

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Campus CAMPO MOURÃO

Resenha Capítulo 2 - APOO

Polimorfismo

Borges e Clinio

Estudante: Reginaldo Gregório de Souza Neto

RA: 2252813

Polimorfismo se trata da capacidade que um código possui de se comportar de diferentes maneiras em contextos diversos no tempo de execução. O polimorfismo necessita de um grau de abstração muito alto, entretanto facilita a incorporação de novos trechos de código em um sistema pré-existente. Em C++ o polimorfismo se dá através da conversão de ponteiros (ou referências) para objetos.

Normalmente se utilizam objetos de uma hierarquia de classes, um vez que uma classe derivada (HOMEM) é capaz de usufruir de todos os atributos de uma classe base (PESSOA). Portanto, podemos dizer que a classe homem pode ser vista como uma pessoa, pois ela possui todos os atributos de uma pessoa. Mas o oposto não é verdadeiro, tendo em vista que a classe pessoa não pode usufruir dos atributos específicos de homem. Essa conversão de perspectiva não é feita através de uma “cópia” de classes, mas sim da conversão de ponteiros.

Exemplo:

PonteiroHOMEM **PODE** apontar para PESSOA;

PonteiroHOMEM **PODE** apontar para HOMEM;

PonteiroPESSOA **PODE** apontar para PESSOA;

PonteiroPESSOA **NÃO PODE** apontar para HOMEM;

Estas conversões só podem ser feitas quando a herança é pública. Se a herança for privada a conversão não é permitida.

Através da redefinição dos métodos em uma hierarquia é possível "sobrescrever" ou "ocultar" métodos de uma classe básica, se houver outro método dentro da classe filha com o mesmo nome. Os métodos da classe pai continuam acessíveis, porém é preciso utilizar um operador de escopo para ser encontrado (::).