Мертвые языки программирования

Канарейкин А. И.

ЭВОЛЮЦИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ. КЛАССИФИКАЦИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Поколения ЭВМ	Характеристика ЭВМ	Языки программирования	Характеристика ЯП
1 поколение (1946-1954г. г.)	Для научно-технических расчетов; носитель информации - перфокарты и перфоленты; программное обеспечение-библиотека стандартных программ, автокоды.	Машинные	Ориентированы на использование в конкретной ЭВМ, сложны в освоении, требуют хорошего знания архитектуры ЭВМ.
2 поколение (1958-1960г. г.)	Для технологических и экономических расчётов; носитель информации-магнитная лента; созданы трансляторы.	Ассемблеры, макроассемблеры	Более удобны для использования, но по-прежнему машинно-зависимы
3 поколение (1960-1966г. г.)	Для управления и технологических расчётов; носитель информации- диск; программное обеспечение-операционные системы; СУБД; пакеты прикладных программ.	Языки высокого уровня	Мобильные, человеко-ориентированные, проще в освоении
4 поколение (1970-1980г. г.)	Для информационного обслуживания, телекоммуникации; носитель информации-гибкий диск; прикладное ПО общего назначения; сетевое ПО; мультимедия.	Непроцедурные, объектно-ориентированные, языки запросов, параллельные	Ориентированы на непрофессионального пользователя и на ЭВМ с параллельной архитектурой
5 поколение (1980г.- по наше время)	Телекоммуникации, информационное обслуживание; носитель информации- гибкий, жёсткий и оптический диски; интеллектуальные программные системы;	Языки искусственного интеллекта, экспертных систем и баз знаний, естественные языки	Ориентированы на повышение интеллектуального уровня ЭВМ и интерфейса с языками

Причины исчезновения языков программирования

Мёртвый язык программирования – это язык, который по определенным причинам вышел из употребления. Точная смерть языка наступает в том случае, когда язык претерпевает эволюцию и развивается в другой язык или даже в группу языков.

Время жизни языка программирования зависит от успешности команды создателей и промоутеров, т.е. **Бейсик и Фортран** существовали на **ЕС-ЭВМ** на **СМ-ЭВМ**, и на ПК в период существования **DOS** первых версий, тогда компьютеры назывались ЭВМ и предназначались для счета, но эти **ЭВМ** уже изжили себя. Вычислительная мощь, это последнее о чем мы сейчас вспоминаем по отношению к компьютерам, мы мало нуждаемся в счете, очень мало людей, чьи вычислительные потребности выходят за рамки обычного калькулятора, но из этих немногих попадая в электронные таблицы типа **Excel**, там и остаются в нирване. Эти языки справлялись со своими задачами и прекрасно жили в условиях тех "сумасшедших ресурсов", но изменился подход к использованию вычислительной техники, и целая категория вычислительной техники превратилась в "абак" – вычислительное средство, представляющее интерес, только исторически.

Что делает язык мощным и пригодным для всех ипостасей?

Прикладное или системное программирование - это наличие различных библиотек, с оптимальным кодом и удачным транслятором (компилятор или интерпретатор). Мы живем в эпоху вымирания языков. Вымирают естественные языки, такой же тенденции подвержены и языки программирования. Причины вымирания языков программирования разнообразны. Раньше огромная польза была от **Pascal**. Сейчас он почти умер. В свое время **Perl** считался крутым, но затем его вытеснил **PHP**. Однако все языки создавались для чего-то, а не просто так. И перед тем как окончательно отправить тот или иной язык в архив истории, следовало бы полностью осознать его наследие, употребляемое в текущей практике программирования. А может статься, что некоторые языки вообще списаны со счетов зазря. Вопрос лишь в том, как грамотно провести ревизию, инвентаризацию всего этого добра из сотен и тысяч языков, как извлечь крупицы смысла из гор отработанной породы?

