VII. Const e Static

Constantes
Objetos constantes
Métodos estáticos
Atributos estáticos

Constantes

- Utilizam o modificador **const**.
- Constantes em C++ são variáveis que não podem ser modificadas.

```
const double PI = 3.1415;
PI = 2.1415;
```

Membros Constantes

```
class Teste
{
   public:
      const double PI = 3.1415;

};

int main()
{
      Teste t1;
      t1.PI = 2.1415;     //??
}
```

Objetos Constantes

- O qualificador **const** na declaração de um objeto indica que este é uma constante e nenhum de seus membros de dados podem ser alterados.
- Só o construtor é permitido.

```
const Data natal(25,12,2017);
natal.mostra(); //??
```

Objetos Constante

- Quando um objeto constante é declarado, o compilador proíbe a ele o acesso a qualquer método, pois não consegue identificar quais métodos alteram os seus dados.
- Colocar a palavra **const** após os parênteses que envolvem a lista de parâmetros libera o método para uso

```
void mostra() const {
    printf("\n%02d/%02d/%04d",dia,mes,ano);
}
```

ATRIBUTOS DE CLASSE

Membros Estáticos

- Atributos estáticos são usados quando é necessário que exista um dado que seja compartilhado por todos os objetos de uma mesma classe.
- Métodos estáticos agem independentemente de qualquer objeto declarado.
  - não podem acessar atributos não estáticos
- Utilizam o modificador **static**.

Atributo privado estático

```
class Cont
{
   static int n = 0; //??
   public:
   Cont() { n++; }
   int getN() { return n; }
};

int main()
{
   Cont c1,c2;
   cout << "\nNumero objetos " << c1.getN();
   cout << "\nNumero objetos " << c2.getN();
   Cont c3, c4;
   cout << "\nNumero objetos " << c3.getN();
}
```

Método estático

```
class Cont
{
   static int n;
   public:
   Cont() { n++; }
   static int getN() { return n; }
};

int Cont::n = 0;
int main()
{
   cout << "\nNumero objetos " << Cont::getN();
   Cont c1,c2;
   cout << "\nNumero objetos " << c1.getN();
   cout << "\nNumero objetos " << c2.getN();
   Cont c3, c4;
   cout << "\nNumero objetos " << c3.getN();
}
```

Atributo público estático

```
class Moeda {
   float CR;

   public:
   static float US;
   Moeda(float cr) { CR = cr; }
   float USdolar() { return CR/US; }
};
float Moeda::US;

int main() {
   Moeda a(5000);
   Moeda b(320);
   a.US = 3.12; // ou Moeda::US = 3.12;
   cout << "\n" << a.USdolar();
   cout << "\n" << b.USdolar();
}
```

Método estático

```
class Data
{
   int dia, mes, ano;

   public:
   static bool bissexto(int ano) {
      return ano%4==0 and ano%100 or ano%400==0;
   }
};


int main()
{
   cout << endl << Data::bissexto(1999);
   cout << endl << Data::bissexto(1900);
   cout << endl << Data::bissexto(2000);
}
```

EXERCÍCIOS

Exercícios

- Para a hierarquia de classes Animal, crie contadores para:
  - contar o número de objetos criados
  - contar o número de vezes que os animais se alimentaram

Exercícios

- Crie uma versão estática do método ValidaData(), na classe Data.

- Crie uma classe Conta, com nome do cliente e valor.
  - Com os métodos saldo(), deposito() e saque()
  - Crie uma variável estática para contar o número de contas ativas.