

Aula 7

Revisão do conteúdo até agora

Sumário das aulas 5 e 6

- Duvidas sobre a lista 2
- Funções/procedimentos/métodos
 - relação com funções matemáticas;
 - dois “momentos” – declaração e uso;
 - parâmetros, variáveis e seu escopo

Conteúdo geral até agora

- História da computação
 - Desenvolvimento de hardware
 - Ábaco
 - Calculador analítico
 - Computador eletromecânico
 - Computador eletrônico
 - Desenvolvimento de software
 - Algoritmos (Ada Lovelace)
 - Linguagens
 - Sistemas Operacionais
 - Compiladores
 - Teoria
 - Decidibilidade (Church, Turing)
 - Programação estruturada (Böhm, Jacopini)

Organização de Computadores

- Modelo de Von Neumann
 - UC, ULA, Memória
- HIPO
 - Linguagem de máquina
 - Linguagem de montagem
- Compilação
- Hardware – SO – Aplicação
- Hardware – SO – JVM - Bytecode

Conteúdo de programação até agora

- variáveis primitivas
 - int, char, float, double, String*, boolean**
 - representações (ex. complemento de 2, IEEE754, ASCII, Unicode)
- Operações lógicas, aritméticas e relacionais
- Controle de fluxo de execução
 - if/then/else
 - switch/case
 - while
 - do..while
 - for
- funções/procedimentos/métodos
 - parâmetros, variáveis e escopo

Aula 7

Paradigmas de programação

Exemplos de linguagens...

- Linguagem de máquina
- ASSEMBLY
- ALGOL, COBOL, BASIC, FORTRAN
- Pascal, C, PHP, Python
- Smalltalk, C++, Java,
- Portugues, Ingles, Espanhol, Chines,...
 - (de onde pode se retirar um subconjunto equivalente a linguagens de programação)

Há algum problema que pode ser resolvido com uma linguagem de programação mas não com outra??

Há algum problema que pode ser resolvido com uma linguagem de programacao mas não com outra??

Para as linguagens citadas, não,
mas existem linguagens com domínio restrito, por exemplo, a que o GREP usa.

Podemos categorizar...

- Linguagem de maquina
- Linguagens de montagem
- Linguagens estruturadas
- Linguagens procedurais
- Linguagens orientadas a objetos
- Linguagem natural
 - O subconjunto citado pode ser chamado, por exemplo, “português estruturado” com que se escreve pseudo-código.

Paradigmas

- Categorias, ou modelos de programação
- Não são mutuamente exclusivos
- Algumas linguagens podem implementar mais de um paradigma.

Comparativo entre alguns paradigmas

Paradigma	Descrição	Característica notável / exemplo
Imperativo	Computação em termos de comandos que diretamente mudam o estado do programa	Atribuição direta, variáveis globais
Estruturado	Paradigma derivado do Imperativo em que a estrutura é mais facilmente entendida	Indentação, ausência de GOTO
Funcional	Trata computação como a avaliação de funções matemáticas. Não é possível mudar o estado fora da função.	Lambda-cálculo, Scheme, Haskell
Procedural	Derivado do paradigma estruturado e da noção de código modular (subrotinas e funções com efeitos colaterais)	Variáveis locais, procedimentos, modularização
Orientado a eventos	O fluxo do programa é determinado por eventos, como apertar botões no mouse...	Eventos, Loop de eventos, apontadores de eventos, execução assíncrona
Orientado a objetos	Trata conjuntos de variáveis e procedimentos (métodos) como objetos.	Classes, objetos, atributos, métodos, assinatura, passagem de mensagens, instanciação, herança, polimorfismo, sobrecarga, sobrescrita.

Linguagens também evoluem

- PHP orientado a objetos;
- Visual Basic;
- Python orientado a objetos;
- Desta forma, a categoria a que uma linguagem pertence também muda.

Aula 7.5

Paradigma orientado a objetos e
JAVA

Conceitos

- **Objeto** (em OOP) é uma abstração de uma “coisa” (para diferenciar de objeto concreto, abstrato,...em linguagem natural)
- **Abstração** é o processo pelo qual retemos o que é essencial para um dado objetivo, eliminando características irrelevantes
- **Classe** é um template para um objeto. Ela contém a definição dos atributos de um objeto (OOP) e os métodos que o objeto pode executar.
- **Atributos** são as características ou propriedades de um objeto e sempre são variáveis da classe.
- **Métodos** são um conjunto de procedimentos que pode ser invocados (chamados) para que se execute algo com o objeto.
- **Instância** é um membro de uma classe, que depois de corretamente preenchido, em tempo de execução, corresponde a um objeto.
- **Mensagem** é o nome dado ao processo de envio de dados de um objeto para outro. Em JAVA a mensagem para um objeto consiste apenas na invocação de um método desse objeto.
- **Encapsular** consiste em limitar escopos e construir códigos com módulos bem definidos.
- **Herança** é o nome dado à capacidade de uma classe ter os mesmos atributos e métodos de outra sem ser necessário duplicar código.
- **Polimorfismo** é o nome dado à possibilidade de existência de vários métodos de mesmo nome com comportamentos diferentes.

Exemplos: Informação de contexto

- Escrever um programa que resolve um problema
 - Entender o problema
 - Achar a solução adequada (nem sempre é possível apresentar uma solução “completa”) Note que essa solução é expressa em alguma linguagem (ex. ling. matemática).
 - Traduzir a solução para alguma linguagem de programação (ou seja, usar o que sabemos dessa linguagem, para expressar a solução nessa linguagem
 - é como tentar conversar com um estrangeiro)

Problemas

- Separar na caixa A os triângulos azuis e na caixa B os triângulos vermelhos. (Só há triângulos sobre a mesa)
- Separar na caixa A os triângulos retângulos e na B os triângulos não retângulos
- Separar na caixa A os triângulos azuis e na caixa B os triângulos vermelhos. (Não há apenas triângulos sobre a mesa)
- Quais são os objetos, atributos e métodos relevantes para a solução dos problemas?
- Como pode-se representar os objetos para conseguir resolver cada problema?
- É necessário considerar a mesa para resolver o problema?

Note que dependendo do problema,
diferentes objetos, atributos e
métodos são necessários

Solução OOP dos problemas