Programação Orientada a Objetos

Prof. Márcio Miguel Gomes

Persistência de Dados

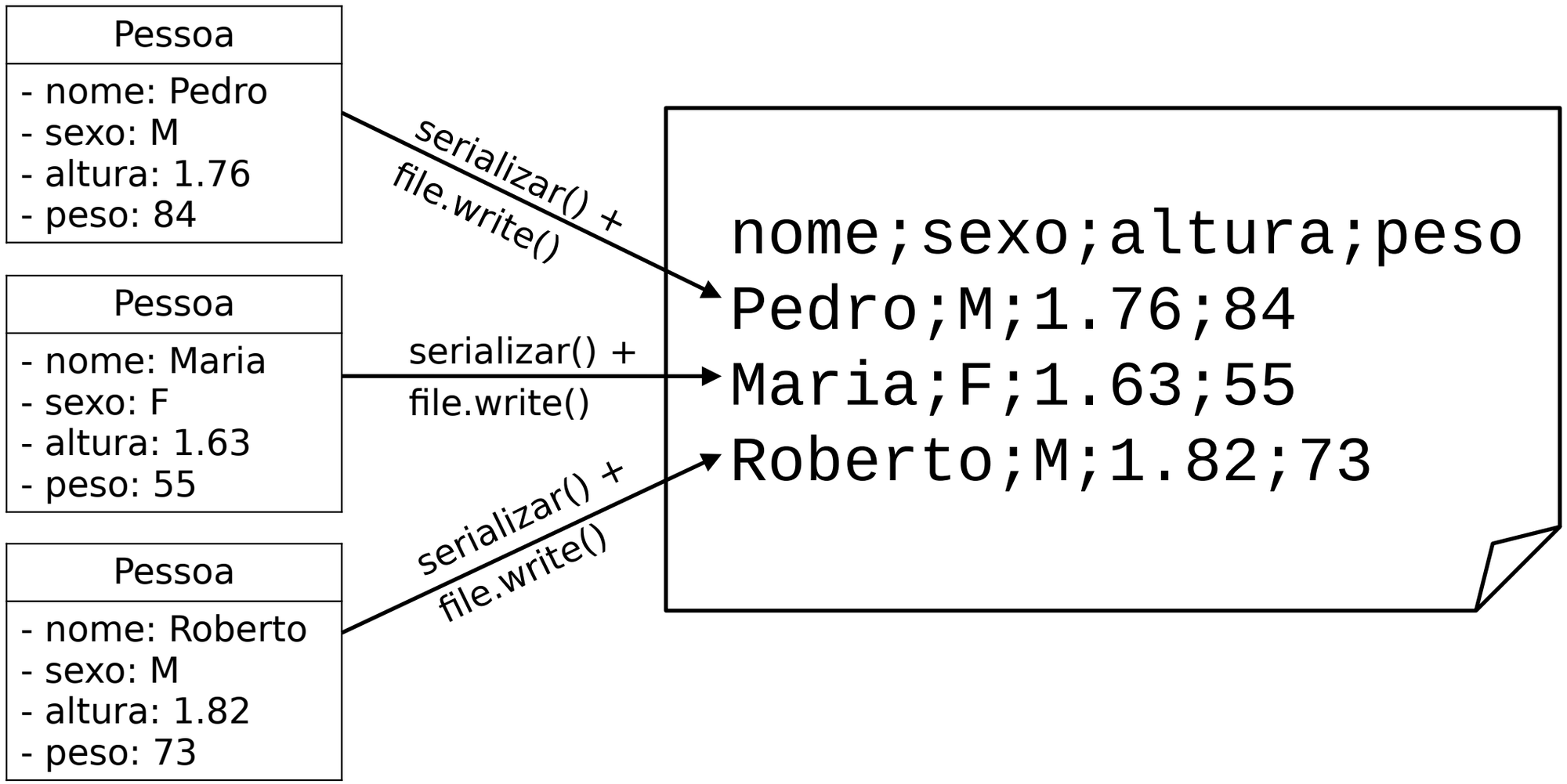
- Após a manipulação dos atributos dos objetos, pode ser necessário guardar as informações para uso futuro
- Para isso, salvamos os atributos dos objetos em um arquivo-texto estruturado, onde cada linha do arquivo representa um objeto
- Normalmente, convertemos todos os atributos de um objeto em uma *string* e usamos como separador os caracteres , ; ou <tab>
- Também existem formatos padronizados, como JSON, XML, CSV e TSV
- Para recuperar os objetos, cada linha do arquivo deve ser "instanciada" e seus dados carregados para dentro do objeto

