Programação Orientada a Objetos

EXERCÍCIO 39
Leia os capítulo 6 da apostila e resolva os exercícios.

EXERCÍCIO 40
Crie uma classe `Pessoa` que tenha os atributos nome, sobrenome e cpf. Faça um construtor `Pessoa(nome, sobreNome, cpf)`.

EXERCÍCIO 41
Crie uma classe `Disciplina` que tenha os atributos código (identificador único), nome da disciplina e o nome do professor. Crie os *getters* e *setters* necessários.

EXERCÍCIO 42
Crie uma classe `Aluno` que tenha os atributos: número de matrícula (identificador único), um objeto `Pessoa` (composição), uma lista de disciplinas (vetor) na qual ele está matriculado e data em que ocorreu a matrícula. Crie os métodos: `bool matricula(disciplina)` que matricula o aluno na disciplina; `bool tranca(disciplina)` que encerra a matrícula do aluno na disciplina; *getters* e *setters* necessários.

OBS: não permitir a matrícula do aluno na mesma disciplina mais de uma vez.

EXERCÍCIO 43
Considere um polinômio de grau n:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x^1 + a_0 x^0$$

Escreva uma classe `Termo` que represente um termo deste polinômio com os seguintes métodos:

`construtor`	Recebe dois parâmetros: a_i e i, e cria um objeto em memória na forma $a_i x^i$.
`insere`	Recebe um objeto da classe `Termo` e substitui os valores $a_i x^i$ do termo corrente por aqueles do termo recebido como parâmetro.
`calcula`	Recebe um valor de x como parâmetro e retorna o valor do termo calculado.

Escreva uma classe `Polinomio` que representa polinômio completo na forma de uma sequência de objetos da classe `Termo`, com os seguintes métodos:

`construtor`	Recebe um objeto da classe `Termo` e cria um polinômio em memória na forma: $P(x) = a_i x^i$
`insere`	Recebe um objeto da classe `Termo` e adiciona o termo $a_i x^i$ ao polinômio recebido como parâmetro. O polinômio pode ter um terno $a_q x^q$ cujo valor de q seja igual a i, neste caso a função deve unificar ambos em um único termo.
`calcula`	Recebe um valor de x como parâmetro e retorna o valor de $P_{(x)}$.
`fusao`	Recebe como parâmetro outro objeto da classe `Polinomio` e realiza a fusão do polinômio recebido como parâmetro com o polinômio corrente.

Acrescente os métodos que forem necessários nas classes solicitadas.

Exercício inspirado em exemplo dos slides de prof. Tomasz Kowaltowski

EXERCÍCIO 44
Um protozoário é um ser vivo simples, porém completo. A partir do seu nascimento, cada protozoário tem um número serial único, conhecido como individualidade, que nunca pode mudar e nunca poderá ser associado a outro protozoário. O genótipo de um protozoário é composto por 10 genes, sendo que cada um pode ter valores de 0 a 3. Um protozoário pode se reproduzir por replicação ou por cópula com outro protozoário*. Um protozoário nascido através da replicação de outro é uma cópia idêntica do pai, exceto por sua individualidade. Já se for gerado por cópula, receberá os genes randomicamente de cada indivíduo, com uma chance de 50% do gene vir de um ou de outro. De vez em quando (em 7% dos casos), pode haver mutação no genótipo durante a reprodução, que altera um único gene no protozoário gerado.

Modele a classe `Protozoario`, que deve possuir os seguintes métodos:

1. `Protozoario()`: cria um protozoário com o genótipo aleatório. Por exemplo: [1,0,3,2,0,2,1,0,3,1];
2. `Protozoario(int[] genotipo)`: cria um protozoário com o genótipo fornecido;
3. `void mutate()`: causa uma mutação no genótipo;
4. `Protozoario getClone()`: retorna um novo protozoário criado por replicação;
5. `Protozoario mate(Protozoario outro)`: cria um filho;
6. `String toString()`: retorna uma representação em string do protozoário.

Crie uma classe de teste e faça algumas simulações com os protozoários (criar, realizar mutações, cruzamentos entre indivíduos, etc)

* OBS: os protozoários não se reproduzem por cópula, somente por divisão simples (replicação).

EXERCÍCIO 45
Escreva a classe `SerieLimitada` que representa objetos que encapsulam um valor de série como os usados em tiragem de notas, gravuras, ingressos, etc. Essa classe deve permitir que um programa crie um número limitado de instâncias dela, cada uma numerada com um valor diferente. O número total de instâncias é controlado pelo campo máximoDeInstâncias e o de instâncias já criadas é controlado pelo campo contadorDeInstâncias. Apesar do valor da série ser gerado automaticamente, tenha cuidado de não permitir criação de dois valores de série iguais (é improvável, mas pode acontecer). Crie o método getValor que retorna o valor da série armazenada no objeto.

O número de série tem 10 dígitos. Por exemplo: 0357415652

BOM ESTUDO!