S T Q Q S S D

__/__/__

Capítulo 1 Pensando em Programação Orientada a Objetos

A programação (POO) é apresentada como uma ideia revolucionária que transformou a forma de programar.
Embora seus fundamentos tenham surgido nos anos 1960, foi na década de 80 que ganhou destaque com a linguagem Smalltalk e a conferência OOPSLA (1986).

1.1 Porque POO é popular?

Primeiro, ela escala bem, servindo tanto para problemas simples como para grandes sistemas corporativos.
Segundo, sua forma de abstração está mais próxima do maneira natural como os seres humanos lidam com os problemas: identificamos objetos, atribuímos, responsabilidades e estabelecemos relações entre eles.
Terceiro, esta linguagem lida com bibliotecas externas, que aceleram o desenvolvimento.

1.2 Linguagem e pensamento

A linguagem molda o pensamento, ela não limita oque podemos resolver, mas direciona o caminho pelo qual chegamos à solução.

1.2.1 Esquimós e neve

O exemplo de que os linguas esquimós tem muitas palavras para neve, ilustra como uma comunidade desenvolve um vocabulário especializado. A linguagem que uma pessoa fala pode levá-la a ver o mundo de uma maneira diferente, mas não a obriga a isso.
Do mesmo forma, usar POO simplifica a criação de soluções OO, mas não força o programador a pensar desta maneira.

__/__/__ S T Q Q S S D

1.2.2 Um exemplo de linguagens de computador

Dois estudantes levaram o mesmo problema de biologia computacional: analisar padrões em sequências de DNA.

- O primeiro programou em FORTRAN.

Essa linguagem trabalhava de forma procedimental, dava foco ao passo a passo de cálculos. Ele fez vários operações repetitivas e isso teve complexidade de $O(N^2)$. No fim o programa foi muito lento para grandes quantidades de dados.

- O segundo aluno usou APL.

Ele reorganizou os dados de DNA em uma matriz e aplicou operações matriciais de uma vez só. Com isso, o numero de operações foi para $O(N \log N)$. Conforme a quantidade de dados cresce, o tempo cresce devagar, sendo melhor que o Fortran.

1.2.3 Conjectura de Church e a hipotese de Whorf

- hipótese de Sapir-Whorf sugere que algumas ideias expressas em uma linguagem podem ser introduzíveis em outra.
- Church na ciência da computação afirma que qualquer computação pode ser realizado por uma maquina de Turing, o que implica que todas as linguagens com recursos mínimos são equivalentes em seu poder computacional.

1.3 Um novo Paradigma

A POO é chamada de novo paradigma de programação, esse termo, vem do Filósofo Thomas Kuhn, representa uma forma de enxergar o mundo, um modelo de organização de conhecimento. Paradigmas antigos são substituídos quando novas formas de pensar se mostram mais eficazes. No Programação, já houve: Programação imperativo (C, Pascal), Programação lógica (Prolog), Programação funcional (Haskell). A POO se baseia em classificação hierarquica, semelhante ao sistema biológico criado por lineanos, ela classifica classe e hierarquias, permitindo reuso e organização do conhecimento.

S T Q Q S S D

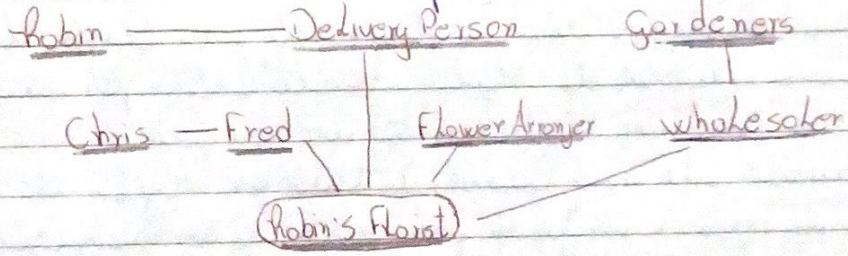
__/__/__

1.4 Uma maneira de ver o mundo

Para ilustrar os conceitos de POO, o livro traz a história de Chris querendo enviar flores para seu amigo Robin em outra cidade. Chris não entrega pessoalmente, mas vai a um florista local, Fred, e passa a ele uma mensagem com sua solicitação.

1.4.1 Agentes e comunidades

A solução para o problema envolveu a cooperação de muitos agentes (florista, um entregador etc) Um programa O.O é estruturado da mesma forma: como uma comunidade de agentes interativos chamados objetos. E cada um tem uma função.



