Laboratório de Programação I

Introdução a Linguagem de Programação

Prof. Carlos Eduardo de B. Paes
Departamento de Ciência da Computação
PUC-SP

Computador

=

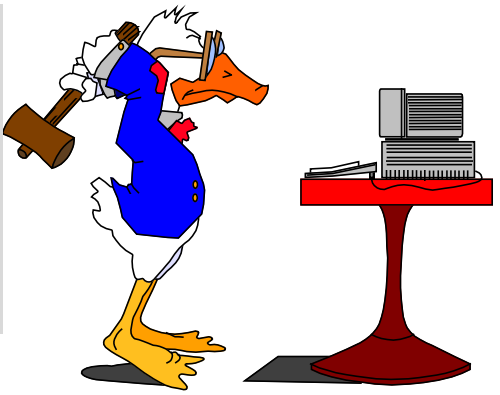
Hardware (tangível, partes mecânicas - "corpo")

+

Software (intangível, programas - "alma")

Para que serve um computador sem software ?

- **Sem sistema operacional ...**
- **Sem editor de textos ...**
- **Sem planilha de cálculos ...**
- **Sem navegador de internet ...**
- **Sem gerenciadores de bancos de dados ...**
- **Sem milhares e milhares de aplicações das quais nós dependemos mais e mais a cada dia que passa ?**



No Curso de Ciência da Computação ...

- **Aprende-se o que é e como funciona o hardware de um computador;**
- **A ênfase, porém, está no domínio das técnicas que permitem criar programas cada vez mais sofisticados e confiáveis, automatizando tarefas de alto-risco para seres humanos, aumentando a eficiência dos processos, reduzindo custos e, finalmente, aumentando a qualidade de vida das pessoas !**

Todo ofício possui as suas ferramentas.

Dentre as várias que são utilizadas pelo bacharel em ciência da computação, sem dúvida as mais importantes são as ...

... linguagens de programação,

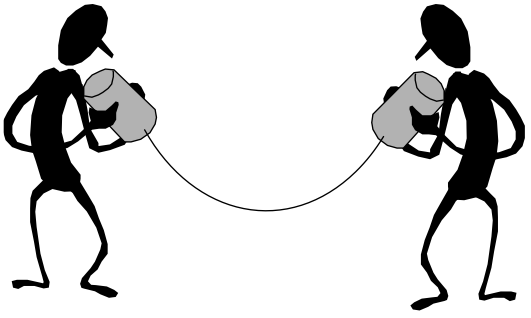
porque são com elas que são escritos os programas que tornam as máquinas úteis.

E é por isso que vocês estão iniciando o seu curso com as atenções voltadas para as linguagens de programação.

Linguagem de programação

Bebes se comunicam através de choro, gritos e sons ininteligíveis.

Linguagem de programação



Adultos se comunicam através da linguagem natural (português, inglês etc).

Linguagem de programação

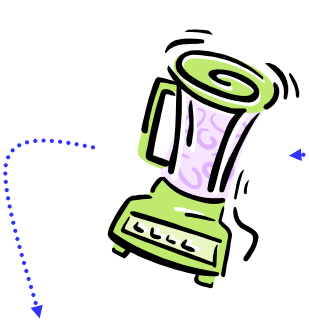
Computadores (internamente) entendem apenas a linguagem dos números.

Linguagem de programação

Como fazer pessoas e computadores se comunicarem ?

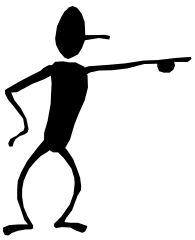
Linguagem de programação

"Por favor me ordene esta lista."



0111001101010110

Linguagem de programação



Pessoas utilizam linguagens artificiais que se parecem com as naturais - as linguagens de programação.

O compilador traduz a linguagem artificial escolhida para a linguagem numérica característica do computador-alvo.

