с момента создания первой ЭВМ. Тем не менее, имя этой замечательной женщины – ***Ada*** – присвоено одному из самых ***мощных*** современных ***языков программирования***, который является базовым для министерства обороны США.

Языки низкого уровня

- ***Машинный язык*** – единственный язык, понятный ЭВМ. Он реализуется аппаратно: каждую команду выполняет некоторое электронное устройство. Программа на машинном языке представляет собой последовательность команд и данных, заданных в цифровом виде.

- Стремление программистов оперировать не цифрами, а символами, привело к созданию мнемонического языка программирования, который называют ***языком ассемблера***, мнемокодом, ***автокодом***.

Языки высокого уровня

Эти языки являются универсальными (с их помощью можно создавать любые прикладные программы) и алгоритмически полными, имеют более широкий спектр типов данных и операций, поддерживают технологии программирования.

Принципиальными отличиями языков высокого уровня от языков низкого уровня являются:

- использование переменных;
- возможность записи сложных выражений;
- расширяемость типов данных за счет конструирования новых типов из базовых;
- расширяемость набора операций за счёт подключения библиотек подпрограмм;
- слабая зависимость от типа ЭВМ.

Новое поколение языков

- ***Универсальные*** языки программирования – позволяет получить быстро эффективную, надежную и безошибочную программу.

- ***Проблемно-ориентированные*** языки программирования – решают экономические задачи (COBOL), задачи реального времени (Modula-2, Ada), символьной обработки (Snobol), моделирования (GPSS, Simula, SmallTalk), численно-аналитические задачи (Analitic) и другие.

- Языки ***сверхвысокого*** уровня - программист задаёт отношения между объектами в программе, например систему линейных уравнений, и определяет, что нужно найти, но не задаёт как получить результат.

Классификация языков программирования

- По степени ориентации на специфические возможности ЭВМ.
- По степени детализации алгоритма получения результата.
- По степени ориентации на решение определенного класса задач.
- По возможности дополнения новыми типами данных и операциями.
- По возможности управления реальными объектами и процессами.
- По способу получения результата.
- По типу решаемых задач

Структуры и типы данных языка программирования

Структуры данных служат теми материалами, из которых строятся программы. Структура данных относится, по существу, к «пространственным» понятиям: её можно свести к схеме организации информации в памяти компьютера. Под ***структурой данных*** в общем случае понимают множество элементов данных и связей между ними.

Типы данных, принятые в языках программирования, включают натуральные и целые числа, вещественные числа (в виде приближенных десятичных дробей), литеры, строки и т.п.

Классификация структур данных

Основные процессы

Транслятор (англ. translator – переводчик) – это программа-переводчик, преобразующая программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд.

Основные группы трансляторов:

1) *Ассемблеры* - системная обслуживающая программа, которая преобразует символические конструкции в команды машинного языка.

2) *Компиляторы* - (англ. compiler – составитель, собиратель) читает всю программу целиком, делает её перевод и создаёт законченный вариант программы на машинном языке, который затем и выполняется.

3) *Интерпретаторы* - (англ. interpreter – истолкователь, устный переводчик) переводит и выполняет программу строка за строкой.

Взаимодействие программ с устройствами ЭВМ

Эмулятор – программа или программно-техническое средство, обеспечивающее возможность без перепрограммирования выполнять на данной ЭВМ программу, использующую коды или способы выполнения операций, отличные от данной ЭВМ.

Перекодировщик – программа или программное устройство, переводящие программы, написанные на машинном языке одной ЭВМ в программы на машинном языке другой ЭВМ.

Макропроцессор – программа, обеспечивающая замену одной последовательности символов другой.

Среда исполнения

Синтаксис – совокупность правил некоторого языка, определяющих формирование его элементов.

Семантика – правила и условия, определяющие соотношения между элементами языка и их смысловыми значениями, а также интерпретацию содержательного значения синтаксических конструкций языка.

Синтаксический анализатор – компонент компилятора, осуществляющий проверку исходных операторов на соответствие синтаксическим правилам и семантике данного языка программирования.

Классификация программных ошибок

По видам вредоносного воздействия:

- ошибки, приводящие к порче пользовательских данных в процессе обработки;
- ошибки, приводящие к неавторизованному доступу к пользовательским данным;
- ошибки, приводящие к исчерпанию системных ресурсов;
- ошибки, приводящие к аварийному завершению исполнения программы;
- ошибки, приводящие к исполнению злонамеренного кода.

Причины появления программных ошибок

1) Влияние квалификации команды разработчиков программы:

- плохой или недостаточный уровень проработки архитектуры программы;
- пренебрежительное отношение программистов к результату своей работы.

2) Уделяется мало внимания различным уровням тестирования программы в процессе разработки:

- недостаточное количество и качество автоматических тестов на компоненты (модульных тестов);
- недостаточность или отсутствие интеграционных тестов;
- недостаточный уровень тестирования программы на устойчивость при обработке некорректных внешних данных.

3) Влияние изменений, вносимых в среду исполнения программы