Причины краха языков программирования:

- Сложны в применении
- Не доступны для простого пользователя
- Ошибки менеджмента
- Отсутствие «рекламы» в СМИ, вернее присутствие антирекламы
- Отсутствие поддержки крупных вендоров, в отличии от конкурентов
- Были вытеснены альтернативным языком, новыми технологиями и изменяющимися условиями выживания
- Низкий порог вхождения
- Отсутствие сертификационных центров

МЕРТВЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ

ЯЗЫК **COBOL** — ЯВЛЯЕТСЯ ОДНИМ ИЗ СТАРЕЙШИХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ. ОН ИСПОЛЬЗОВАЛСЯ В СФЕРЕ БИЗНЕСА И ФИНАНСОВ. НАЗВАНИЕ ТАК И РАСШИФРОВЫВАЕТСЯ — ОРИЕНТИРОВАННЫЙ НА БИЗНЕС ЯЗЫК (АНГЛ. COMMON BUSINESS ORIENTED LANGUAGE).

ОН ДЕБЮТИРОВАЛ В 1959 ГОДУ И ШИРОКО ИСПОЛЬЗОВАЛСЯ НА МЕЙНФРЕЙМАХ КОМПАНИИ IBM ДЛЯ БИЗНЕС-ЗАДАЧ, ТАКИХ, КАК БУХГАЛТЕРСКИЙ УЧЕТ. ЭТОТ ЯЗЫК ИМЕЛ СВОИ ПЛЮСЫ И МИНУСЫ. К ПЕРВОМУ МОЖНО ОТНЕСТИ УДОБНЫЙ ИНСТРУМЕНТАРИЙ ДЛЯ РАБОТЫ С БОЛЬШИМИ СТРУКТУРАМИ ДАННЫХ И ФАЙЛАМИ. ОДНАКО СИНТАКСИС ЯЗЫКА СТРАДАЛ ИЗ-ЗА ТОГО, ЧТО КОНСТРУКЦИЯ КОДА БЫЛА МАКСИМАЛЬНО ПРИБЛИЖЕНА К АНГЛИЙСКОМУ ЯЗЫКУ.

ПОСЛЕ 2000 ГОДА ЕГО ПОПУЛЯРНОСТЬ СНИЗИЛАСЬ. СЕЙЧАС ЭТОТ ЯЗЫК СЧИТАЕТСЯ МЕРТВЫМ, ТАК КАК НА НЕМ ПЕРЕСТАЛИ ПИСАТЬ. УЖЕ НЕСКОЛЬКО ДЕСЯТКОВ ЛЕТ ОН ИСКЛЮЧЕН ИЗ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ. НЕСМОТРЯ НА ЭТО, IBM ПРОДОЛЖАЕТ ПРОДАВАТЬ МЕЙНФРЕЙМЫ, СОВМЕСТИМЫЕ С COBOL.

ПРОБЛЕМЫ С УСТАРЕВШИМИ ЯЗЫКАМИ

Управление по подотчетности правительства США неоднократно предупреждало об использовании устаревших языков программирования для критически важных систем. Например, система Департамента образования для обработки федеральных заявок на помощь студентам была внедрена в 1973 году. Для обслуживания системы требуется 18 подрядчиков, а поскольку она написана на языке COBOL, для нее требуется специальное оборудование и его трудно интегрировать с новыми языками программного обеспечения.

Найти сотрудников, которые вообще знают, как писать подобный код, невероятно сложно. Это также означает, что, когда система ломается, нет людей, которые могли бы ее исправить в кратчайшие сроки. В такой ситуации сейчас оказались власти этих штатов, с провисшей системой и нехваткой квалифицированных инженеров.

Подобная ситуация уже возникала на мировом уровне, когда появилась «Проблема 2000 года». Тогда было неясно, может ли дата нового тысячелетия вызвать лавинные ошибки во всех мировых вычислительных системах. В ответ на это легионы программистов, свободно владеющих в основном забытыми языками, такими как COBOL, были специально наняты для исправления правительственного и корпоративного кода. Более 20 лет спустя многие государственные, федеральные и банковские системы все еще работают на этих самых языках программирования.

Несмотря на свой возраст и тот факт, что очень много программистов перешли на C и Java, COBOL по-прежнему является широко используемым языком программирования. По данным Reuters, сегодня почти половина банковских систем мира работает на языке COBOL, и более 80% транзакций на основе карт используют этот код.