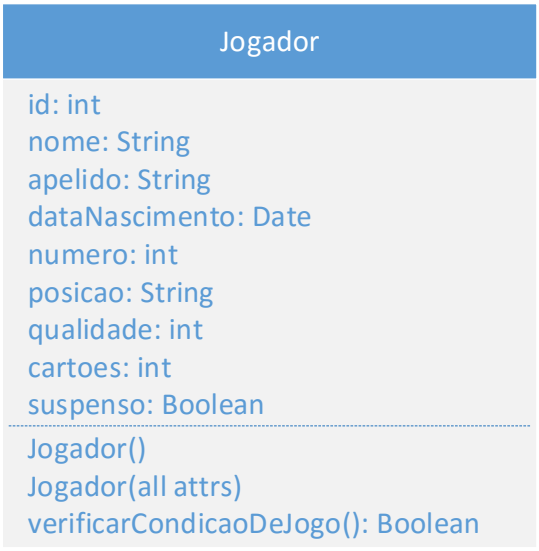
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SUL-RIO-GRANDENSE	**Curso Técnico em Informática**
Nome:	Data:
Disciplina de Linguagem de Programação I	Professor: Ricardo Luis dos Santos

Exercícios sobre Orientação a Objetos

1) Crie a classe Jogador descrita no “diagrama” abaixo:



a) Cadastrar/Instanciar pelo menos um time completo 11 jogadores.

b) Criar um método que verifica a condição de jogo, ou seja, um método booleano que retornará ***true*** se o jogador está apto a jogar e ***false*** se o jogador está suspenso. Note que um jogador está suspenso pelo 3 cartão amarelo ou quando recebe uma punição cartão vermelho ou em uma decisão do tribunal.

c) Em uma outra classe, crie o método ***main***, o qual cadastra os jogadores e ao final imprimirá a lista do “plantel” juntamente com a informação de quem está apto a jogar, conforme a figura abaixo.

```
<terminated> Menu (6) [Java Application] C:\Program Files\Java\jdk1.8.0_60\bin\javaw.exe (29 de abr de 2016 00:15:48)
JOGADORES CADASTRADOS
Goleiro: 1 - Marcelo Grohe (Muralha Tricolor) - 13/jan/1987 CONDIÇÃO: SUSPENSO
Lateral Direito: 2 - Marcelo Hermes (Hermes) - 01/fev/1995 CONDIÇÃO: TÁ PRA JOGO
```

d) Crie novos métodos na classe Jogador:

- ***aplicarCartao(int quantidade): void*** - Aplica a quantidade de cartões informada ao jogador, adicionalmente pode tornar um jogador suspenso.
- ***cumprirSuspencao(): void*** – Esse método vai zerar a quantidade de cartões e tornar o jogador apto a jogar
- ***sofrerLesao(): void –*** Este método vai gerar aleatoriamente um lesão no jogador. A gravidade da lesão irá se refletir em uma redução da qualidade do jogador, quanto mais grave maior a redução da qualidade. Crie uma escala de redução de no mínimo 1 ponto até o máximo de 15% da qualidade

total do jogador. Note que a qualidade jamais pode ficar negativa. A tabela abaixo pode ser utilizada como referência:

Probabilidade	Qualidade decrementada
5%	15% do total da qualidade
10%	10% do total da qualidade
15%	5% do total da qualidade
30%	2 pontos
40%	1 ponto

- ***executarTreinamento(): void –*** A exemplo do método anterior, este método aleatoriamente vai aumentar a qualidade do jogador. Note que só pode ser executado 1 treinamento antes de cada partida (você deve adicionar um atributo na classe para poder controlar essa informação). Além disso, deve existir uma maior probabilidade para pequenos incrementos na qualidade conforme a seguinte tabela:

Probabilidade	Qualidade incrementada
5%	5 pontos
10%	4 pontos
15%	3 pontos
30%	2 pontos
40%	1 ponto

Observe que a qualidade nunca pode superar o máximo de 100 pontos. Além disso, você deverá adicionar mais um atributo na classe jogador para poder controlar os jogadores que já efetuaram o treinamento e que só poderão treinar após o próximo jogo.

2) Crie uma nova classe chamada Time conforme abaixo:

a) Cadastrar/Instanciar pelo menos 2 times e em cada time pelo menos 23 jogadores.

b) Observe que o método ***relacionarJogadores*** deve ser criado. Este método irá selecionar os 11 melhores jogadores de acordo com a qualidade (os 11 jogadores com a maior qualidade devem ser relacionados).

c) Em uma outra classe, crie o método ***main***, o qual cadastra os times e os jogadores e ao final imprimirá as escalações de ambos os times.