Linguagem de programação

Linguagens de programação de alto-nível

Linguagens de montagem (baixo-nível)

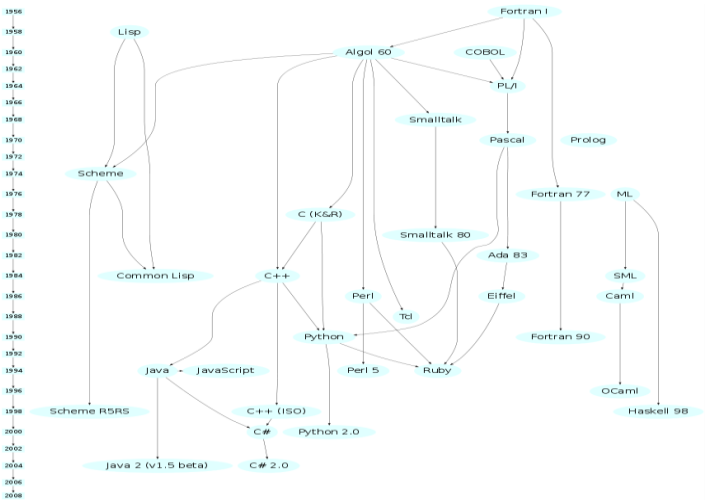
Linguagens de máquina (baixo-nível)

Poder
Dependência do hardware

Legibilidade
Expressividade
Abstração
Segurança

Linguagem de programação

- Total de linguagens de programação ≈2500



Aspectos preliminares

- ***Motivação para estudar linguagens de programação;***
- ***Áreas de aplicação das linguagens de programação;***
- ***Critérios de avaliação de uma linguagem de programação;***
- ***Aspectos econômicos;***
- ***Fatores que influenciam o projeto de uma linguagem de programação;***
- ***Principais categorias de linguagens;***
- ***Métodos de implementação de linguagens de programação***

Motivação

- **Melhoria na capacidade de expressar idéias**
  - Nossa capacidade intelectual é influenciada pelo poder expressivo da linguagem através da qual comunicamos nossos pensamentos.
  - É difícil conceber estruturas ou idéias abstratas que não possam ser facilmente verbalizadas ou redigidas.
  - A abstração (ou modelagem da solução de um problema) é uma propriedade intelectual que possui um papel fundamental na ciência da computação.
  - Sem boas linguagens não há boas abstrações. Sem boas abstrações não há bons programas

Motivação

- **Melhoria na capacidade de expressar idéias**
  - Nossa capacidade intelectual é influenciada pelo poder expressivo da linguagem através da qual comunicamos nossos pensamentos.
  - É difícil conceber estruturas ou idéias abstratas que não possam ser facilmente verbalizadas ou redigidas.
  - A abstração (ou modelagem da solução de um problema) é uma propriedade intelectual que possui um papel fundamental na ciência da computação.
  - Sem boas linguagens não há boas abstrações. Sem boas abstrações não há bons programas.

Motivação

- **Maior qualidade no processo de seleção**
  - A proliferação das linguagens fez com que haja grande aderência destas em relação à áreas específicas de aplicação;
  - O uso repetitivo das mesmas poucas linguagens apresenta grandes riscos (obsolescência, inadequação etc);
  - Portanto, a escolha da melhor linguagem, ou pelo menos da mais adequada, pode ser fator determinante de sucesso ou fracasso em um projeto;
  - O conhecimento profundo de linguagens de programação, assim como da área de aplicação, é o fator que irá fazer a diferença entre uma boa e uma má escolha

Motivação

- **Maior capacidade de aprendizado**
  - Novas linguagens surgem todos os dias e a grande quantidade de linguagens existentes hoje em dia já constitui um fator impeditivo para que se busque o domínio de todas elas;
  - Sendo assim, é preferível dominar os conceitos fundamentais que lhes são comuns, particularizando-os conforme a linguagem que se opte por utilizar num certo projeto;
  - A maioria das linguagens são instâncias de um conjunto básico de conceitos. Isto faz que, de uma forma geral, linguagens sejam diferentes na aparência porém similares na essência.
  - Ao dominar os conceitos e uma primeira linguagem, o aprendizado de uma segunda ficará mais fácil, de uma terceira ainda mais e assim por diante

