Aula 5

Objectivos: (Continuação da aula anterior.)
Construtores de cópia.

Preâmbulo:

A biblioteca gráfica `SIMPLE_GRAPHICAL_MODULE` permite a simulação de uma consola de texto (similar à existente com os operadores `cout` e `cin`, mas com os objectos `sgout` e `sgin`).
Para utilizar esta facilidade basta – após a inicialização da biblioteca (método `sgout.initialize()`), chamar o método: `sgout.enable_console()`.
É possível também definir a janela de acção da consola através do procedimento: `sgout.set_console_window(...)`.
Em qualquer altura pode-se, alternadamente, activar e desactivar (método `sgout.disable_console()`) a consola, tendo no entanto o cuidado de só utilizar os serviços de entrada e saída da consola com esta activa.
A interface da biblioteca – `simple_graphical_module.h` – contém uma descrição completa dos serviços existentes para interagir com a consola.
Chama-se a atenção que esta biblioteca memoriza todos os gráficos desenhados pelos métodos `add_*`, pelo que basta invocar estes serviços uma única vez. Após essa invocação, os gráficos passam a ter uma identificação única na biblioteca (`DRAWABLE_ID`) que pode ser recolhida pelo método: `last_added_drawable_id`, logo após a invocação do método respectivo.
Caso se queira apagar, esconder ou mostrar um qualquer gráfico já adicionado à biblioteca deve-se invocar, respectivamente, os métodos:
`remove_drawable`, `hide_drawable` e `show_drawable`.

Problema A1

Continuando os problemas A1 e A2 da aula anterior, pretende-se agora ir construindo um programa interactivo de desenho de figuras (do género do *Paint*).
Para tal, vamos começar por criar uma nova classe – `DESENHO` – com, pelo menos, a seguinte interface:

```
class DESENHO
{
public:

  void adiciona(FIGURA *fig);     // adiciona figura ao desenho (que é
                                  //   imediatamente desenhada)
                                  // esta figura passa a estar seleccionada

  void apaga(void);               // apaga a figura seleccionada no desenho

  void limpa(void);               // apaga o desenho todo

  void primeira(void);            // faz com que a figura seleccionada seja
                                  //   a primeira acrescentada ao desenho

  void seguinte(void);            // faz com que a figura seleccionada seja
                                  //   a que foi acrescentada após a actual

  FIGURA *actual(void);           // devolve a figura seleccionada da lista

  int figura_seleccionada(void); // diz se há alguma figura seleccionada
                                  //   ou não

  int e_primeira(void);           // diz se a figura seleccionada é a
                                  //   primeira acrescentada ao desenho

  int e_ultima(void);             // diz se a figura seleccionada é a
                                  //   última acrescentada ao desenho

};
```

(**Nota:** Aproveite à classe `LISTA_FIGURAS` já desenvolvida para implementar esta nova classe.)

Na interacção com o utilizador, no programa principal, passe agora a utilizar a consola existente na biblioteca gráfica (basta substituir, após a activação da consola, os objectos `cout` e `cin`, por `sgout` e `sgin`). Para que não haja interferência entre a consola e o desenho, pode acrescentar a possibilidade de activar e desactivar a consola.

A1.1. Para facilitar o programa de interacção com o utilizador, altere as classes das figuras por forma a permitir que estas sejam criadas sem nenhuma informação com o construtor sem argumentos (convém registar este estado, para evitar o desenho de figuras não definidas), e acrescente um método de entrada dos dados de cada figura a partir da consola:

```
void leitura(void);
```

Verifique em que sentido é que esta alteração simplifica a interacção com o utilizador.

A1.2. Acrescente o construtor de cópia em todas as figuras, por forma a permitir a sua replicação.

A1.3. Utilize os métodos `set_draw_bounding_box` e `reset_draw_bounding_box` para tornar claro ao utilizador qual o figura seleccionada do desenho (acrescente esse estado às classes das figuras, e altere apropriadamente os métodos de desenho).

A1.4. No programa principal faça com que se possa seleccionar figuras do desenho através das teclas: `PageUp` e `PageDown`.

A1.5. Acrescente a possibilidade de mover, no desenho, a figura seleccionada com as teclas das setas (caso exista).

A1.6. Acrescente a possibilidade de apagar, no desenho, a figura seleccionada.