DESAFIOS:

a) Altere o método ***relacionarJogadores*** para selecionar 18 jogadores, 11 titulares e 7 reservas ordenados pela qualidade, ou seja, os 11 titulares devem estar alocados nas primeiras posições do ***array*** e os 7 reservas nas últimas posições do ***array***.

b) Altere o método ***relacionarJogadores*** para considerar não apenas a qualidade dos jogadores, mas também a sua posição, assim os 18 relacionados devem ser os melhores em sua posição, mas não necessariamente os jogadores com as maiores qualidades no elenco.

c) Em uma outra classe, crie o método ***main***, o qual cadastra os times e os jogadores e ao final imprimirá as escalações de ambos os times, porém, agora também devem ser informados os jogadores titulares e os jogadores que compõem o banco de reservas.

3) Crie a classe conforme o diagrama abaixo:

a) O método ***gerarResultado*** deve criar um resultado aleatório considerando o total da qualidade dos titulares. Isso quer dizer que quanto maior o somatório da qualidade dos jogadores titulares maior será a probabilidade da equipe ganhar. Por exemplo, o somatório da qualidade dos jogadores titulares da equipe A é 584, enquanto que o somatório da qualidade dos jogadores titulares da equipe B é 723. Considere esses valores no momento de definir o resultado e o ganhador do confronto. Pode utilizar tanto uma clusterização, por exemplo, 50% de vitória para o time com maior qualidade, 30% de vitória para o time com menor qualidade e 20% para dar empate.

b) O método ***gerarLesoes*** deve gerar aleatoriamente de 0 até duas lesões (independente do time) por jogo. O jogador que sofre a lesão também deve ser escolhido aleatoriamente.

c) O método ***gerarCartoes*** deve gerar aleatoriamente de 0 até 10 cartões por jogo (independente do time). O jogador que recebe o cartão deve ser escolhido aleatoriamente.

d) O método ***permitirTreinamento*** após a conclusão do jogo deve possibilitar que os jogadores possam realizar outro treinamento.

DESAFIOS:

a) No método ***gerarResultado*** considere a qualidade total dos jogadores de uma forma a não clusterizar as probabilidades.

b) No método ***gerarCartoes*** verifique se o mesmo jogador é escolhido duas vezes, neste caso, ele deve ser expulso do jogo alterando as probabilidades de vitória de cada time. Além disso, ele deve ficar suspenso da próxima partida.

4) Crie uma classe Funcionário com atributos que permitam mapear o nome do funcionário, a matrícula, o salário, a data de admissão e o seu CPF.

a) A classe deve conter um método ***receberAumento***, o qual aumenta o salário do funcionário de acordo com o parâmetro informado (double), sem retornar nenhum valor.

b) A classe deve conter um método ***calcularGanhoBrutoAnual*** que não recebe nenhum parâmetro e retorna apenas o total bruto anual recebido pelo funcionário.

c) Crie um método ***calcularImposto***, o qual não recebe nenhum parâmetro e retorno o imposto pago pelo funcionário no decorrer de um ano (double).

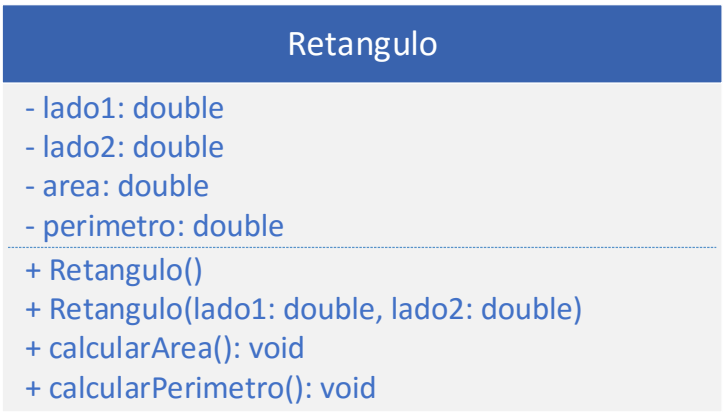
d) Crie um método ***calcularGanhoLiquidoMensal*** que não recebe nenhum parâmetro e retorna o salário líquido do funcionário, sendo que existem os seguintes descontos:
   - O funcionário paga 11% de INSS
   - Se o salário é maior que R$ 2.500,00 é cobrado 17,5% de IR do que exceder os R$ 2.500,00.

e) Crie um método ***calcularGanhoLiquidoAnual*** que não recebe nenhum parâmetro e retorna o total líquido recebido pelo funcionário, ou seja, o bruto reduzido dos impostos devidos.

f) Crie a representação dessa classe (atributos e métodos) de acordo com a notação UML.