Motivação

- **Entender a importância da implementação**
  - Implica o uso "consciente" da linguagem;
  - Por um lado, nos faz entender os motivos de certas construções serem como são e o porque da existência da restrições aparentemente arbitrárias;
  - Por outro lado, nos induz um sentimento crítico quanto ao desempenho do programa (tempo x espaço) em função das construções escolhidas, permitindo a obtenção de programas mais eficientes.
  - Mas não é essencial para o estudo das linguagens, em particular neste nível introdutório. Na disciplina da "Compiladores" este assunto será abordado com maior profundidade

Motivação

- **Projetar novas linguagens**
  - Talvez sejam poucos os que venham a projetar uma linguagem de grande porte inteiramente nova. Mas não serão poucos os que serão chamados a desenvolver linguagens de pequeno ou médio porte para interagir com usuários como um subsistema dentro de uma aplicação maior (ex:entrada de comandos);
  - Também não será incomum a necessidade de fazer manutenção em linguagens criadas por outros, introduzindo novas construções, corrigindo problemas ou simplesmente atualizando-as conforme o estado-da-arte na área, prolongando assim a sua vida útil;
  - Qualquer que seja o caso, o conhecimento de conceitos básicos de linguagens será de grande valia para que estas tarefas possam ser desempenhadas satisfatoriamente

Motivação

- **Acompanhar a evolução da computação**
  - Vários dos principais avanços da computação manifestaram-se primeiramente no campo das linguagens de programação;
  - Devido à limitações de programadores e gerentes em certos período, várias boas - e inovadoras - linguagens tiveram pouco sucesso em sua época; uma melhor capacidade para identificar avanços provavelmente reverteria em uma adoção mais intensa da mesma com mais vantagem para toda a comunidade de usuários (ex: Algol);
  - É uma área onde ainda hoje existe muita pesquisa. Universidades e empresas são muito ativas nesta área, cujo o objetivo é reduzir o custo do software, geralmente muito superior ao custo do hardware no qual eles são executados

Áreas de aplicação

- **Científica**
  - Os primeiros computadores foram desenvolvidos para fins militares e tinham por objetivo determinar a trajetória de mísseis durante a segunda guerra mundial;
  - Assim, desde a sua concepção, os computadores foram orientados à resolução de problemas que envolvem processamento númerico em larga escala ("number-crunching");
  - Nos dias de hoje, além do uso militar, muitas outras aplicações de caráter científico tem sido suportadas por computadores especializados em processamento numérico. FORTRAN foi a linguagem que inaugurou esta linhagem, e suas sucessoras continuam dominando a área

Áreas de aplicação

- **Comercial**
  - No lastro do sucesso do uso dos computadores na área científica, logo começaram a surgir linguagens orientadas ao desenvolvimento de programas de suporte a negócios, com acesso a bancos de dados, muitas facilidades para formatação de dados e geração de relatórios entre outras características;
  - A linguagem que prevaleceu foi o COBOL, largamente utilizada ainda nos dias de hoje; outras linguagens tiveram pouco sucesso;
  - Estima-se que a maior quantidade de programas em funcionamento nos dias de hoje tenha sido desenvolvido em COBOL;
  - Do ponto de vista teórico ou conceitual, foram poucas as contribuições desta classe de linguagens para o aprimoramento do estudo do assunto como um todo

Áreas de aplicação

- **Inteligência artificial**
  - Linguagens que foram desenvolvidas especificamente para lidar com problemas do ponto de vista simbólico e não numérico. Por exemplo, fazendo manipulação algébrica de equações ou acumulando conhecimento e fazendo inferências sobre o aprendizado;
  - As linguagens LISP e PROLOG foram as pioneiras nesta área, inicialmente restritas aos ambientes acadêmicos e centros avançados de pesquisa. Elas contribuíram não apenas com a solução de problemas considerados difíceis quando abordados através de linguagens tradicionais, como também trouxeram grandes contribuições ao desenvolvimento das linguagens de programação como um todo;
  - Atualmente há inúmeros descendentes destas linguagens, muitos deles sendo utilizados em ambientes de produção

