Fatec
Rio Claro

Linguagem de Programação II

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

Pilares do Paradigma

Fatec
Rio Claro

O paradigma é sustentado por quatro pilares principais:

Abstração: focar nas características essenciais do objeto, ignorando detalhes irrelevantes.

Encapsulamento: proteger os dados, controlando como são acessados ou modificados.

Herança: permitir que uma classe herde atributos e métodos de outra, facilitando o reuso de código.

Polimorfismo: possibilitar que diferentes objetos respondam de formas distintas a uma mesma mensagem/método.

Orientação a Objetos

Conta bancária

Fatec
Rio Claro

Uma lista de operações que uma pessoa desejaria realizar com uma conta bancária incluiria:

- Criar (uma conta)

- Depositar

- Sacar

- Verificar saldo

A seguir, aqui está uma lista mínima dos dados necessários para representar uma conta bancária:

- Nome do cliente

- Senha

- Saldo

Conta bancária

Fatec
Rio Claro

Observe que todas as operações são palavras de ação (verbos) e todos os itens de dados são coisas (substantivos).

Uma conta bancária real certamente seria capaz de realizar muito mais operações e conteria dados adicionais (como endereço, número de telefone e número do Seguro Social do titular da conta), mas para manter a discussão clara, começarei com apenas essas quatro ações e três dados.

Além disso, para manter as coisas simples e focadas, definirei todos os valores como um número inteiro em dólares. Devo também ressaltar que, em um aplicativo bancário real, as senhas não seriam mantidas em texto simples (sem criptografia) como nestes exemplos.

Prática

Fatec
Rio Claro

Bank1_OneAccount.py

O programa começa inicializando três variáveis para representar os dados de uma conta.

Em seguida, ele exibe um menu que permite a escolha de operações.

O código principal do programa atua diretamente nas variáveis globais da conta.

Conta bancária

Fatec
Rio Claro

Neste exemplo, todas as ações estão no nível principal; não há funções no código.

O programa funciona bem, mas pode parecer um pouco longo.

Uma abordagem típica para tornar programas mais longos mais claros é mover o código relacionado para funções e fazer chamadas para essas funções.

Prática

Fatec
Rio Claro

Bank2_OneAccountWithFunctions.py

Nesta versão, criei uma função para cada uma das operações que identificamos para uma conta bancária (criar, verificar saldo , depositar e sacar) e reorganizei o código para que o código principal contenha chamadas para as diferentes funções.

Conta bancária

Fatec
Rio Claro

Como resultado, o programa principal é muito mais legível.

Por exemplo, se o usuário digitar d para indicar que deseja fazer um depósito, o código agora chama uma função chamada deposit(), passando o valor a ser depositado e a senha da conta inserida pelo usuário.

Conta bancária

Fatec
Rio Claro

No entanto, se você observar a definição de qualquer uma dessas funções — por exemplo, a função withdraw() — verá que o código usa instruções globais para acessar (obter ou definir) as variáveis que representam a conta.

Em Python, uma instrução global só é necessária se você quiser alterar o valor de uma variável global em uma função.

No entanto, estou usando-as aqui para deixar claro que essas funções se referem a variáveis globais, mesmo que estejam apenas obtendo um valor.

Prática

Fatec
Rio Claro

Bank3_TwoAccounts.py

Mesmo com apenas duas contas, você pode ver que essa abordagem sai do controle rapidamente. Primeiro, definimos três variáveis globais para cada conta.

Além disso, cada função agora tem uma instrução if para escolher qual conjunto de variáveis globais acessar ou alterar.

Conta bancária

Fatec
Rio Claro

Sempre que quisermos adicionar outra conta, precisaremos adicionar outro conjunto de variáveis globais e mais instruções if em cada função.

Essa abordagem simplesmente não é viável.

Precisamos de uma maneira diferente de lidar com um número arbitrário de contas.

Prática

Fatec
Rio Claro

Bank4_N_Accounts.py

No início do programa, defini todas as três listas como a lista vazia 1. Para criar uma nova conta, adicionei o valor apropriado a cada uma das três listas 2.

Lidando com múltiplas contas, utilizo o conceito básico de um número de conta bancária.

Sempre que um usuário cria uma conta, o código usa a função len() em uma das listas e retorna esse número como o número da conta do usuário.

Conta bancária

Fatec
Rio Claro

Os dados são mantidos como três listas globais do Python, onde cada lista representa uma coluna nesta tabela.

Por exemplo, como você pode ver na coluna destacada, todas as senhas são agrupadas em uma única lista.

Account number	Name	Password	Balance
0	Joe	soup	100
1	Mary	nuts	3550
2	Bill	frisbee	1000
3	Sue	xxyyzz	750
4	Henry	PW	10000

Prática

Fatec
Rio Claro

Bank5_Dictionary.py

Com essa abordagem, todos os dados associados a uma conta podem ser encontrados em um único dicionário.

Para criar uma nova conta, criamos um dicionário e o anexamos à lista de contas.

Cada conta recebe um número (um inteiro simples), e esse número deve ser fornecido ao realizar qualquer ação com a conta.

Conta bancária
Abordagens Não OO

Fatec
Rio Claro

Os exemplos mostrados neste capítulo compartilham um problema comum: todos os dados com os quais as funções operam são armazenados em uma ou mais variáveis globais. Pelos seguintes motivos, usar muitos dados globais com programação procedural é uma má prática de codificação:

Qualquer função que use e/ou altere dados globais não pode ser facilmente reutilizada em um programa diferente. Uma função que acessa dados globais está operando em dados que residem em um nível diferente (superior) do que o código da própria função. Essa função precisará de uma instrução global para acessar esses dados. Você não pode simplesmente pegar uma função que depende de dados globais e reutilizá-la em outro programa; ela só pode ser reutilizada em um programa com dados globais semelhantes.

Conta bancária
Abordagens Não OO

Fatec
Rio Claro

Muitos programas procedurais tendem a ter grandes coleções de variáveis globais. Por definição, uma variável global pode ser usada ou alterada por qualquer trecho de código em qualquer lugar do programa. Atribuições a variáveis globais costumam estar amplamente espalhadas por programas procedurais, tanto no código principal quanto dentro das funções.

Como os valores das variáveis podem mudar em qualquer lugar, pode ser extremamente difícil depurar e manter programas escritos dessa maneira. Funções escritas para usar dados globais frequentemente têm acesso a muitos dados.

Conta bancária
Abordagens Não OO

Fatec
Rio Claro

Quando uma função usa uma lista global, um dicionário ou qualquer outra estrutura de dados global, ela tem acesso a todos os dados dessa estrutura.

No entanto, normalmente, a função deve operar em apenas uma parte (ou apenas uma pequena quantidade) desses dados.

Ter a capacidade de ler e modificar quaisquer dados em uma estrutura de dados grande pode levar a erros, como usar ou sobrescrever acidentalmente dados que a função não pretendia acessar.

Fatec
Rio Claro

Dúvidas

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

Prática

Fatec
Rio Claro

Account.py

Discutimos várias maneiras diferentes de usar programação procedural para implementar a simulação e descrevemos alguns dos problemas que essa abordagem cria. Por fim, temos nesse código a primeira ideia de como seria o código que descreve uma conta bancária se fosse escrito usando uma classe.

Desenvolva uma simulação de banco, criando-se cinco objetos da classe modelo. Ajuste a classe com atributos e métodos que julgar necessário.

Bibliografia

KALB, Irv. Object-Oriented Python. San Francisco: No Starch Press, 2022. 416 p.