JESUÍTAS BRASIL UNISINOS

Gravação

- Cada objeto deve ter uma função que retorna seus atributos padronizados no formato do arquivo, ou seja, uma lista com os valores dos atributos separados por um separador padrão
- Isso se chama “serializar” um objeto
- Então, para cada objeto que se quer salvar, deve-se chamar o método de “serialização” e gravar seus dados no arquivo

Gravação



JESUÍTAS BRASIL UNISINOS

Gravação

- Exemplo de algoritmo para salvar objetos:

1. Abrir o arquivo para armazenar os objetos
2. Adicionar um cabeçalho, caso necessário
3. Para cada objeto:
4. Chamar a função de “serialização”
5. Adicionar o objeto serializado no arquivo
6. Fechar o arquivo

Gravação

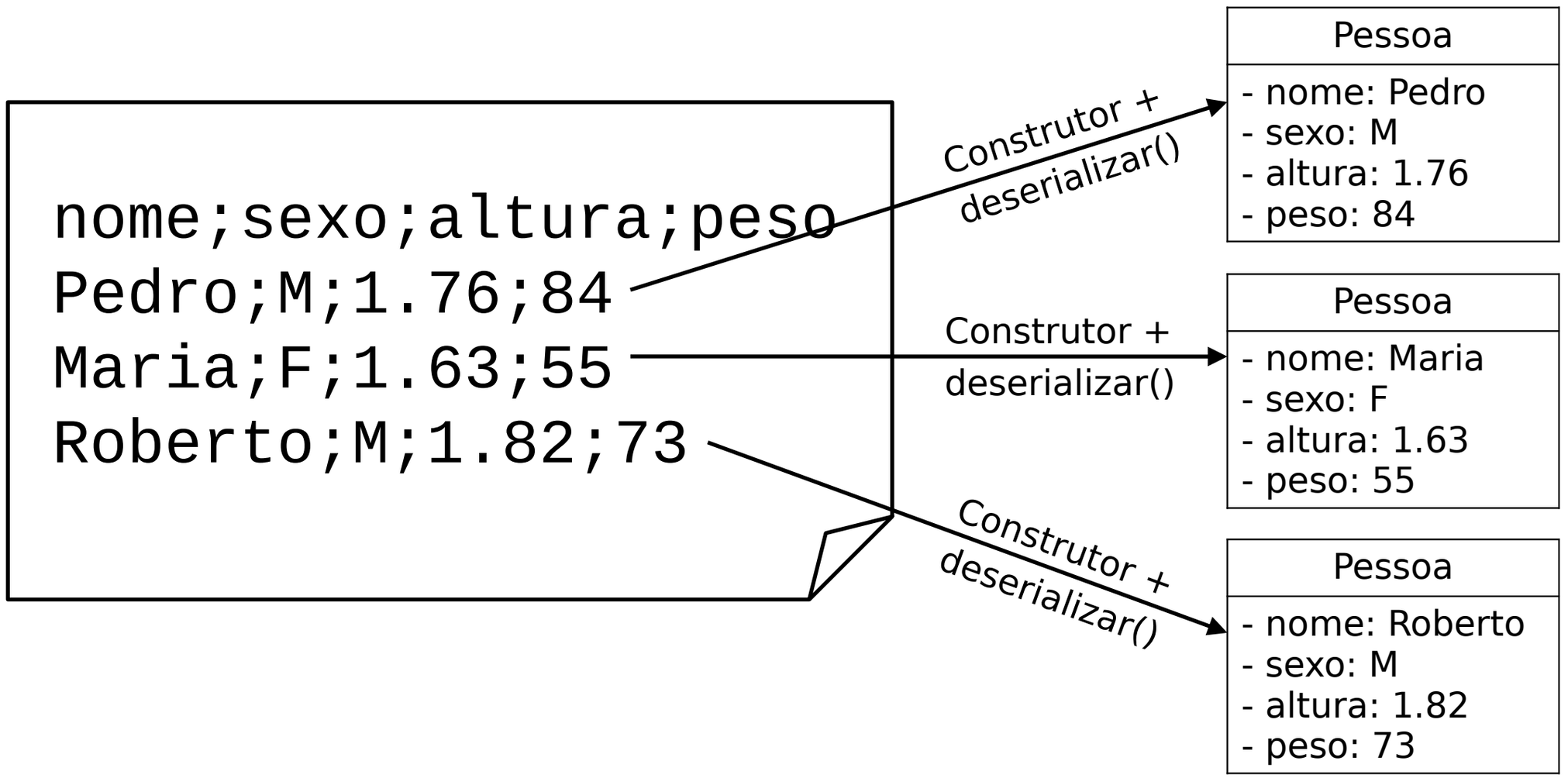
```python
class Pessoa:
  def __init__(self, nome, sexo, idade, altura):
    self._nome = nome
    self._sexo = sexo
    self._idade = idade
    self._altura = altura

  def serializar(self):
    return f'\n{self._nome};{self._sexo};{self._idade};{self._altura}'

if __name__ == '__main__':
  p = Pessoa('Pedro', 'M', 42, 1.76)
  arq = open('pessoa.txt', 'w')
  arq.write('nome;sexo;altura;peso') # Escreve o cabeçalho do arquivo
  arq.write(p.serializar())          # Escreve os dados da pessoa
  arq.close()
```


