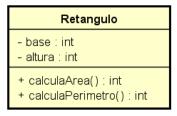
POO - Lista de Exercícios 2 - Prof. Adriano Nakamura

1. Defina uma classe de acordo com a figura a seguir e implemente os métodos:



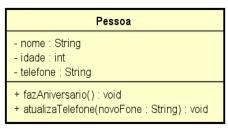
- método calculaArea deve retornar a área do retângulo, que é calculada multiplicando-se a base pela altura.
- método calculaPerimetro deve retornar o perímetro do retângulo, que é calculado somando-se o valor dos quatro lados do mesmo.

powered by Astah

Defina uma segunda classe chamada **Principal1** onde deverá ser implementado o método **main**. Dentro deste método declare e crie o objeto **ret** da classe **Retangulo**; atribua à base o valor 5 e à altura o valor 3 e gere a seguinte saída:

```
A base do retangulo e 5.
A altura do retangulo e 3.
A area do retangulo e 15.
O perimetro do retangulo e 16.
```

2. Defina uma classe de acordo com a figura a seguir e implemente os métodos:



- Método **fazAniversário** deve incrementar o atributo **idade** em uma unidade;
- Médoto **atualizaTelefone** deve atualizar o atributo **telefone** com o valor recebido por parâmetro

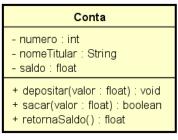
powered by Astah

Defina uma segunda classe chamada **Principal** onde deverá ser implementado o método **main**. Dentro deste método declare e crie o objeto **p** da classe **Pessoa**, faça a leitura dos dados (nome, idade, telefone), exiba todos, faça a chamada do método **fazAniversário**, atualize o telefone para "3344-5555" (faça a chamada do método **atualiza Telefone**) e exiba novamente a idade e o telefone. A saída do seu programa deverá ser:

```
Nome: <valor do atributo nome>.

Idade: <valor do atributo idade>
Telefone: <valor do atributo telefone>
Idade depois do aniversário: <valor do atributo idade>
Novo telefone: <valor do atributo telefone>
```

3. Defina uma classe de acordo com a figura a seguir e implemente os métodos:



- método depositar deverá receber como parâmetro o valor do depósito que deverá ser acrescentado ao saldo
- método sacar deverá receber como parâmetro o valor da retirada e se o saldo for suficiente o valor da retirada deverá ser subtraído e o método deve retornar *true*, do contrário o método deve retornar false.
- método retornaSaldo deverá retornar o valor do saldo.

powered by Astah

Defina uma segunda classe chamada **Principal** onde deverá ser implementado o método **main**. Dentro deste método declare e crie o objeto **cliente** da classe **Conta**, crie um "loop" exibindo o menu a seguir e programe as opções.

CONTA CORRENTE

1 - Depósito

2 - Saque

3 - Consulta Saldo

4 - Finalizar

Opção:

Obs.: para as operações de depósito e saque deverá ser solicitado o valor da operação antes de evocar o respectivo método. Para o saque deverá ser testado se a operação foi realizado com sucesso (true) ou não (false) neste caso exiba a mensagem "Saldo insuficiente!". A opção 4 deverá encerrar o "loop"

4. Defina uma classe de acordo com a figura a seguir e implemente os métodos:

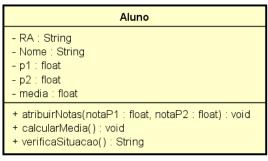
Produto - descrição : String - preço : float - qtde_disp : int - qtde_vend : int + repor(qtde : int) : void + vender(qtde : int) : boolean + exibirDados() : void

powered by Astah

- método construtor deve receber por parâmetro os dados para inicializar os atributos descrição, preço;
- método repor deve receber por parâmetro a qtde a ser reposta no estoque e somar à qtde disponível;
 - método vender deve receber por parâmetro a qtde a ser vendida, caso esta seja o possível (qtde_disp >=qtde) subtrair da qtde disponível e retornar *true* do constrário retornar *false*;
 - método **exibirDados** deve exibir todos os atributos (colocar uma legenda antes dos dados

Defina uma segunda classe chamada **Principal1** onde deverá ser implementado o método main. Dentro deste método declare e crie um objeto da classe **Produto**, reponha o estoque e faça uma venda solicitando as devidas quantidades ao usuário e exiba os dados. Obs: para a venda verifique se a operação foi realizada e exiba as mensagens "Venda realizada!" ou "Quantidade insuficiente!".

5. Defina uma classe de acordo com a figura a seguir e implemente os métodos:



powered by Astah

- método construtor deve receber por parâmetro os dados para inicializar os atributos RA e nome;
- método atribuirNotas deve receber por parâmetro nas notas da P1 e P2 e inicializar os respectivos atributos;
- método calcular Média deve calcular a média e retornar o seu valor;
- método **verificaSituacao** deve retornar a situação do aluno:
 - "Aprovado" se média >=7
 - "Exame" se média <7 e media >=4
 - "Reprovado" se média <4

Defina uma segunda classe chamada **Principal2** onde deverá ser implementado o método main. Dentro deste método declare e crie um objeto da classe **Aluno**, faça a atribuição das notas exiba a sua média e sua situação.