

**Candidato:** 067

**Desafio escolhido:** Crossbets (cassino)

**Classe presente:**

Apostador

- Responsável por armazenar informações sobre o cliente, principalmente a quantidade de “créditos” (dinheiro) que ele possui, mas também a quantidade de vitórias e derrotas.
- O construtor inicializa as variáveis vitórias e derrotas em 0 e os créditos com base numa quantidade inserida pelo usuário.
- É utilizado em todos os jogos para somar/subtrair o valor de créditos com base na aposta feita pelo jogador.

**Funções presentes:**

int main()

- Responsável por inicializar a seed aleatória, dar as boas vindas ao jogo, criar um objeto de classe Apostador e chamar a função cassino.

void cassino(Apostador \*cliente)

- Um loop que informa as opções do jogador (consultar os dados, jogar ou sair) e chama a função processaEscolha para decidir o que fazer com base no input.
- O loop se interrompe com base no retorno dado pela função processaEscolha, que retorna verdadeiro (para sinalizar o fim) caso o jogador escolha a opção de parar.

bool processaEscolha(int escolha, Apostador \*cliente)

- Uma extensão da função cassino, separada por questões de organização, é responsável por chamar as funções que o jogador pediu (mostrar as informações, jogar ou parar) e interromper o loop do cassino quando necessário.

void mostraInformacoes(Apostador \*cliente)

- Mostra pro jogador a quantidade de créditos disponível, a quantidade de vitórias e a quantidade de derrotas.

void chamaJogos(Apostador \*cliente)

- Mostra ao jogador as opções de jogos (Blackjack, Caça-Níqueis e Roleta) e chama as funções relativas a cada jogo com base no input do jogador.

- Usa a função pedeAposta para perguntar quanto de dinheiro o jogador quer apostar e dá essa informação para as funções dos jogos.
- Caso o jogador perca todo o dinheiro ele é “expulso” do cassino, ou seja, o programa é finalizado.

int pedeAposta(Apostador \*cliente)

- Uma extensão da função chamaJogos, separada por organização também, pede para o jogador inserir a quantia que quer apostar e retorna esse valor caso seja válido, se não for pede para o jogador inserir novamente até que seja inserido algo válido.
- Para decidir se é válido é verificado se o valor inserido não é menor que 0 nem maior que a quantia de dinheiro atual do jogador.

void blackjack(int aposta, Apostador \*cliente)

- Usa-se um loop para dar continuidade ao jogo. A cada rodada é informado ao jogador a quantidade de pontos dele e da banca, então é perguntado se ele deseja continuar jogando (inclusive na 1ª, mas isso faz ele perder).
- Se o jogador disser sim, é sorteado para ele um valor de 1 a 10, representando as cartas do Ás ao rei, então é mostrado a ele qual carta foi. Após isso, é feito o mesmo para a banca, mas para o caso dela é decidido se ela continua jogando com base na quantidade total de pontos, se for 16 ou menos ela joga mais uma vez, caso contrário ela para.
- O loop também é interrompido se qualquer um dos dois ultrapassar o valor de 21.
- Ao final do jogo verifica-se uma série de condições para decidir a vitória

void cacaNiqueis(int aposta, Apostador \*cliente)

- Sorteia 3 valores int aleatórios entre 0 e 9 e os coloca num vetor, mostrando ao jogador como símbolos da tabela ascii entre 33 (!) e 42 (\*).
- A vitória é decidida com base numa variável booleana que inicialmente recebe o valor verdadeiro, com o jogador ganhando caso o valor seja verdadeiro e perdendo se for falso.
- A partir da segunda iteração do loop que faz o sorteio, é verificada se o símbolo atual é diferente do símbolo anterior, caso seja, a variável de vitória recebe o valor falso, caso não seja (ou seja, se forem o mesmo símbolo), o valor permanece inalterado.
- Atualmente a chance de vitória é de 1% (1000 combinações diferentes já que cada coluna pode ter 10 valores diferentes, e 10 casos onde os símbolos são iguais), mas pode ser ajustada alterando a quantidade de colunas sorteando o valor ou alterando a quantidade de símbolos que podem ser sorteados.

void roleta(int aposta, Apostador \*cliente)

- Pergunta ao jogador que tipo de aposta quer fazer, com as opções sendo números individuais de 0 a 36, cores (onde, para simplificar as coisas, o preto é o mesmo que números pares exceto 0 e o vermelho o mesmo que números ímpares), ou 3 grupos cada um com 12 números.
- Dependendo do tipo de aposta escolhida, pergunta qual número, cor ou grupo de número o jogador quer.
- Então, sorteia o número e com base na escolha compara se o jogador ganhou ou perdeu.
- Para o caso de números simplesmente é verificado se os números são iguais.
- Para a cor, vê se o número sorteado é par ou ímpar.
- Para o grupo de números, exceto 0 que não entra, pega a divisão inteira do número sorteado por 12, o que faz os números serem de 0 a 3, e soma 1 caso o número seja múltiplo de 12, o que faz bater com a divisão de grupos.