

Aula 7

Objectivo: Redefinição de operadores e herança

Problema 1

Altere o problema 1 da aula anterior, criando uma classe abstracta (sem representação interna para a DATA) com a seguinte definição:

```
typedef enum {Dom,Seg,Ter,Qua,Qui,Sex,Sab};

class DATA
{
public:
    virtual int dia(void) = 0;
    // mês
    // ano
    // dia da semana

    virtual const char *nome_do_mes(void); // Pode ter já uma implementação!
    // nome do dia da semana (pode ter também uma implementação)

    virtual void escreve(void);

    virtual void operator ++(int) = 0; // postfix
    // postfix operator --

    virtual void operator +=(int num_dias) = 0;
    // operator -=

    virtual int operator -(DATA &outra);
    virtual DATA &operator =(DATA &outra) = 0;
    // operator ==

    static int e_valida(int dia,int mes,int ano,int dia_da_semana);
    static int dias_do_mes(int mes,int ano);
    static int dias_do_ano(int ano);
    static int ano_bissesto(int ano);
};
```

1.1. Altere o nome da classe definida e implementada na aula anterior para DATA_DIA_MES_ANO, e faça com que ela seja descendente (especialização) da classe abstracta DATA.

Nota: Uma vez que o mecanismo de sobrecarga de métodos (*overloading*) e o de redefinição geram ambiguidades (não facilmente resolúveis em C++), mantenha, na classe descendente, exactamente a mesma assinatura dos operadores que tomam como parâmetros `DATAS`, e utilize *dynamic casts* para os converter para o tipo da classe descendente (`DATA_DIA_MES_ANO`).

1.2. Faça com que os programas do exercício da aula anterior continuem a funcionar com estas novas classes (minimizando as modificações).

1.3. Faça uma nova implementação para a classe abstracta `DATA`, usando como representação interna para a data o número de dias desde 1 de Janeiro de 2000 (naturalmente que o valor pode ser negativo). Chame a esta nova classe: `DATA_NUMERO_DIAS`.

1.4. Verifique se os programas continuam a funcionar se se passar objectos do tipo `DATA_NUMERO_DIAS` onde se esperam ponteiros para `DATA`.