LISTA II

1) Crie uma classe **Retangulo** que obedeça à descrição abaixo:

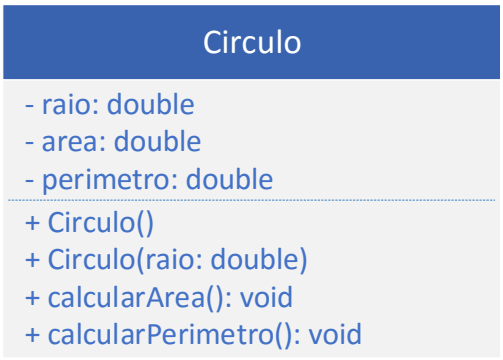


- A classe possui os atributos lado1, lado2, area e perimetro, todos do tipo float.
- O método calcularArea deve realizar o cálculo da área do retângulo (area = lado1 * lado2). Em seguida, deve escreve o valor da área na tela.
- O método calcularPerimetro faz o cálculo do perimetro (perimetro = 2*lado1 + 2*lado2). Em seguida, deve escreve o valor do perímetro na tela.

Crie o método main e instancie a classe Retangulo, criando um objeto novoRetangulo do tipo Retangulo.

- Atribua o valor 10 ao atributo lado1.
- Atribua o valor 5 ao atributo lado2.
- Chame o método calcularArea.
- Chame o método calcularPerimetro.
- Atribua o valor 7 ao atributo lado2.
- Chame o método calcularArea.
- Chame o método calcularPerimetro.
- Crie outras 5 instancias de retângulos, conforme as instruções anteriores.

2) Crie uma classe **Circulo** que obedeça à descrição abaixo:

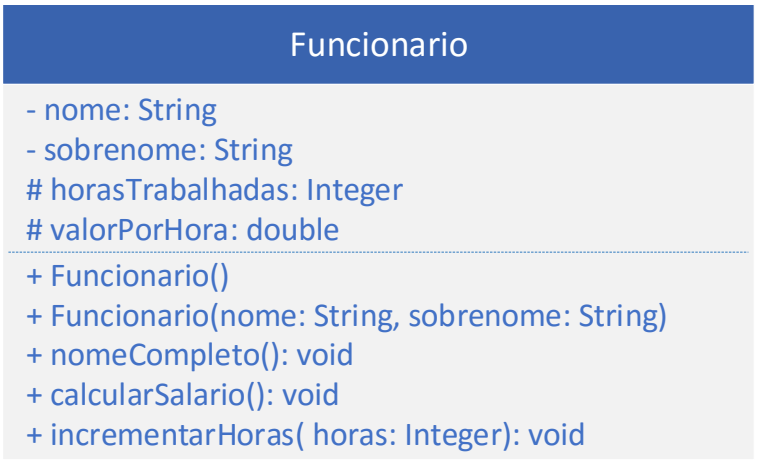


- A classe possui os atributos raio, area e perímetro, todos do tipo float.
- O método calcularArea deve realizar o cálculo da área do retângulo (area = raio * raio * 3.14). Em seguida, deve escreve o valor da area na tela.
- O método calcularPerimetro faz o cálculo do perimetro (perimetro = 2 * 3.14 * raio). Em seguida, deve escreve o valor do perímetro na tela.

Crie o método main e instancie a classe Circulo, criando um objeto novoCirculo do tipo Circulo.

- Atribua o valor 10 ao atributo raio.
- Chame o método calcularArea.
- Escreva na tela o valor da área.
- Chame o método calcularPerimetro.
- Escreva na tela o valor do perímetro.
- Atribua o valor 4 ao atributo raio.
- Chame o método calcularArea.
- Escreva na tela o valor da área.
- Chame o método calcularPerimetro.
- Escreva na tela o valor do perímetro.
- Instancie outros 5 círculos conforme instruções anteriores.

3) Crie uma classe **Funcionario** que obedeça à descrição abaixo:



- A classe possui os atributos nome, sobrenome, horasTrabalhadas e valorPorHora.
- O método nomeCompleto deve escrever na tela o atributo nome concatenado ao atributo sobrenome.
- O método calcularSalario faz o cálculo de quanto o funcionário irá receber no mês, multiplicando o atributo horasTrabalhadas pelo atributo valorPorHora. Em seguida, escreve o valor na tela.
- O método incrementarHoras adiciona um valor passado por parâmetro ao valor já existente no atributo valorPorHora.

Crie o método main e instancie a classe Funcionário criada, criando um objeto novoFuncionario do tipo Funcionario.

