ЯЗЫКИ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

ПЛАН

- Классификация языков программирования по поколениям ЭВМ
  - машинно-ориентированные языки
  - процедурные языки высокого уровня
  - декларативные языки программирования
  - объектно-ориентированные языки программирования
  - агентно-ориентированное программирование
- Классификация языков по технологии программирования
  - восходящее программирование
  - нисходящее программирование
  - процедурное программирование
  - модульное программирование
  - компонентное проектирование
  - языки написания сценариев

Машинно-ориентированные языки Первое поколение. 40-50-е годы

- машинные коды;
- ассемблеры;
- макроассемблеры.

Процедурные языки высокого уровня. Второе поколение 50-е – 70-е гг.

- FORTRAN (FORmula TRANslation) разработан в IBM в середине 1950-х для компьютеров серии 704 и долгое время интенсивно использовался в научном и инженерном программировании.
- COBOL (COmmon Business Oriented Language) ориентирован на решение экономических задач, управления и бизнеса.
- ALGOL (ALGOrithmic Language). ALGOL-60, ALGOL-68.

  ALGOL оказал сильное влияние на развитие теории программирования. Идеи ALGOLa о построении алгоритмов в виде последовательности процедур нашли отражение в языке PASCAL.
- РАЯ (Русский алгоритмический язык) - русифицированная версия ALGOLa.

  Разработчик академик А.П. Ершов.
- В середине 1960-х в IBM разработали язык **PL/I** (Programming Language One) для компьютеров IBM System/360.

Процедурные языки высокого уровня. Второе поколение 50-е – 70-е гг.

- BASIC (Beginners All-purpose Symbolic Instruction Code) разработан в 1964 г. Джоном Кемени (John Kemeny) и Томасом Курцем (Thomas Kurtz) из Дартмутского университета для обучения студентов.
- На домашних компьютерах дебют БЕЙСИКа состоялся в 1975 г., когда Билл Гейтс и Пол Аллен написали интерпретатор БЕЙСИКа для компьютера «Альтаир 8800». Этот интерпретатор стал первым продуктом основанной ими корпорации Microsoft.
- PASCAL разработан в 1970-71 гг. швейцарским профессором информатики Николасом Виртом. Язык унаследовал структуру АЛГОЛа и средства КОБОЛа для работы с записями.
- На Паскале частично основан язык программирования ADA, разработанный для Минобороны США. Он назван в честь Ады Аугусты Байрон.
- C. Этот чрезвычайно популярный язык был создан в 1969-1973 гг. в основном усилиями Денниса Ритчи из Bell Telephone Laboratories.

Все перечисленные языки относятся к *процедурным* языкам, предназначенным для компьютеров с архитектурой фон Неймана.

НЕПРОЦЕДУРНЫЕ ЯЗЫКИ ВЫСОКОГО УРОВНЯ
ДЕКЛАРАТИВНЫЕ (ОПИСАТЕЛЬНЫЕ) ЯЗЫКИ ПРОГРАММИРОВАНИЯ ТРЕТЬЕ ПОКОЛЕНИЕ. 70—80-Е ГОДЫ.

Функциональное (аппликативное) программирование –процесс вычисления трактуется как вычисление значений функций.

В функциональных языках программирования отсутствуют операторы: все действия, в том числе и управляющие конструкции, выполняются при помощи вызовов функций.

- Наиболее известные языки *функционального программирования:*
- XQuery
- Haskell — чистый функциональный. Назван в честь Хаскелла Карри.
- LISP (Джон МакКарти, 1958, множество его потомков, наиболее современные из которых — Scheme и Common Lisp).
- ML (Робин Милнер, 1979, из современных -Standard ML и Objective CAML).
- Miranda (Дэвид Тёрнер, 1985, который впоследствии дал развитие языку Haskell).
- Erlang — (Joe Armstrong, 1986) функц-й язык с поддержкой процессов.
- Nemerle — гибридный функционально/императивный язык.
- F# — функциональный язык для платформы .NET

НЕПРОЦЕДУРНЫЕ ЯЗЫКИ ВЫСОКОГО УРОВНЯ
ДЕКЛАРАТИВНЫЕ (ОПИСАТЕЛЬНЫЕ) ЯЗЫКИ ПРОГРАММИРОВАНИЯ ТРЕТЬЕ ПОКОЛЕНИЕ. 70—80-Е ГОДЫ.

Логическое программирование – программист должен сделать только постановку задачи, а поиск решения выполняет транслятор.

Язык логического программирования PROLOG был создан в начале 70-х годов Аланом Колмероэ.

