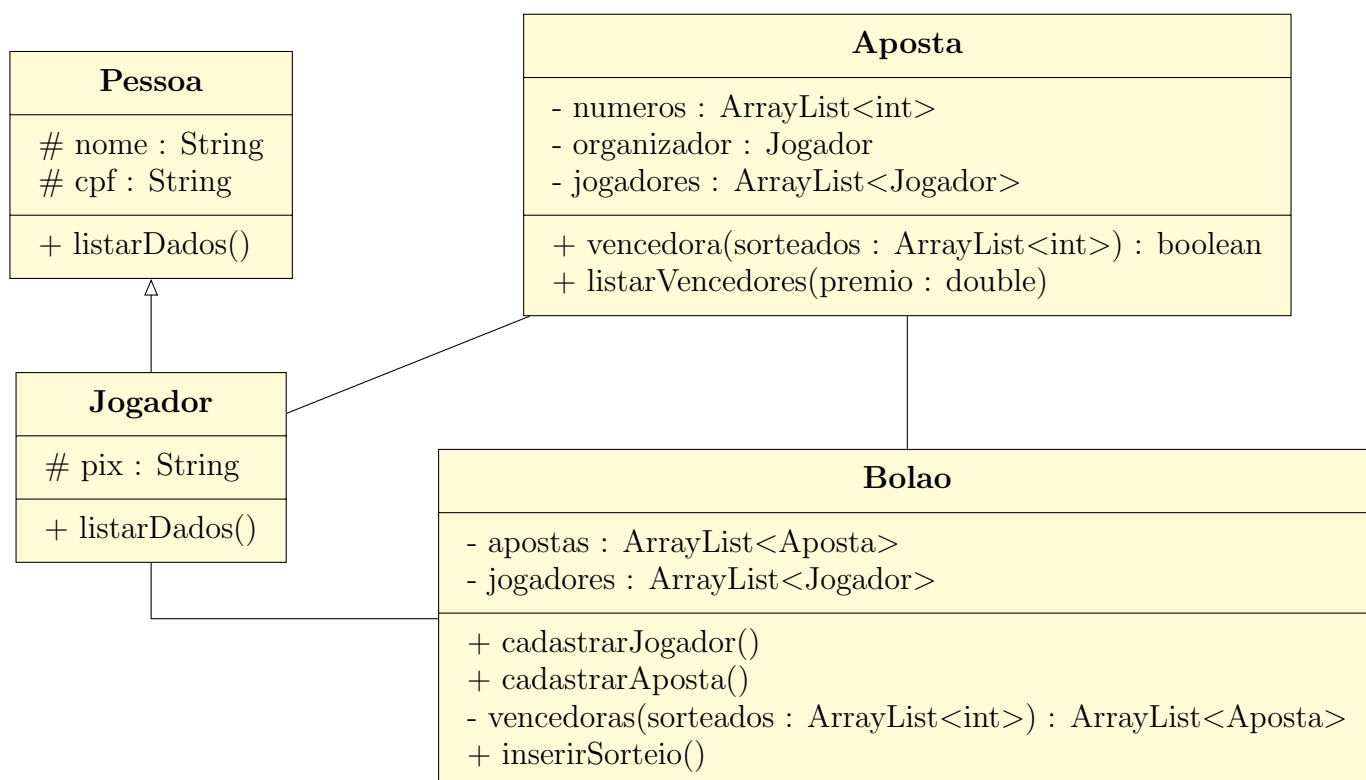


# 1 Bolão

Bolão é uma modalidade de aposta na qual vários apostadores se juntam para adquirir uma série de cartões de apostas, aumentando assim a probabilidade de acertos. Pode-se optar também por uma única aposta, mas com uma quantidade maior de números. Por exemplo, escolhem-se 8 números para apostar na Mega Sena, sendo que apenas 6 números são sorteados nesta loteria. Se 6 desses números forem sorteados, a aposta é considerada vencedora e o prêmio é dividido entre os participantes do bolão. As chances de acerto aumentam, entretanto é bem mais caro apostar mais de 6 números em um mesmo cartão.

Ganancioso, você decidiu criar um sistema para administrar os bolões que irá criar em sua empresa. Cada aposta no sistema representa apenas um cartão, mas a quantidade de números apostados no cartão pode variar (no mínimo 6 números, que é a quantidade de números sorteados). Além disso, cada aposta terá ao menos dois jogadores participando, sendo que um deles será o organizador do bilhete. Caso o bilhete seja premiado, 10% do valor será do organizador, e o restante será dividido igualmente entre ele e os demais jogadores.

Você optou por definir as seguintes classes no sistema:



Você deve implementar ao menos as classes e métodos descritos no diagrama acima, utilizando exatamente o mesmo encapsulamento para cada atributo e método.

O sistema exibirá um menu com as seguintes alternativas:

1. Cadastrar jogador

2. Cadastrar aposta
3. Inserir sorteio e listar vencedores
4. Sair

O arquivo principal deve criar **apenas** um objeto da classe Bolao. Nenhum outro objeto pode ser criado no arquivo principal. Todas as funcionalidades devem ser executadas através dos métodos deste objeto. Métodos auxiliares podem ser criados, se necessários (sugestão: faça um método estático para exibir as opções do menu, ler a opção desejada pelo usuário e retornar a opção lida, caso seja válida).