- Atribua o valor “Luis” ao atributo nome.
- Atribua o valor “Silva” ao atributo sobrenome.
- Atribua o valor 10 ao atributo horasTrabalhadas
- Atribua o valor 25.50 ao atributo valorPorHora.
- Chame o método nomeCompleto.
- Chame o método calcularSalario.
- Adicione 8 ao atributo horasTrabalhadas utilizando o método incrementarHoras.
- Chame novamente o método calcularSalario.
- Crie um ArrayList e instancie outros 9 funcionários com diferentes atributos a esse ArrayList.

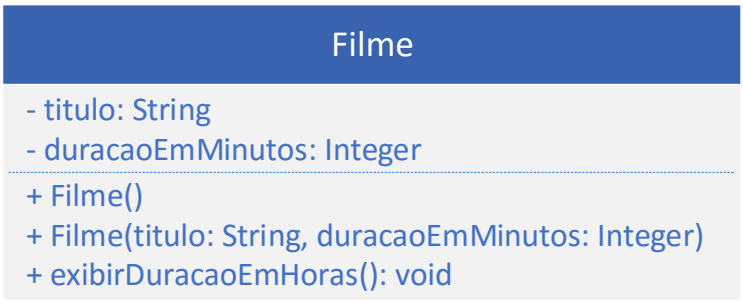
4) Construa a classe Livros em Java, que obedeça à descrição abaixo:

- A classe possui os atributos titulo, qtdPaginas e paginasLidas. Esses atributos devem ser marcados com o modificador de acesso private.
- Crie os métodos get e set para cada um dos atributos.
- Crie ainda o método verificarProgresso que deverá calcular a porcentagem de leitura do livro até o momento. Para isso, deverá usar os valores dos atributos qtdPaginas e paginasLidas, através da formula: porcentagem = paginasLidas * 100 / qtdPaginas. O valor da porcentagem deverá ser mostrado na tela conforme a mensagem "Você já leu X por cento do livro", onde o valor de X é o valor calculado pela fórmula apresentada anteriormente.

5) Crie uma classe **TestarLivros** no mesmo pacote da classe Livros da questão anterior. Essa classe possuirá apenas o método **main** que servirá para testar a classe Livros. As seguintes ações devem ser realizadas:

- Crie um objeto **livrofavorito** do tipo Livro.
- Altere o atributo **titulo** para "O Pequeno Príncipe". Utilize, para isso, o método **setTitulo**;
- Altere o atributo **qtdPaginas** para 98. Utilize, para isso, o método **setQtdPaginas**; Escreva na tela a mensagem: "O livro X possui Y páginas", onde no lugar de X deverá aparecer o valor do atributo **titulo** e, no lugar de Y deverá aparecer o valor do atributo **qtdPaginas**. Utilize, para tanto, os métodos **getTitulo** e **getQtdPaginas**.
- Altere a quantidade de **paginasLidas** para 20;
- Chame o método **verificarProgresso**.
- Altere a quantidade de paginasLidas para 50;
- Chame o método **verificarProgresso**.
- Instancie outros 10 livros no método main e chame os métodos desenvolvidos, conforme o exemplo anterior.

6) Construa a classe Filmes em Java, que obedeça à descrição abaixo:



- A classe deve possuir dois atributos privados: titulo (do tipo String) e duracaoEmMinutos (do tipo int).
- Crie os métodos de acesso (get e set) para os atributos titulo e duracaoEmMinutos.
- Crie um método exibirDuracaoEmHoras que converta o valor em minutos para horas e apresente a mensagem "O filme TITULO possui X horas e Y minutos de duração".
- Por exemplo, dado o filme com título Titanic que possui 194 minutos de duração, a mensagem que deverá ser exibida para o usuário será:

"*O filme Titanic possui 3 horas e 14 minutos de duração*"

7) Crie uma classe TestarFilme que possua um método main de modo que seja possível testar a classe Filme criada na questão anterior.
   - Crie um objeto filme1 do tipo Filme.
   - Altere o atributo título para "Os Vingadores".
   - Altere o atributo duracaoEmMinutos para 142.
   - Chame o método exibirDuracaoEmHoras() para mostrar quantas horas o filme possui.
   - Crie um objeto filme2 do tipo Filme.
   - Altere o atributo título do filme2 para "Hotel Transilvânia".
   - Altere o atributo duracaoEmMinutos do filme2 para 93.
   - Chame o método exibirDuracaoEmHoras() do filme2 para mostrar quantas horas o filme possui.
   - Exiba a mensagem: "Os filmes em cartaz são X e Y", onde no lugar de X, deverá aparecer o título do filme1 e no lugar de Y deverá aparecer o título do filme2.
   - Instancie outros 5 filmes e faça as mesmas ações descritas acima, porém utilizando novos valores.