Программа на этом языке строится из последовательности фактов и правил, а затем формулируется утверждение, которое PROLOG будет пытаться доказать с помощью введенных правил.

Функциональное и логическое программирование – представители декларативного программирования. Это стиль программирования, который в программах описывает способ решения поставленной задачи, а не предписывает шаги для получения результата.

Декларативные языки предназначены для решения задач искусственного интеллекта, их иногда называют языками программирования сверхвысокого уровня (ЯСВУ).

ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ. ЧЕТВЕРТОЕ ПОКОЛЕНИЕ. 80-90-Е ГОДЫ.

- SIMULA 67 — первый в мире объектно-ориентированный язык программирования. Разработан в конце 60-х в Норвегии. Это объектное расширение языка Algol 60.
- В середине 70-х годов Керниганом и Риччи активно изучается идея объектно-ориентированного программирования.
- OBJECT PASCAL — Язык программирования, разработанный в фирме Apple Computer в 1986 году группой Ларри Теслера.
- C++. Объектно-ориентированное расширение языка C. Язык возник в начале 1980-х годов (Бьёрн Страуструп - Bell Laboratories).
- C#. Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET.
- JAVA. Язык создан компанией Sun в начале 90-х годов на основе C++.
- SMALLTALK. Интерпретируемый язык, созданный в корпорации XEROX в 1980 году. Сегодня версия VisualAge for Smalltalk активно развивается компанией IBM.

АГЕНТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД К ПРОГРАММИРОВАНИЮ.

- Агентно-ориентированный подход — разновидность представления программ, или парадигма программирования, в которой основополагающими концепциями являются понятия агента и его поведения, которое зависит от среды, в которой он находится. Концепция была предложена Шохемом в 1990 г. Определение парадигмы, данное автором: Эту новую парадигму программирования вполне разумно назвать рациональным программированием. Точно также как объектно-ориентированное программирование сдвинуло парадигму с написания процедур к созданию объектов, рациональное программирование сдвинуло парадигму с создания информационных объектов к созданию мотивированных агентов.
- Агентом является всё, что может рассматриваться как воспринимающее свою среду с помощью датчиков и воздействующее на эту среду с помощью исполнительных механизмов.

АГЕНТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД К ПРОГРАММИРОВАНИЮ.

Это же определение в терминах Java:

Агент

|

|->Ресурсы

|

|->Объекты

|

|->функции

|

|->данные

АГЕНТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД К ПРОГРАММИРОВАНИЮ.

Типичным примером агентно-ориентированного подхода можно назвать специальные программы, имитирующие действия игроков в компьютерных играх.

Распространенное название для таких программ — боты.

Однако использование ботов не ограничивается применением в компьютерных играх. Их можно встретить как элемент программы в чатах и форумах, выполняющих роль «собеседника» и поддерживающих «дискуссии».

Однако некоторые боты выполняют не только роль имитаторов в процессе общения или игры, они используются для редактирования текстовой информации (в том числе и в Википедии), для наладки сетевых соединений, модерирования ресурсов и т. п.

ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ

- Процедурное (императивное)
  - операциональное
  - структурное
  - объектно-ориентированное
  - агентно-ориентированное
- Непроцедурное (декларативное)
  - функциональное (аппликативное)
  - логическое.

КЛАССИФИКАЦИЯ ЯЗЫКОВ ПО ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

- *восходящее программирование*
- *нисходящее программирование*
- *процедурное программирование*
- *модульное программирование*
- *компонентное проектирование*
- *языки написания сценариев*

ВОСХОДЯЩЕЕ ПРОЕКТИРОВАНИЕ («СНИЗУ-ВВЕРХ»).

- На первом этапе реализуют простые подпрограммы, из которых затем «собирают» сложную программу. (Fortran, Basic)
- Проблемы: обнаружение большого количества ошибок при сборке программного продукта.
- В результате был предложен новый подход к программированию, который был назван «структурным».

НИСХОДЯЩЕЕ ПРОЕКТИРОВАНИЕ («СВЕРХУ-ВНИЗ»).

- В основе структурного подхода лежит *декомпозиция* (разбиение на части) сложных задач с целью последующей реализации в виде отдельных небольших подпрограмм.
- Вначале задача решается в виде крупных блоков.
- Затем идет детализация каждого блока решения.

ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ

- Поддержка принципов структурного программирования была заложена в основу *процедурного программирования*. Языки включали основные «структурные» операторы передачи управления, поддерживали вложение подпрограмм, локализацию и ограничение области «видимости» данных: PL/1, ALGOL-68, Pascal, C.
- Стремление разграничить доступ к глобальным данным программы, чтобы уменьшить количество ошибок, привело к возникновению технологии модульного программирования.

МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ

- *Модульное программирование* предполагает выделение групп подпрограмм, использующих одни и те же глобальные данные, в отдельно компилируемые модули (библиотеки подпрограмм), например, модуль графических ресурсов, модуль подпрограмм вывода на принтер и т.п.
- Узким местом модульного программирования является то, что ошибка в интерфейсе при вызове подпрограммы выявляется только при выполнении программы (из-за раздельной компиляции модулей).

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

- При увеличении размера программы обычно возрастает сложность межмодульных интерфейсов, и с некоторого момента предусмотреть взаимовлияние отдельных частей программы становится практически невозможно. Для разработки программного обеспечения большого объема было предложено использовать объектный подход.
- Это технология создания сложного программного обеспечения, основанная на представлении программы в виде совокупности *объектов*, каждый из которых является экземпляром определенного типа (*класса*), а классы образуют иерархию с *наследованием* свойств.

ВИЗУАЛЬНОЕ ПРОГРАММИРОВАНИЕ

- С*редства визуального проектирования* программ для Windows:
- Basic: Microsoft Visual Basic;
- Pascal: Borland Delphi;
- C++: Visual C++, Borland C++Builder;
- Java: Symantec Café.

При использовании объектного подхода сохраняется зависимость модулей программного обеспечения от адресов экспортируемых полей и методов, а также структур и форматов данных.

Эта зависимость объективна, так как модули должны взаимодействовать между собой, обращаясь к ресурсам друг друга.

Связи модулей нельзя разорвать, но можно попробовать стандартизировать их взаимодействие, на чем и основан *компонентный подход к программированию*.

КОМПОНЕНТНЫЙ ПОДХОД К ПРОГРАММИРОВАНИЮ

Компонентный подход предполагает построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения, которые взаимодействуют между собой через *стандартизованные двоичные интерфейсы*. В отличие от обычных объектов объекты-компоненты можно собрать в динамически вызываемые библиотеки или исполняемые файлы, распространять в двоичном виде (без исходных текстов) и использовать в любом языке программирования, поддерживающем соответствующую технологию. Компонентные технологии разработки ПО поддерживают: JavaBeans, EJB, CORBA, ActiveX, VBA, COM, DCOM, .Net.

КОМПОНЕНТНЫЙ ПОДХОД К ПРОГРАММИРОВАНИЮ

- Частный случай компонентного подхода — доступ к серверным приложениям из браузеров через Интернет.
- Сегодня наиболее популярны три компонентные технологии:
  - **CORBA** консорциума OMG,
  - **Java Beans** компании Sun,
  - **COM+** корпорации Microsoft.

ЯЗЫКИ СОЗДАНИЯ СЦЕНАРИЕВ

С широким распространением Internet связано появление *языков создания сценариев (скрипт-языков)*. Языки являются интерпретируемыми, интерпретаторы для них распространяются бесплатно, а сами программы – в исходных текстах.

- **HTML**. Общеизвестный язык для оформления гипертекстовых документов. Он очень прост и содержит элементарные команды форматирования текста, добавления рисунков, задания шрифтов и цветов, организации ссылок и таблиц. Все Web-страницы написаны на языке HTML или используют его расширения.

- **Perl**. Разработан в 80-х годах Ларри Уоллом. Он задумывался как средство эффективной обработки больших текстовых файлов, генерации текстовых отчетов и управления задачами. По мощности Perl значительно превосходит языки типа С. В него введено много часто используемых функций работы со строками, массивами, всевозможные средства преобразования данных, управления процессами, работы с системной информацией и др.

ЯЗЫКИ СОЗДАНИЯ СЦЕНАРИЕВ

- **Tcl/Tk**. Разработан в конце 80-х годов Джоном Аустираутом (Tcl – язык, Tk – библиотека). Tcl состоит из мощных команд, предназначенных для работы с абстрактными нетипизированными объектами. Он независим от типа системы и при этом позволяет создавать программы с графическим интерфейсом.
- **VRML**. Был создан в 1994 году для организации виртуальных трехмерных интерфейсов в Интернете. Он позволяет описывать в текстовом виде различные трехмерные сцены, освещение и тени, текстуры (покрытия объектов), создавать свои миры, путешествовать по ним, «облетать» со всех сторон, вращать в любых направлениях, масштабировать, регулировать освещенность и т. д.
- **Java-скрипт**, **VB-скрипт**.
- Язык **PHP** один из доминирующих языков в области web - программирования.