



История программного обеспечения

ПОДГОТОВИЛ УЧЕНИК
9 КЛАССА В
МБОУ ЦО ШКОЛЫ №44
СОРОКИН ЕГОР

Программное обеспечение

- ▶ Программное обеспечение. История развития. Классификация программного обеспечения.
- ▶ 2 Программа - последовательность указаний (команд) для ввода исходных данных, их обработки и выдачи результатов. Программное обеспечение (ПО, Software) - совокупность программ, используемых на ПК.
- ▶ 3 Системное ПО предназначено для эксплуатации и технического обслуживания ПК, управления и организации вычислительного процесса при решении любой конкретной задачи на ПК и т.д. Системное ПО обязательная часть ПО, к нему относятся: · Операционные системы (Windows, Linux) · Оболочки операционных систем (Windows Commander, Norton Commander (NC), DOS Shell, Volkov Commander) · Программы-утилиты (драйверы, архиваторы, антивирусы и т.д.)

Прикладное ПО

- ▶ Прикладное ПО (или пакеты прикладных программ) предназначено для решения определенного класса задач, т.е. это программы, используемые как инструмент при создании документов в повседневной деятельности. ИЛИ ПРОГРАММЫ, с помощью которых пользователь решает свои информационные задачи, не прибегая к программированию. К ним относятся: · текстовые и графические редакторы (Corel Draw, Page Maker, Lexicon) · электронные таблицы (Lotus 1-2-3, SuperCalc) · системы управления базами данных (FoxPro, dBASE, Paradox, Clipper, Clarion, Access) · интегрированные среды (FrameWork, Works) · обучающие и учебные программы · игры · мультимедиа и т.д.

Инструментальное ПО

- ▶ Инструментальное ПО (или системы программирования) обеспечивают создание ВСЕХ КЛАССОВ ПРОГРАММ: системных, прикладных и новых систем программирования или обеспечивает создание новых прикладных программ для компьютера. Языки программирования высокого уровня, такие как Qbasic, Pascal, C, Prolog, Ada, JavaScript, VBScript. Разработка любого программного обеспечения осуществляется с помощью систем программирования.



- ▶ Инструментальные средства Транслятор (англ. translator переводчик) это программа-переводчик. Она преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд. Компилятор (англ. compiler составитель, собиратель) читает всю программу целиком, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется. Интерпретатор (англ. interpreter истолкователь, устный переводчик) переводит и выполняет программу строка за строкой.
- ▶ Эволюция языков программирования машинно-зависимый ассемблерного типа высокого уровня непроцедурные скриптовые
- ▶ 1 этап Языки программирования первого поколения были строго машинно-зависимыми, представляя собой по сути набор машинных команд, который определялся архитектурой конкретной ЭВМ.

Этапы развития

- ▶ 2 этап Второе поколение характеризуется созданием ЯП ассемблерного типа, позволяющих вместо бинарных и иных форматов машинных команд использовать символьные обозначения. Являясь существенным шагом вперед, ассемблерные языки все еще оставались машинно-зависимыми и пользователь должен был быть хорошо знаком с аппаратной средой конкретного типа ЭВМ. Ассемблерные программы затруднительны для чтения, более трудоемки при отладке, хотя и по сей день они используются для создания минимального по объему и максимального по производительности ПО.

3 этап В 1954 году начинается третье поколение языков программирования: в недрах корпорации IBM группой разработчиков во главе с Джоном Бэкусом (John Backus) был создан язык программирования Fortran. Значение этого события трудно переоценить. Это первый язык программирования высокого уровня. Ключевой идеей, отличающей новый язык от ассемблера, была концепция подпрограмм. В 1960 году был создан язык программирования Cobol. Он задумывался как язык для создания коммерческих приложений, и он стал таковым. На Коболе написаны тысячи прикладных коммерческих систем. Отличительной особенностью языка является возможность эффективной работы с большими массивами данных, что характерно именно коммерческих приложений. Популярность Кобола столь высока, что даже сейчас, при всех его недостатках (по структуре и замыслу Кобол во многом напоминает Фортран) появляются новые его диалекты и реализации.

3 этап

- ▶ 2 этап Второе поколение характеризуется созданием ЯП ассемблерного типа, позволяющих вместо бинарных и иных форматов машинных команд использовать символьные обозначения. Являясь существенным шагом вперед, ассемблерные языки все еще оставались машинно-зависимыми и пользователь должен был быть хорошо знаком с аппаратной средой конкретного типа ЭВМ. Ассемблерные программы затруднительны для чтения, более трудоемки при отладке, хотя и по сей день они используются для создания минимального по объему и максимального по производительности ПО.

- ▶ 11 3 этап В 1954 году начинается третье поколение языков программирования: в недрах корпорации IBM группой разработчиков во главе с Джоном Бэкусом (John Backus) был создан язык программирования Fortran. Значение этого события трудно переоценить. Это первый язык программирования высокого уровня. Ключевой идеей, отличающей новый язык от ассемблера, была концепция подпрограмм. В 1960 году был создан язык программирования Cobol. Он задумывался как язык для создания коммерческих приложений, и он стал таковым. На Коболе написаны тысячи прикладных коммерческих систем. Отличительной особенностью языка является возможность эффективной работы с большими массивами данных, что характерно именно коммерческих приложений. Популярность Кобола столь высока, что даже сейчас, при всех его недостатках (по структуре и замыслу Кобол во многом напоминает Фортран) появляются новые его диалекты и реализации.

4 этап

- ▶ 4 этап Языки четвертого поколения носили непроцедурный характер, т.е. программы на таких языках описывают только что надо делать, но не как это сделать. Типичными примерами являются языки искусственного интеллекта или логического программирования. Родоначальником таких языков стал язык Prolog, появившийся в 1971 году. Вторым направлением эволюции ЯВУ явилось появление объектно-ориентированных языков программирования. Первым объектно-ориентированным языком был язык Simula (1967). Этот язык был предназначен для моделирования различных объектов и процессов, и объектно-ориентированные черты появились в нем именно для описания свойств модельных объектов. объектно-ориентированных

5 этап

- ▶ 5 этап Широкое распространением так называемые скриптовые языки. К пятому поколению языков программирования также можно отнести языки искусственного интеллекта, экспертных систем и баз знаний, а также естественные языки.
- ▶ 5 этап Язык JavaScript был создан в компании Netscape Communications в качестве языка для описания сложного поведения веб-страниц. Интерпретируется браузером во время отображения веб-страницы. По синтаксису схож с Java и (отдаленно) с C/C++. Имеет возможность использовать встроенную в браузер объектную функциональность, однако подлинно объектно-ориентированным языком не является. VBScript был создан в корпорации Microsoft во многом в качестве альтернативы JavaScript. Имеет схожую область применения. Синтаксически схож с языком Visual Basic (и является усеченной версией последнего). Так же, как и JavaScript, исполняется браузером при отображении веб-страниц и имеет ту же степень объектно-ориентированности. Perl создавался в помощь системному администратору операционной системы Unix для обработки различного рода текстов и выделения нужной информации. Развился до мощного средства работы с текстами. Является интерпретируемым языком и реализован практически на всех существующих платформах. Применяется при обработке текстов, а также для динамической генерации веб-страниц на веб-серверах. Python - интерпретируемый объектно-ориентированный язык программирования. По структуре и области применения близок к Perl, однако менее распространен и более строг и логичен. Имеются реализации для большинства существующих платформ.

Спасибо за просмотр !