Áreas de aplicação

- **Programação de sistemas**
  - Trata-se da área responsável pelo desenvolvimento dos programas considerados básicos na atividade de desenvolvimento de software. Incluem sistemas operacionais, editores de texto, compiladores, montadores etc, e são coletivamente chamados de software básico de um computador;
  - Por serem geralmente muito próximas do hardware, e também por carregarem o ônus de contribuir significativamente para a qualidade do software desenvolvido por seu intermédio, este conjunto de ferramentas necessitava ser desenvolvido com linguagens muito eficientes;
  - Assim, cada principal fabricante de computadores lançou a sua linguagem para programação de sistemas, linguagem esta que normalmente era bastante dependente do hardware por ele fabricado. As coisas começaram a mudar com o aparecimento da Linguagem Java, utilizada no sistema UNIX.

Áreas de aplicação

- **Controle em tempo-real**
  - Aplicações particularmente críticas no que se refere ao seu tamanho, tempo de execução, ou, especialmente, confiabilidade e segurança;
  - São usadas para desenvolver aplicações de alto-risco, tais como sistema de navegação de aeronaves, sistemas de controle de instrumentos médicos, monitoramento de usinas nucleares etc.
  - Tais aplicações demandam tempos de resposta muito curtos, altíssima confiabilidade, longos períodos de teste, manipulação simultânea de múltiplos eventos e facilidade de atualização ou correção sem que seja necessário desativar totalmente a aplicação;
  - A linguagem Ada, um projeto patrocinado pelo DoD dos EUA foi desenvolvida especialmente para satisfazer estes critérios, visto as deficiências nas demais linguagens então avaliadas

Áreas de aplicação

- **Scripting**
  - Conjuntos - geralmente simples - de comandos que orientam um certo sistema nas tarefas que este deve executar;
  - Incluem os arquivos de lotes do MS-DOS, o "shell" do UNIX, macros em diversos aplicativos e linguagens como AWK ou PERL, que funcionam como filtros sobre cadeias fornecidas como entradas;
  - São excelentes ferramentas no dia-a-dia do programador, e permitem resolver problemas simples com rapidez e facilidade. Algumas delas, no entanto, sofisticaram-se tanto que viraram verdadeiras linguagens de programação.
  - Apesar da sua pouca contribuição para as linguagens de propósito geral, as linguagens de "scripting" são elementos valiosos na bagagem de qualquer cientista da computação

Áreas de aplicação

- **Altamente especializadas**
  - RPG, para a produção de relatórios comerciais;
  - GPSS e SIMULA, para simulação;
  - SNOBOL, para processamento de cadeias de caracteres;
  - CHILL, para programação de centrais telefônicas digitais;
  - HTML, para representação de páginas ilustradas na internet;
  - Todas as linguagens de máquina;
  - E muitas, muitas outras linguagens com alto grau de especialização.
  - De uma forma geral, não possuem interesse didático sob o ponto de vista do estudo dos conceitos básicos de linguagens de programação

Áreas de aplicação

- **Propósito geral**
  - Utilizadas em desenvolvimentos que não tenham demanda por características ou comportamentos excessivamente particulares ou dependentes de máquina;
  - Procuram privilegiar a clareza na exposição das idéias, a facilidade de leitura, de documentação e de manutenção, apresentam níveis satisfatórios de segurança e confiabilidade e são aplicáveis a uma vasta gama de problemas;
  - C; C++
  - JAVA; Pascal;
  - De uma forma geral, apresentam elevado interesse didático sob o ponto de vista do estudo dos conceitos básicos de linguagens de programação. Possuem excelentes realizações comerciais

Critérios de avaliação

- O estabelecimento de critérios de avaliação possibilita a comparação entre projetos diferentes;
- A comparação crítica de projetos realça falhas e virtudes das linguagens em questão, fundamentando tecnicamente a escolha por uma ou outra;
- Não existe consenso sobre os melhores critérios de avaliação;
- Cada critério é influenciado por um conjunto de fatores

