LabXI: Banco - Herança

Para esse exercício suponha um pequeno Banco que possui 4 Clientes. Cada cliente possui uma única Conta associada com número e saldo. Essa Conta pode ser Corrente ou Poupança. A cada mês a Conta deve atualizar os saldos [se a conta for Corrente é debitado 1 real do saldo para manutenção e se for Poupança é acrescido 1% no saldo]. Modele a estrutura e implemente estas classes.

Regras

1. O Cliente (nome, Conta (número, saldo)) se cadastre no Banco fornecendo um saldo inicial.

2. Se possa depositar, retirar dinheiro e consultar saldo de uma conta pelo seu número ou pelo nome do cliente.

3. Se possa aplicar a atualização mensal quando necessário.

4. Se exiba todo o valor guardado no Banco.

5. Compile os fontes.

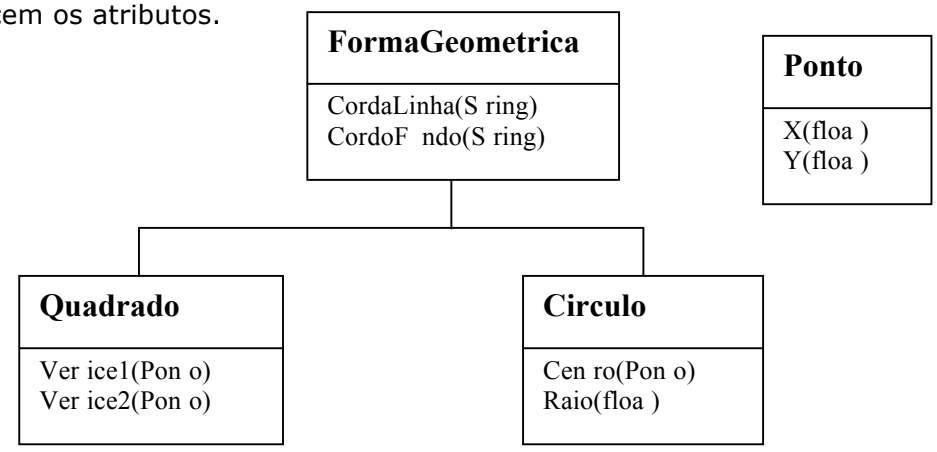
Demonstração

Crie um programa de exemplo que efetue as seguintes operações:

- Cadastre com saldo inicial de 100 reais cada, 3 clientes com conta poupança, e 1 clientes com conta corrente;
- Saque 30 pelo número da conta na 1ª conta corrente criada;
- Deposite 34 reais pelo nome do cliente na 1ª conta poupança criada;
- Atualize os valores devido a virada de mês
- Exiba pelo número da conta o saldo da 1ª conta-corrente
- Exiba o valor guardado no Banco.

LabXII: Formas Geométricas - Herança

A figura abaixo representa um diagrama simplificado de classes onde só aparecem os atributos.



Implemente as classes acima usando a linguagem Java, respeitando o fato de que os métodos deverão ser implementados de maneira a atender os requisitos abaixo relacionados:

Regras

1. Cada classe deve ter dois construtores. Um com a assinatura "default" e outro que recebe como argumento os atributos da classe.
2. Todos os atributos devem ter métodos de acesso (tanto para definição quanto para recuperação) aos seus valores.
3. As classes circulo e quadrado devem possuir um método para calcular a área da respectiva forma geométrica e outro para imprimir o tipo da forma geométrica(Nome da Classe), os valores de seus atributos e a área.

Demonstração

Crie um programa de exemplo que efetue as seguintes operações:
• Crie um Quadrado e um Círculo usando variáveis do tipo FormaGeométrica.
• Exiba o resultado do método imprime de cada Forma Geométrica Criada.

LabXIII: Forno - Agregação

Escreva um programa em Java que simule um forno, um termostato e um usuário. Primeiro o termostato é configurado pelo usuário que define a temperatura chamada de inicial. Depois o usuário passa a temperatura do forno para o termostato. Então esse compara as duas temperaturas e caso a temperatura do forno estiver abaixo da inicial o termostato avisa que o forno deve ser ligado. Quando o forno é ligado ou desligado um aviso é emitido. O programa termina aqui. Dica: Crie classes separadas para cada uma das entidades do programa(Usuário, Forno, Termostato).

Demonstração

Os valores das temperaturas necessárias para a realização do programa devem ser definidos por você no método main do programa. Crie uma classe chamada de Teste que tem o método main para testar o programa.

LabXIV: PARXY – Comparação de Objetos

Defina, desenvolva e teste uma classe PARXY, cujas instâncias são pares (x,y) de dois objetos de qualquer classe.Cada instância desta classe deverá ser capaz de responder às seguintes mensagens:
• getX() / putX(obj) – devolve / define o objeto em x;
• getY() / putY(obj) – devolve / define o objeto em y;
• iguaisXY() – indica se os objetos X e Y "são o mesmo" objeto.

