

Paradigma e Linguagem Orientada a Objetos

Prof. Fabio T. Matsunaga

Ciência da Computação

Universidade Estadual do Paraná (UNESPAR)

Campus Apucarana

Trabalho do Bimestre

Faça as seguintes atividades considerando os conceitos de orientação a objetos visto em aula. O trabalho pode ser desenvolvido em equipes. Por isso, organizem-se em equipes de 3-4 pessoas. **Entregar até o dia 19/07.** Utilize a linguagem C++ para construção de classes e objetos.

1. Escreva uma classe em Java chamada Contador, que contenha um atributo de valor inteiro. A classe deve encapsular esse valor contendo os seguintes métodos:
 - Construtores sobrecarregados que inicializem um valor de diferentes formas:
 - Inicialize com zero (construtor sem parâmetro);
 - Inicialize com um valor (construtor com um parâmetro);
 - Zerar;
 - Incrementar;
 - Decrementar;
 - Retornar o valor atual do contador;
2. Crie uma classe denominada Hora que representara uma hora com o formato padrão. A classe contém três atributos: horas, minutos e segundos. Essa classe também conterá os métodos:
 - Construtores sobrecarregados:
 - Um construtor sem parâmetros, que inicialize a hora com 00 : 00 : 00;
 - Um construtor com três parâmetros, que inicialize as horas, minutos e segundos com os valores passados;
 - Em cada construtor acima, fazer um algoritmo de validação da hora, para verificar se o usuário inseriu um formato e valores de hora válido;
 - Getters e setters;
 - Método para retornar a quantidade total da hora em segundos;
 - Método para retornar o horário em um formato string hh:mm:ss;
 - Implemente um método de sobrecarga de operadores com o operador (–) de modo que seja possível de subtrair dois objetos do tipo Hora. Nessa subtração, considere as regras padrões de calcular a diferença de uma hora com outra;

3. Implemente uma classe em Java denominada Pessoa. Essa classe pessoa deve conter os seguintes atributos: nome, CPF e data de nascimento, além de ter um construtor que inicialize esses três atributos. Em seguida, faça uma herança a partir da classe Pessoa mais duas subclasses: Professor, Aluno. As subclasses, além de ter os métodos e atributos da classe Pessoa, deve conter alguns atributos particulares. Professor por exemplo pode possuir um salário e um nome de escola. Já o aluno possui um número de matrícula, uma escola e a série.
4. Escreva uma classe em Java chamada `Reta2D`, que represente uma reta ($y = ax + b$) no plano Cartesiano. Os atributos da classe devem ser o coeficiente angular a e o coeficiente linear da reta b . Em seguida, forneça os seguintes métodos da classe:
 - Construtores sobrecarregados que inicialize a equação da reta de acordo com os seguintes valores recebidos:
 - Dois parâmetros representando o coeficiente angular a e o coeficiente linear da reta b ;
 - Getters para o coeficiente angular e linear da reta;
 - Método que verifique se um ponto pertence à reta;
 - Método que gere uma String da equação no formato $y = ax + b$;
 - Método que dado um valor de x , retorne o valor de y ;
5. Escreva uma classe em Java chamada `Circunferencia2D`, que represente uma circunferência no plano Cartesiano. O atributo da classe deve ser o raio e o centro (x, y) da circunferência. Forneça os seguintes métodos da classe:
 - Construtores sobrecarregados que recebam os seguintes valores:
 - Valor do raio e um ponto (x, y) , posicionando o centro da circunferência no ponto (x, y)
 - Somente o valor do raio, posicionando o centro da circunferência na origem $(0, 0)$.
 - Método de acesso ao valor do raio da circunferência (getter);
 - Método de acesso ao centro do círculo (getter);
 - Métodos de aumentar e diminuir o círculo de acordo com um valor de raio fornecido como parâmetro;
 - Método para retornar a área do círculo $A = \pi r^2$;
 - Método para retornar o comprimento da circunferência ($P = 2\pi r$);
 - Método para mover a circunferência para o local indicado por um outro ponto;
6. Escreva uma classe que representa um país. Um país é representado pelos atributos: nome, população e dimensão em Km^2 . Além disso, cada país mantém uma lista de outros países com os quais ele faz fronteira. Considerando-se isso, escreva a classe em C++ para representar o país e seus membros:

- Construtor que inicialize os atributos da classe;
- Métodos de acesso (getter e setter) de cada atributo;
- Um método que retorne a densidade populacional do país (habitantes por Km^2);
- Um método que liste todos os vizinhos do país;

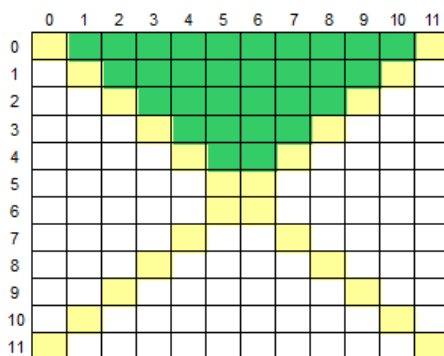
Na mesma atividade crie uma classe Continente, que possui um nome e é composto de um conjunto de países, que este contém construtor que inicializa o nome do continente. Além disso, possui outros métodos:

- Um método que adicione objetos de países na classe Continente;
- Um método que retorne a dimensão total do continente;
- Um método que retorne a população total do continente;
- Um método que retorne o país com a maior população do continente;
- Um método que retorne o país com a menor população do continente;
- Um método que retorne o país com a maior dimensão territorial do continente;
- Um método que retorne o país com a menor dimensão territorial do continente;

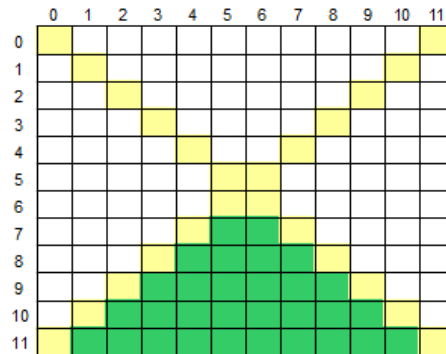
7. Crie uma classe denominada Vetor que representará um vetor de inteiros e suas propriedades. A classe deve conter dois atributos: um vetor de inteiros e seu tamanho. Essa classe conterá os métodos:

- Construtores sobrecarregados:
 - Um construtor com um parâmetro que representa o tamanho inicial do vetor. Para esta operação, realize a alocação dinâmica, isto é, a alocação em tempo de execução do espaço de memória reservado para o vetor.
- Getters e setters;
- Método para inserir um número no vetor (na ordem em que foi inserida);
- Método para calcular o somatório do vetor e retornar o valor;
- Método para inverter o vetor;
- Método para imprimir os valores do vetor;
- Implemente uma sobrecarga de operadores com as operações $+$, $-$ e $*$ de modo que possa somar, subtrair e multiplicar dois vetores, considerando que todos têm a mesma quantidade de elementos;
- Implemente uma sobrecarga de operadores com as operações $>$ e $<$ de modo que possa comparar dois vetores. Considere a somatória dos vetores como parâmetro de comparação.
- Pesquise dois métodos de ordenação de vetores e implemente esses métodos na sua classe, de modo que ordene o vetor em ordem crescente;

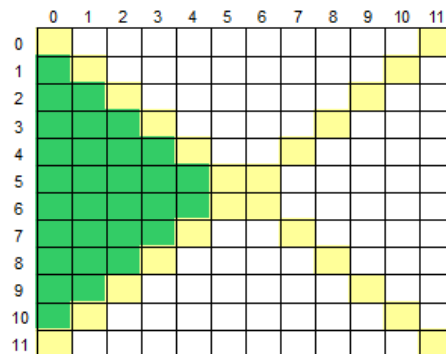
- Faça um método de ordenação em ordem decrescente sem reimplementar o método de ordenação. Apenas aproveite os métodos já existentes;
 - Um destrutor para liberar o espaço de memória alocado;
8. Crie uma classe denominada Matriz que representa uma matriz quadrada bidimensional de números reais e suas propriedades. A classe contém atributos de uma matriz de ponto flutuante e mais um valor n representando a dimensão (dimensão $n \times n$). A classe conterá os seguintes métodos:
- Um construtor que receba a dimensão da matriz e aloque dinamicamente um espaço de memória para a matriz.
 - Um método que receba um valor de uma linha e retorne em forma de um vetor o conteúdo dessa linha;
 - Um método que receba um valor de uma coluna e retorne em forma de um vetor o conteúdo dessa coluna;
 - Um método para retornar a diagonal primária em forma de um vetor;
 - Um mesmo método como acima para retornar a diagonal secundária;
 - Um método para retornar a matriz transposta (as linhas se transformam em colunas e as colunas em linhas);
 - Um método para informar se a matriz é simétrica, isto é, se uma matriz é igual a sua transposta (retorna 0 ou 1);
 - Um método para informar se a matriz é uma matriz identidade, isto é, quando os elementos da diagonal principal são todos iguais a 1 e os elementos restantes são iguais a zero (retorna 0 ou 1);
 - Um método que receba como parâmetros as seguintes opções e calcule a soma dos elementos conforme a área destacada de cada matriz:



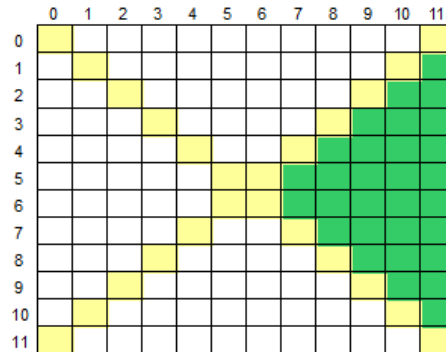
– Opção 1:



– Opção 2:



– Opção 3:



– Opção 4:

- Implemente uma sobrecarga de operadores com as operações $+$ e $-$ para que possa somar e subtrair dois objetos do tipo Matriz;
- Implemente uma sobrecarga de operadores com a operação $==$ para que possa comparar duas matrizes. Duas matrizes serão iguais se todos os elementos forem iguais;
- Um destrutor para liberar o espaço de memória alocado;