Critérios de avaliação

- **Fator Simplicidade**
  - Poucos componentes básicos
  - Única ou poucas maneiras de realizar uma certa operação particular
  - Cada símbolo com um único significado
  - Estruturação adequada do fluxo de controle e dos dados
- **Fator Ortogonalidade**
  - Conjunto pequeno de primitivas de controle e de dados pode ser combinado de forma arbitrária através de um conjunto também pequeno de operações;
  - Ortogonalidade implica ausência de exceções e maior regularidade;
  - Excesso de ortogonalidade possibilita o surgimento de construções muito complexas e de difícil entendimento

Critérios de avaliação

- **Fator Estruturas de controle**
  - Simulação através de desvios incondicionais arbitrários é muito mais difícil.
  - Goto or not goto ?
- **Fator Tipos e estruturas de dados**
  - Quando suficientemente ricos e variados, possibilitam um melhor mapeamento da abstração ou do objeto que se deseja modelar em termos da linguagem escolhida com elevado grau de naturalidade

Critérios de avaliação

- **Fator Sintaxe e semântica**
  - Sintaxe (forma) determina a naturalidade com que um texto pode ser lido ou escrito; quanto mais próxima da linguagem natural melhor, quanto mais consistente também;
  - Semântica (significado) não está associado diretamente à sintaxe, porém esta última pode ser elaborada de forma a sugerir o significado das construções;
- **Fator Suporte para abstração**
  - Não vinculação entre as relações entre os objetos ou componentes de um programa e os detalhes da sua realização física;
  - Permite criar programas que representam mais naturalmente fenômenos ou processos reais;
  - Abstração de dados e abstração de processo

Critérios de avaliação

- **Fator Expressividade**
  - Escolha de construções compactas que executem várias tarefas de forma combinada, ou ainda, construções similares mas que, uma vez dada a opção de escolha, permitem maior comodidade e naturalidade para o programador.
- **Fator Verificação de tipos**
  - Processo que visa garantir a validade das combinações entre operandos e operadores conforme as regras da linguagem. Visa detectar erros de programação, tanto durante a compilação quanto durante a execução dos programas

Critérios de avaliação

- **Fator Tratamento de exceções**
  - Mecanismos que permitem a detecção e a intervenção, pelo próprio programa, de situações de erro durante a sua execução, evitando o término anormal do mesmo
- **Fator "Aliasing" (sinônimos)**
  - Mecanismo que permite que um mesmo objeto seja referenciado de mais de uma forma diferente, tornando a análise do programa mais complexa, produzindo efeitos colaterais indesejáveis e dificultando a realização de algumas otimizações pelo compilador.

Aspectos econômicos

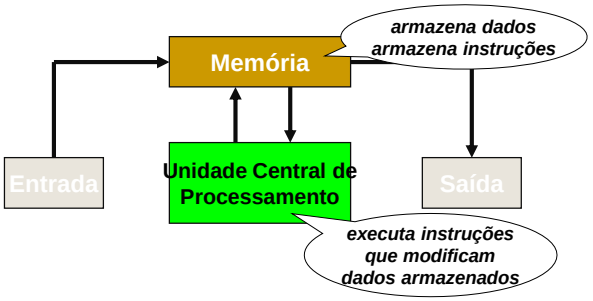
- Custo de uma linguagem de programação
  - Treinamento de programadores
  - Criação de programas
  - Compilação
  - Execução
  - Construção do compilador e ferramentas / ambientes associados
  - Conseqüências oriundas da confiabilidade dos programas
  - Manutenção
  - Portabilidade
  - Generalidade
  - Qualidade e disponibilidade da definição

Influências no projeto

- Arquitetura do computador
  - Modelo de von Neumann
    - Projeto teórico que identificou e sistematizou a função dos principais blocos funcionais de um computador moderno e a forma como eles devem ser organizados para compor um sistema eficiente de processamento de dados;
    - A arquitetura da grande maioria dos computadores
    - - desde a década de 40 até hoje - é baseada ainda neste modelo de computação.