Demonstração

Crie uma instância de PARXY e use todos os seus métodos para testar a sua implementação.

LabXV: PONTO3D

Desenvolva a classe PONTO3D, cujas instâncias são pontos de três coordenadas inteiras (x,y,z). Cada instância desta classe deverá ser capaz de responder às seguintes mensagens:
• putX(int), putY(int), putZ(int) – altera as coordenadas do ponto;
• getX(),getY(),getZ()– devolve as coordenadas do ponto.

LabXVI: PLANO3D

Desenvolva a classe PLANO3D, cujas instâncias são planos formados a partir de três PONTO3D não alinhados (i.e. não pertencem à mesma reta). Cada instância desta classe deverá ser capaz de responder às seguintes mensagens:

• horizontal() – verdadeiro se o plano for horizontal;
• perpendicular() – verdadeiro se o plano for perpendicular a qualquer dos eixos(X ou Y).

Demonstração

Crie uma instância de PLANO3D e use todos os seus métodos para testar a sua implementação.

LabXVII: RETÂNGULO

Defina, desenvolva e teste uma classe Retangulo, cujas instâncias apresentem um comportamento semelhante ao de um Retângulo cujas dimensões (comprimento e largura) são valores inteiros. Cada instância desta classe, deverá ser capaz de responder às seguintes mensagens:
• getComprimento () / setComprimento(comp) – devolve / define o comprimento;
• getLargura() / setLargura(l) – devolve / define a largura;
• getArea(c,l) – calcula a área do retângulo;
• perimetro() – calcula o perímetro do retângulo;
• divide() – o retângulo é dividido e é retornado o retângulo(uma nova instância) resultante da divisão;

Demonstração

Crie uma instância de Retangulo e use todos os seus métodos para testar a sua implementação.

LabXVIII: Lista de Objetos

Defina e desenvolva uma classe Lista, cujas instâncias são listas não limitadas de objetos de qualquer tipo. Cada instância desta classe deverá ser capaz de responder às seguintes mensagens:

- putObj(obj) – coloca obj numa posição livre da lista e retorna o índice dessa posição;
- remObj(indice) – remove o obj que está na posição índice;
- posObj(obj) – determina o índice na lista de obj (0 caso não exista);
- temObj(obj) – determina se contém numa posição da lista, o objeto obj;
- isEmpty(ind) – determina se a lista está vazia na posição ind (null);
- isEmpty() – determina se a lista está vazia (se apenas contiver objetos null);
- tamanho() – determina o número de objetos não null que a lista contém.

Demonstração

Crie uma instância de Lista e use todos os seus métodos para testar a sua implementação.

LabXX: Tratamento de Exceções

Construa a classe TestaStack que, usando uma instância de Stack (veja documentação anexada), deve efetuar a seguinte seqüência de ações de modo a que o programa não termine (tratar exceção): push(3), push(4), pop(), pop(), pop(), push(5).

LabXXI - A: Tratamento de Exceções

Construa a classe TestaStack que, usando uma instância de Stack (veja documentação do pacote java.util), deve efetuar a seguinte seqüência de ações de modo a que o programa não termine (tratar exceção): push(3), push(4), pop(), pop(), pop(), push(5).

LabXXI - B: Tratamento de Exceções

Construa e teste a classe Queue (fila de tamanho fixo que ao contrário da stack, tem um comportamento FIFO (FirstIn-FirstOut).

LabXXI - C: Tratamento de Exceções

Construa a classe StackInt (subclasse de Stack) que apenas permite que os seus elementos sejam inteiros,não permitindo repetições.

LabXXI - D: Tratamento de Exceções

Pretende-se criar a classe Console que através de métodos estáticos permita fazer a leitura validada de tipos primitivos. Esta classe poderá vir a ser utilizada em problemas futuros.

LabXXVIII

Faça um programa orientado a objetos para cadastrar pessoas que se inscreveram em um Projeto de Pesquisa que tem como pesquisador maior a Sra. Mara Andrade. Os pesquisadores que trabalham neste Projeto de Pesquisa podem ser Coordenadores de Pesquisa que têm pesquisadores e outros coordenadores de pesquisa sob sua supervisão ou então simples professores.

Uma classe Pesquisador deve ser criada para auxiliar a sua solução. Todo pesquisador deve ser capaz de informar quantos pesquisadores existem sobre sua supervisão. Todo pesquisador possui um coordenador de pesquisa, que também é do tipo pesquisador.

A classe Projeto de Pesquisa deve ser construída para armazenar os pesquisadores cadastrados (método inserirPesquisador), para retornar a quantidade de pesquisadores associados a um pesquisador (método getQtdPesquisadores) e para retornar o valor a ser pago pelo pesquisador para se inscrever no Projeto (método getValorPago).

Para calcular o valor a ser pago pelo pesquisador é preciso levar em consideração que professores pagam R$ 220,00 mais 30% do que paga seu coordenador de pesquisa, enquanto que coordenadores de pesquisa pagam R$ 300,00 menos 2,5% por cada pesquisador que esteja vinculado a ele