A1.7. Acrescente a possibilidade de duplicar, no desenho, a figura seleccionada (utilize o construtor de cópia).

A1.8. Acrescente uma classe de figura composta. Esta nova figura será uma lista de outras quaisquer figuras (podendo incluir mesmo outras figuras compostas).
O centro desta figura será o centro de massa dos centros suas figuras.

Problema A2

Continuando o problema anterior analise e implemente uma forma (adequada) de definir as dimensões das figuras utilizando o rato (em vez da consola).

Problema B1

B1.1. Construa uma classe `CConjuntoInt` que guarda um conjunto de números inteiros (estes não podem repetir). Implemente as seguintes funções:

- `void Insert(int n);`

  para inserir um elemento novo no conjunto. Caso este elemento já exista, a função não faz nada; Inicialmente não se sabe quantos elementos vamos inserir.

- `bool Contains(int n);`

  para indicar se um dado elemento está no conjunto;

- `void Remove(int n);`

  para remover um elemento do conjunto. Caso este elemento não se encontre no conjunto, a função não faz nada;

- `void Empty();`

  para apagar todos os elementos do conjunto;

- `void Display();`

  para visualizar no écran os elementos do conjunto;

- `unsigned Size();`

  para calcular o número de elementos do conjunto.

B1.2. Adiciona à classe `CConjuntoInt` as funções seguintes:

- `CConjuntoInt Unir(const CConjuntoInt& add);`

  para construir um conjunto novo que representa a união do `this` com os elementos do conjunto representado pelo objecto `add`. O conjunto resultante não deve conter elementos repetidos.

- `CConjuntoInt Substrair(const CConjuntoInt& dif);`

  para construir um conjunto novo que representa a diferença do `this` e dos elementos do conjunto representado pelo objecto `dif`.

- `CConjuntoInt Interseccao(const CConjuntoInt& inter);`

  para construir um conjunto novo que representa a intersecção do `this` com os elementos do conjunto representado pelo objecto `inter`. O conjunto resultante não deve conter elementos repetidos.

B1.3. Teste a classe desenvolvida com a função `main` seguinte:

```
int main(int argc, char* argv[])
{
  CConjuntoInt c1;
  c1.Insert(4); c1.Insert(7); c1.Insert(6); c1.Insert(5);

  CConjuntoInt c2 = c1;
  c2.Insert(3); c2.Insert(2); c2.Insert(4); c2.Insert(7);
  c2.Remove(3); c2.Remove(5); c2.Remove(6);

  c1.Display(); c2.Display();

  cout << "Numero de elementos em c1: " << c1.Size() << endl;
  cout << "Numero de elementos em c2: " << c2.Size() << endl;

  cout << "c1 contem 6 ?: " << c1.Contains(6) << endl;
  cout << "c2 contem 6 ?: " << c2.Contains(6) << endl;

  cout << "Uniao:" << endl;
  c1.Display(); c2.Display();
  CConjuntoInt c3 = c1.Unir(c2); c3.Display();

  c2.Insert(3);
  cout << "Interseccao:" << endl;
  CConjuntoInt c4 = c1.Interseccao(c2); c4.Display();

  cout << "Diferenca:" << endl;
  CConjuntoInt c5 = c2.Substrair(c1); c5.Display();

  c1.Empty(); c1.Display();

  return 0;
}
```

Os resultados devem ser os seguintes:

```
Elementos do conjunto: 4 7 6 5
Elementos do conjunto: 4 7 2

Numero de elementos em c1: 4
Numero de elementos em c2: 3
c1 contem 6 ?: 1
c2 contem 6 ?: 0
Uniao:
Elementos do conjunto: 4 7 6 5

Elementos do conjunto: 4 7 2

Elementos do conjunto: 4 7 6 5 2

Interseccao:
Elementos do conjunto: 4 7

Diferenca:
Elementos do conjunto: 2 3

Elementos do conjunto:
```

Problema B2

Para cada uma das afirmações seguintes indique se é verdadeira ou falsa.

- Uma classe `livro` pode ser derivada da classe `biblioteca`.
- Uma classe `veículo` pode incluir objectos da classe `roda`.
- Quando criamos um objecto da classe derivada primeiro é chamado o construtor da classe derivada e, a seguir, o construtor da classe base.
- A definição da seguinte função `inc` é correcta:

```
int& inc (int v)
{
  v++;
  return v;
}
```