A seguir, detalhamos cada uma das classes e seus métodos obrigatórios.

## Classe Pessoa

O construtor da classe deve ler os dados digitados pelo usuário (nome e CPF). O método *listarDados()* imprime todas as informações sobre a pessoa (nome e CPF).

## Classe Jogador

O construtor também não recebe nenhum parâmetro, e os dados devem ser lidos do teclado. O campo **pix** define o identificador a ser usado para possível pagamento de seu prêmio. O método *listarDados()* imprime todas as informações sobre o jogador (nome, CPF e pix).

## Classe Aposta

O construtor das classes Aposta deve apenas inicializar seus ArrayLists vazios.

O método *vencedora()* verifica se a aposta foi vencedora em um sorteio, dado um ArrayList com os 6 números sorteados. Para que ela seja vencedora, é preciso que todos os seis números sorteados tenham sido apostados (lembre-se que uma aposta pode conter mais de 6 números).

O método *listarVencedores()* recebe como parâmetro o prêmio destinado àquela aposta, e imprime os dados de todos os participantes desta aposta (organizador e jogadores) e qual o valor do prêmio que cada um deles receberá.

Como o construtor não preenche os dados da aposta, você **pode** criar métodos auxiliares para tal. Por exemplo:

- *inserirNumeros()*: Pede que o usuário digite a quantidade  $q$  de números a serem apostados no bilhete. Em seguida, faz a leitura dos  $q$  números. Não permita que o usuário digite números repetidos ou inválidos (aceite apenas números entre 1 e 60). A inserção deve acontecer **enquanto** o ArrayList ainda não contiver os  $q$  números necessários.

- *inserirOrganizador()*: Recebe como parâmetro o ArrayList com todos os jogadores cadastrados no sistema, imprime os dados dos jogadores, pede que o usuário escolha um jogador para ser o organizador da aposta (digitando seu CPF) e salva-o na aposta.
- *inserirJogadores()*: Dado o ArrayList com todos os jogadores cadastrados no sistema e solicita que o usuário digite o número  $j$  de jogadores do bilhete (além do organizador, que já foi cadastrado). Em seguida, para cada  $1 < i \leq j$ , lista os jogadores cadastrados no sistema, pede que o usuário informe o CPF do jogador, e insere-lo no ArrayList com os jogadores desta aposta. É possível inserir jogadores repetidos (um mesmo jogador por ter mais de uma cota de participação no bilhete).

Caso o CPF digitado do organizador ou do jogador seja inválido, você pode optar por selecionar, por exemplo, o primeiro jogador encontrado.

## Classe Bolao

O construtor da classe Bolao também deve apenas inicializar seus ArrayLists vazios.

O método *cadaststrarJogador()* cria um novo jogador e insere-o no ArrayList de jogadores.

O método *cadaststrarAposta()* cria uma nova aposta (que, a princípio, estará vazia), faz a leitura de seus dados (preferencialmente utilizando os métodos auxiliares da classe Aposta), e insere esta nova aposta no ArrayList de apostas.

O método *vencedoras* recebe o ArrayList com os 6 números sorteados, e retorna um novo ArrayList contendo todas as apostas premiadas. Ele é privado pois será utilizado apenas como método auxiliar do método *inserirSorteio()*.

O método *inserirSorteio()* faz a leitura dos 6 números sorteados e imprime as informações sobre as apostas premiadas (os dados de seus jogadores e o valor do prêmio recebido por cada um). Sugestão:

1. Comece o método fazendo a leitura dos 6 números sorteados.
2. Faça a leitura do prêmio total a ser dividido entre os bilhetes premiados e salve esta informação.
3. Crie um ArrayList auxiliar contendo as apostas vencedoras (com auxílio do método *vencedoras()*).
4. Calcule o prêmio a ser recebido por cada bilhete premiado (divida o valor do prêmio total igualmente pela quantidade de apostas vencedoras).
5. Liste as apostas vencedoras. Para cada aposta vencedora no ArrayList auxiliar, chame o método *listarVencedores()* desta aposta, passando como parâmetro o valor do prêmio a ser recebido por cada bilhete.

## 2 Observações

- O trabalho vale 30 pontos e deve ser entregue até 26/04.
- O código deve ser feito em Java, considerando as boas práticas de POO (como reuso de código, por exemplo, sempre que possível).
- O trabalho pode ser feito em grupos de até dois integrantes.
- Trabalhos entregues após o prazo serão automaticamente rejeitados.
- Trabalhos com erro de execução, com formato de saída incorreto, ou que não compilarem terão nota 0.
- Trabalhos considerados plágio terão nota 0 para quem copiou e para quem forneceu o trabalho, e serão enviados ao Conselho de Ética.
- O trabalho deve ser enviado na sala da disciplina do AVA.
- Em caso de dúvidas na especificação do trabalho ou no próprio trabalho, contate-me em sala de aula.