

# Laboratório de programação II

Classes  
Composição

Edson Moreno  
[edson.moreno@pucrs.br](mailto:edson.moreno@pucrs.br)

<http://www.inf.pucrs.br/~emoreno>

# Conceituação

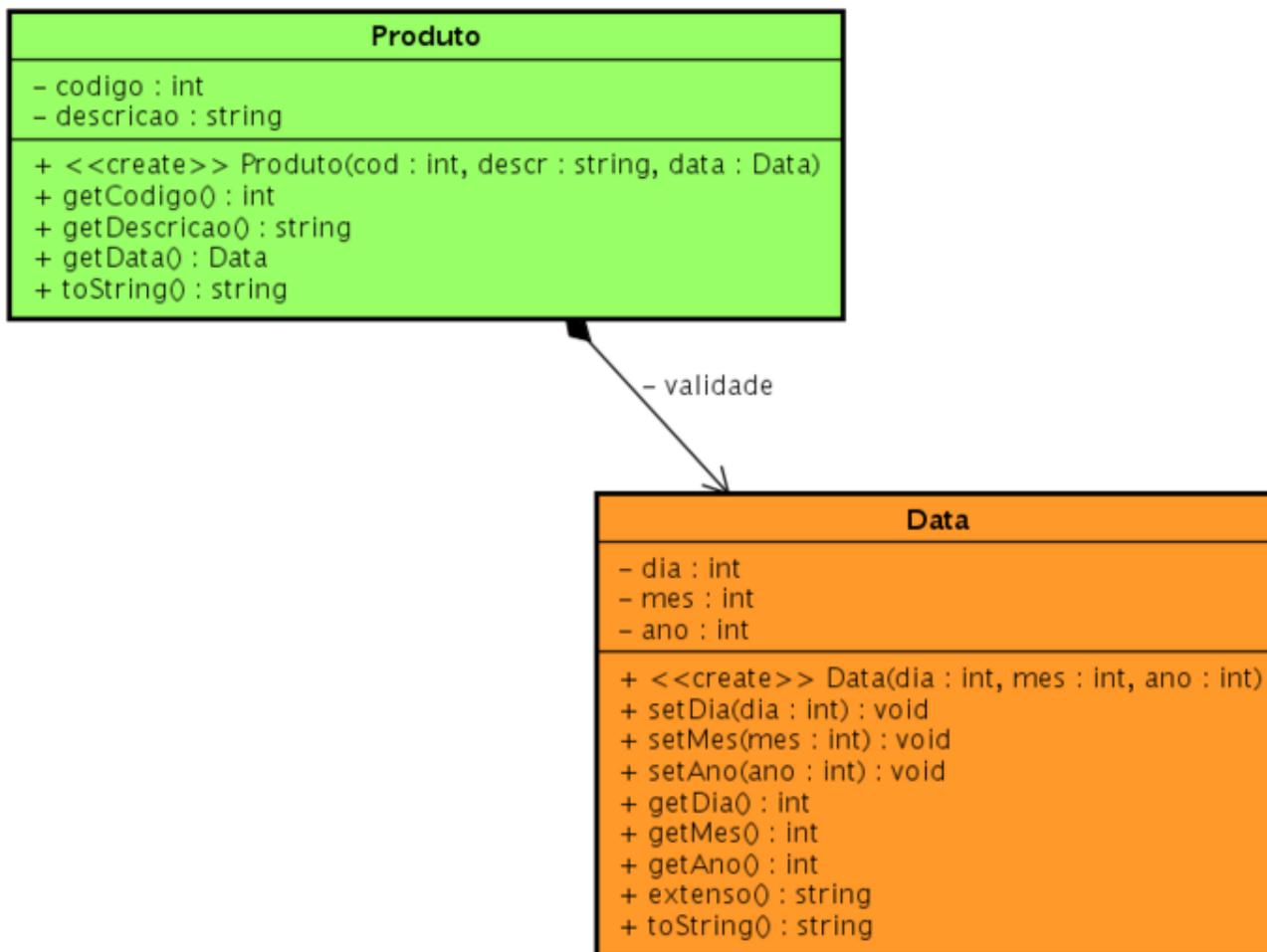
- Composição
  - Objetos fazendo parte de outros objetos
  - Princípio básico da engenharia de software
    - Módulos menores fazem parte de módulos maiores
- Exemplo
  - Carro
    - Motor – Um carro possui um motor
    - Pneus[4] – Um carro possui quatro rodas

# Conceituação

- Funcionamento
  - Objetos membros inicializados antes dos objetos que fazem parte
    - Pode-se explorar os métodos construtores para inicialização
- Exemplo
  - Um produto com data de validade
    - Data de validade é um atributo da classe produto
    - ... Mas data de validade também é um objeto da classe data
    - Desta forma pode-se dizer que
      - produto depende de data
      - Ou seja, produto utiliza os “serviços” da classe data