1.4.2 Mensagem e métodos

A ação começa com transmissão de uma mensagem a um agente (objeto) responsável por essa ação. A mensagem codifica a solicitação e é acompanhado pelos argumentos necessários. O objeto que recebe a mensagem (o receptor) executa um método para atender à solicitação. Um princípio fundamental é o ocultação de informações: o remetente não precisa saber como a solicitação será atendida. Uma mensagem difere de uma chamada de procedimentos em dois aspectos: a mensagem tem um receptor designado, e a interpretação da mensagem (o método a ser executado) é determinada pelo próprio receptor. Isso permite a "ligação tardia", onde a decisão sobre qual método é invocar só é feito em tempo de execução.

spiral

__/__/__ S T Q Q S S D

1.4.3 Responsabilidades

A POO descreve o comportamento em termos de responsabilidades. Chefs apenas informam o resultado desejado (flores entregues a Helen), Fred tem a responsabilidade e a liberdade de usar qualquer técnica para atingir esse objetivo. Essa abordagem eleva o nível de abstração e promove a independência entre os objetos. O conjunto de responsabilidades de um objeto é chamado de seu "protocolo". A mudança de perspectiva é resumida em uma frase. Pergunte o que suas estruturas de dados podem fazer por você.

1.4.4 Classes e Instâncias

Todos os objetos são instâncias de uma classe. Uma classe representa uma categoria de objetos semelhantes (por exemplo, florista). O método que um objeto invoca em resposta a uma mensagem é determinado pela sua classe, e todos os objetos da mesma classe usam o mesmo método para responder às mensagens semelhantes.

1.4.5 Hierarquias de classes - herança

As classes podem ser organizadas em uma estrutura de herança hierárquica. Por exemplo, Fred é um florista que pode ser um exemplo de subclasse de logista, que é subclasse de humano... Uma subclasse herda atributos e comportamentos da superclasse. Classes como mamíferos podem ser classes mãe abstratas, que não possuem instâncias diretas, mas servem para agrupar subclasses.

1.4.6 Vinculação e substituição de métodos

Para lidar com exceções a uma regra geral, uma subclasse pode substituir informações herdadas de uma classe mãe. Quando uma mensagem é enviada a um objeto, a busca por um método correspondente começa na classe do objeto, sobe na classe mãe até que um método seja encontrado. Ele é executado e é chamado de substituição. A capacidade de vários objetos responderem a mesma mensagem com métodos diferentes é polimorfismo.

S T Q Q S S D

__/__/__

1.4.7 Resumo dos Conceitos Orientados a Objetos

Alan Kay identificou 6 características fundamentais

1. Tudo é um objeto
2. A computação é realizada por objetos que se comunicam enviando e recebendo mensagens.
3. Cada objeto tem seu próprio memória, composto por outros objetos.
4. Todo obj. é instância de uma classe.
5. A classe é o repositório do comportamento de um objeto.
6. As classes são organizadas em uma única hierarquia de herança em forma de uma árvore.

1.5 Computação como Simulação

A visão tradicional de computação é um modelo de "processo - estado", onde o computador é um gerenciador de dados que move valores entre endereços de memória ("compartimentos"). Em contraste, o POO vê a computação como um universo de objetos que interagem pedindo uns aos outros para a realização das ações. Essa visão é muito semelhante à simulação de eventos discretos na qual o usuário cria modelos comportacionais de vários elementos e os coloca em movimento para interagir. Assim no POO, a computação é vista como simulação.

1.5.1 O poder da metáfora: Pensar nos problemas em termos de objetos com comportamentos e responsabilidades permite que os programadores usem a intuição de suas experiências cotidianas. Essa abordagem metafórica, como a do brinquedo "Sr. Cabeça de Batata", explica porque os novatos aprendem POO mais fácil que os experientes, que podem estar presos em visões mais antigas ou tradicionais.

spiral

__/__/__

1.5.2 Evitando regressão infinita

Nem todo trabalho pode ser feito simplesmente passando solicitações a outros objetos, pois levaria um ciclo infinito. Em algum ponto, um objeto precisa realizar trabalho real. Em linguagens híbridas (C++), isto é feito por métodos escritos na linguagem base não orientada a objetos. Em linguagens puras (Java) isso é realizado por operações primitivas ou "nativas" fornecidas pelo sistema.

1.6 Uma breve história

A programação orientada a objetos não é um fenômeno recente. Seus principais conceitos com objetos, classe e herança, foram desenvolvidos na década de 1960 como parte de linguagem.

- 1960 - Simula: primeira linguagem a introduzir objetos, classes e heranças, pensada para simulação.
- 1970 - Smalltalk: criada por Alan Kay e equipe no Xerox PARC. Focada em ser intuitiva até para crianças, consolidou a visão de POO como simulação.
- 1980 - C++: Bjarne criou uma extensão de C que adicionava classes e objetos, unindo eficiência com abstração.

Outras linguagens surgiram: Objective-C, Eiffel, Actor...

- 1986 - OOPSLA: primeira grande conferência sobre POO, marco da difusão mundial do tema.

Desde então, a POO passou de revolucionária a popular, transformando grande parte da ciência da computação.