

**Programação de Computadores II - BCC702**  
**Lista de Exercício – Containers Associativos e POO**

**Exercício 01**

Defina uma classe *Data* com os atributos inteiros dia, mês e ano. Crie os métodos *getters* e *setters* para cada atributo e um construtor que recebe três inteiros como parâmetros para atribuir a dia, mês e ano, respectivamente.

Faça a sobrecarga do operador *>* para comparar se uma data é maior que outra.

**Exercício 02**

- a. Crie uma classe *Pessoa*, contendo os atributos encapsulados, com seus respectivos métodos *getters* e *setters*. Os atributos são: String nome; String endereço; Data dataNascimento;
- b. Considere, como subclasse da classe *Pessoa* (desenvolvida no item anterior) a classe *Fornecedor*. Considere que cada instância da classe *Fornecedor* tem, além dos atributos que caracterizam a classe *Pessoa*, os atributos *valorCredito* (correspondente ao crédito máximo atribuído ao fornecedor) e *valorDivida* (montante da dívida com o fornecedor).
- c. Implemente na classe *Fornecedor*, além dos usuais métodos *getters* e *setters*, um método obterSaldo() que devolve a diferença entre os valores dos atributos *valorCredito* e *valorDivida*.
- d. Crie, na função *main*, um *set* para armazenar os dados de vários fornecedores. Calcule e exiba:
  - i. o nome do fornecedor com maior dívida;
  - ii. os nomes dos fornecedores com saldo maior que um determinado valor (este valor deve ser lido);
  - iii. o nome do fornecedor mais novo.

### Exercício 03

Faça um programa que criptografa uma frase, alterando as vogais dela (desconsidere acentos). Para criptografar a frase, você deve criar um *map*, cuja chave é uma vogal e o valor correspondente é o símbolo de criptografia, de acordo com a tabela abaixo.

Chave	Registro
a	!
e	@
i	#
o	\$
u	%

Digite uma frase: Ja estou com saudades de prog2.

Frase criptografada: J! @st\$% c\$m s@%d!d@s d@ pr\$g2.

#### Dica:

Para pesquisar um elemento no mapeamento use a função membro *find*.

```
// Retorna um iterador para o elemento val se ele for encontrado em m
// Caso contrário retorna m.end()
map<string,int>::iterator itr = m.find(val);

// Exemplo:
// Se "Maps" não está presente no mapeamento como a chave de um valor
// então itr == m.end().
map<string,int>::iterator itr = m.find("Maps");
```