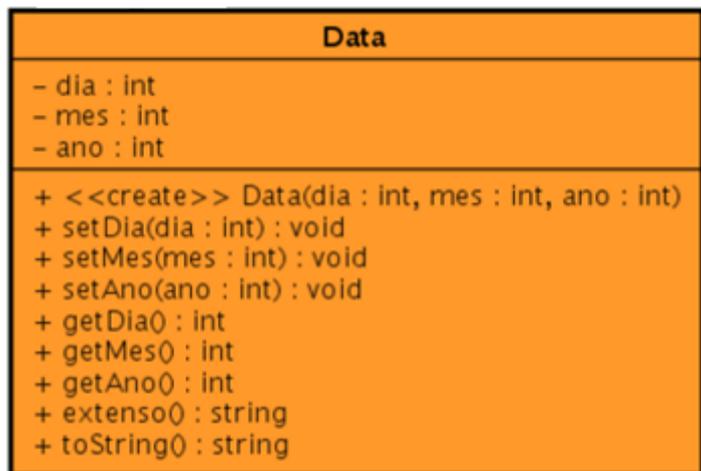
# Caso de uso

- Exemplo



# Caso de uso

- Classe data
  - Construtor com parâmetros padrão
    - Necessário para utilização dentro de outro objeto
  - Solução alternative
    - Construtor sem parâmetro + construtor com parâmetros

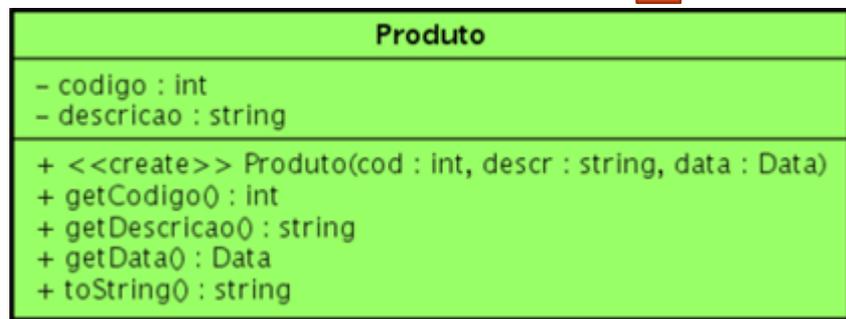


```
class Data {  
    public:  
        Data(int uDia=1, int uMes=1, int uAno=2012);  
        int getDia();  
        int getMes();  
        int getAno();  
        void setDia(int d);  
        void setMes(int m);  
        void setAno(int a);  
        string toString();  
    private:  
        int dia, mes, ano;  
};
```



# Caso de uso

- Classe Produto
  - Observar o construtor
    - **Const data &val**
      - Indica que o
        - Objeto é passado por parâmetro
        - Não pode ser alterado dentro do construtor



```
class Produto {
    public:
        Produto(int cod, string descr, const Data &val);
        int getCodigo();
        string getDescricao();
        Data getValidade();
        string toString();

    private:
        int codigo;
        string descricao;
        Data validade;
};
```

# Caso de uso

- Arquivo principal

```
int main()
{
    // Cria um objeto Data e já ajusta os seus atributos
    Data d1(5,5,2007);

    Produto p1(56,"Bolo de chocolate", d1);

    // Alternativa: cria-se um objeto "anônimo" na própria chamada
    // ao construtor de Produto

    Produto p2(57,"Bolo de laranja", Data(12,7,2007));

    // Escreve os dados de p1 e p2
    cout << p1.toString() << endl;
    cout << p2.toString() << endl;
}
```

- Compilação

- g++ data.cpp produto.cpp main.cpp -o run.x

# Exercício

- Criar uma classe automóvel que possui 4 pneus e um motor.
  - Considere que:
    - A classe automóvel possui uma marca e uma quilometragem atual;
    - A classe pneu tem uma marca e a pressão do ar presente no mesmo;
    - A classe motor apresenta uma certa quantidade de combustível e uma potência expressa em cavalos (hp).
  - Defina os métodos essenciais para essas classes.
  - No final, crie um programa principal que instancie um automóvel, contendo um motor e 4 pneus

# Exercício

- Faça um programa em C++ que contenha uma classe que representa um funcionário, registrando seu nome, salário e data de admissão. Crie por último uma classe que representa uma empresa, registrando seu nome e CNPJ. Em todas as classes defina os atributos como privados e crie métodos públicos para acessar e modificar os atributos.
  - Finalmente, faça um programa que:
    - Crie uma empresa;
    - Adicione a empresa alguns funcionários (solicitar no início quantos);