JESUÍTAS BRASIL UNISINOS

Leitura



JESUÍTAS BRASIL UNISINOS

Leitura

- A classe deve ter um método que recebe uma *string* representando uma única linha do arquivo de dados serializados
- Esse método analisa a *string*, identifica e separa cada dado e atualiza os atributos privados do objeto
- Isso se chama “deserializar” um objeto, e o processo de análise se chama “parse”
- Então, para cada objeto que se quer recuperar, deve-se ler uma linha do arquivo, instanciar dinamicamente o objeto e chamar o método de “deserialização”
- Dica: O próprio método construtor pode disparar a deserialização

JESUÍTAS BRASIL UNISINOS

Leitura

- Exemplo de algoritmo para recuperar objetos:

1. Abrir o arquivo contendo os objetos
2. Descartar o cabeçalho, caso exista
3. Para cada linha do arquivo:
4. Instanciar dinamicamente um objeto
5. Chamar a função “deserializar” passando a linha como parâmetro
6. Fechar o arquivo

Leitura

```
class Pessoa:
  def __init__(self, linha):
    self.deserializar(linha)

  def deserializar(self, linha):
    dados = linha.split(';')
    self._nome = dados[0]
    self._sexo = dados[1]
    self._idade = dados[2]
    self._altura = dados[3]

  def exibe_dados(self):
    print('Nome:', self._nome)
    print('Sexo:', self._sexo)
    print('Idade:', self._idade)
    print('Altura:', self._altura)

if __name__ == '__main__':
  arq = open('pessoa.txt')
  arq.readline() # Descarta o cabeçalho
  p = Pessoa(arq.readline())
  arq.close()
  p.exibe_dados()
```


JESUÍTAS BRASIL UNISINOS

Atividade

- Defina a classe “Veiculo” com os atributos código, fabricante, modelo, ano, cor e preço. Implemente uma função de *serialização* e outra de *deserialização* usando <tab> como separador dos dados
- Faça um programa com o seguinte menu:
  1. **Cadastrar**: pede todos os dados de um veículo, cria o objeto e salva em arquivo. Não deve apagar os dados preexistentes
  2. **Listar tudo**: mostra todos os veículos armazenados no arquivo, instanciando cada objeto e chamando o método exibe_dados()
  3. **Listar por cor**: pede para o usuário informar uma cor, e exibe todos os dados dos veículos com a cor informada
  4. **Listar por preço**: pede para o usuário informar um preço inicial e um final, e exibe todos os dados dos veículos com o preço dentro da faixa informada

Observação: O arquivo deve ter um cabeçalho com o nome de cada atributo