Influências no projeto

- Arquitetura do computador
  - Modelo de von Neumann



Influências no projeto

- Arquitetura do computador
  - Este modelo exerceu grande influência sobre o projeto das principais linguagens desde então. Por fazer uso extensivo de posições de memória e seus respectivos conteúdos, assim como da execução seqüencial e iterativa de comandos, deu origem às linguagens imperativas, grande maioria nos dias de hoje;
  - Linguagens diferenciadas (funcionais e lógicas, por exemplo), são simuladas em máquinas de von Neumann, e arcam com uma certa ineficiência por isto;
  - Máquinas paralelas são exploradas através de adaptações nas linguagens imperativas tradicionais.
  - Novas descobertas na área de arquitetura irão ter muita influência sobre o projeto de novas linguagens, possivelmente reabilitando algumas atuais também

Influências no projeto

- Metodologia de desenvolvimento
  - A crise do software:
  - A partir de 60/70, os computadores começaram a baixar de preço e a serem mais amplamente utilizados;
  - Novas e mais complexas aplicações surgiam todos os dias;
  - O custo de desenvolvimento de software começou a ultrapassar os custos de projeto de hardware;
  - Percebeu-se a necessidade de analisar não apenas hardware e linguagens, mas também o processo de desenvolvimento de software;
  - Surge o conceito de metodologia de desenvolvimento de software.

Influências no projeto

- Metodologia de desenvolvimento
  - Ênfase no processo:
    - Projeto "ad-hoc" de programas;
    - Programação estruturada, top-down e com refinamentos graduais;
    - Modularização;
    - Concorrência;
  - Ênfase nos dados:
    - Estruturação dos dados;
    - Abstração de dados;
    - Orientação a objetos;
  - Modelos de ciclo de desenvolvimento:
    - Requisitos, especificação, codificação, testes, documentação etc..

Influências no projeto

- Metodologia de desenvolvimento
  - Conclusão:
    - Todos estes fatores (ou fases) possuíram grande influência sobre as linguagens de programação projetadas na época, as quais visavam - e ainda visam - suportar uma ou mais das metodologias mais atuais ou mais aceitas;
    - Além das linguagens, surgiram os ambientes integrados e ferramentas complementares para auxiliar o programador na tarefa de criar e manter programas seguindo as normas da metodologia escolhida.

Influências no projeto

- Metodologia de desenvolvimento
  - Uma linguagem suporta uma metodologia se ela apresenta construções destinadas especificamente à representação e/ou manipulação dos principais conceitos abstratos da metodologia.
  - Uma linguagem apenas permite a aplicação de uma metodologia se for possível, de alguma forma, utilizar suas construções primitivas para implementar ou simular os conceitos abstratos da metodologia.
  - Linguagens devem suportar e não apenas permitir o uso de uma metodologia

Influências no projeto

- Aderência à área de aplicação
  - Linguagens de uso específico (que não sejam de propósito geral) devem apresentar construções que representem abstrações diretas dos elementos presentes nos problemas típicos da área à qual ela se destina;
  - Devem, também, oferecer operações pré-definidas sobre tais abstrações, garantido assim uma boa legibilidade dos programas, grande facilidade de escrita dos mesmos e possibilidade de verificações que garantam a confiabilidade e a segurança da aplicação;
  - Como conseqüência, elas acabam por permitir o seu uso por especialistas na área de aplicação, e não apenas por programadores profissionais ou cientistas da computação

Principais categorias

- Linguagens imperativas
  - Caracterizam-se por especificar em detalhes a execução dos programas;
  - A organização e a manipulação dos dados pode se apresentar de forma variada.
- Linguagens orientadas à objetos
  - Derivadas das imperativas, apresentam várias das suas principais características;
  - Diferenciam-se daquelas pela maior disciplina na organização e no acesso aos dados que, procuram modelar objetos da aplicação-alvo

Principais categorias

- Linguagens funcionais
  - Usadas em aplicações de IA, fazem amplo uso do conceito de funções, muito mais amplo do que o encontrado na maioria das linguagens convencionais.
- Linguagens lógicas
  - Diferentemente das demais, não especificam uma "computação" diretamente, mas sim um conjunto de regras e fatos através das quais é possível deduzir a solução de um problema. Por este motivo, são também chamadas de linguagens declarativas