Konstruowanie obiektów

Przypomnienie wiadomości o obiektach plus obiekt w obiekcie! https://www.javappa.com/kurs-java/wlasne-obiekty



Założenia:

- Losowanie kart osobna klasa
- Logika gry osobna klasa
- Wyświetlanie informacji dla gracza osobna klasa
- Klasa główna programu

W tym ćwiczeniu zasymulujemy grę Blackjack.

Otrzymasz dwie karty i będziesz mógł zobaczyć odkrytą kartę krupiera (jedną). Wszystkie karty są reprezentowane za pomocą ciągu, takiego jak "as", "król", "trzy", "dwa" itp. Wartości każdej karty to:

card	value	card	value
as	11	osiem	8
dwa	2	dziewięć	9
trzey	3	dziesięć	10
cztery	4	walet	10
pięć	5	dama	10
sześć	6	król	10
siedem	7	inna	0

Note: Zazwyczaj AS przyjmuje wartość 1 lub 11 ale dla uproszczenia przyjmij wartość tylko 11.

W zależności od Twoich dwóch kart i karty krupiera, istnieje strategia na pierwszą turę gry, w której masz następujące opcje:

- Stand (S) czekam
- Hit (H) dobieram
- Split (P) rozdziel: (jeśli masz dwie takie same karty) gracz ma teraz dwa

zakłady, do każdego dobiera kartę i wybiera lepszy układ kard dla siebie

- Automatically win (W) – zwycięstwo automatyczne

Chociaż nie jest to jeszcze optymalna strategia, zastosujesz strategię rozwijaną przez twojego przyjaciela Alexa, która wygląda następująco:

Kategoria: Duża ręka

- Jeśli masz parę asów, zawsze musisz je rozdzielić (S).
- Jeśli masz blackjacka (dwie karty, które sumują się do wartości 21), a krupier nie ma asa, walata, damy, króla ani dziesiątki, automatycznie wygrywasz (W). Jeśli krupier ma jakąkolwiek z tych kart, będziesz musiał stać i czekać na odkrycie drugiej karty.

Kategoria: Małą ręka

- Jeśli suma twoich kart wynosi 17 lub więcej, zawsze powinieneś stać.
- Jeśli suma twoich kart wynosi 11 lub mniej, zawsze powinieneś dobrać (H).

• Jeśli suma twoich kart osiąga wartość w zakresie [12, 16], zawsze powinieneś czekać(S), chyba że krupier ma 7 lub wyższą, w takim przypadku zawsze powinieneś dobrać(H).

Ogólna logika została już wdrożona. Masz cztery zadania:

1. Oblicz wynik dowolnej karty.

Zaimplementuj funkcję do obliczania wartości liczbowej karty podaną jej nazwą za pomocą warunków zapisanych w tabeli.



// returns 11

2. Sprawdź, czy dwie karty składają się na blackjacka.

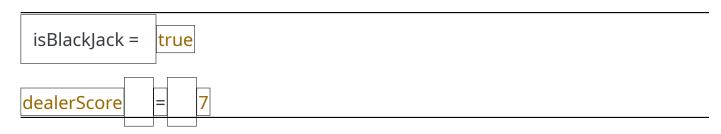
Zaimplementuj funkcję, która zwraca prawdę, jeśli dwie karty tworzą blackjacka, w przeciwnym razie fałsz.



// returns true

3. Zastosuj logikę decyzyjną dla wyników rozdań większych niż 20 punktów.

Zaimplementuj funkcję, która zwraca ciąg reprezentujący decyzję podaną na Twoich kartach. Ta funkcja jest wywoływana tylko wtedy, gdy handScore jest większy niż 20. Otrzyma 2 argumenty: isBlackJack i dealerScore. Powinna implementować warunki w kategorii "Duża ręka" powyżej.



largeHand(isBlackJack, dealerScore)

// returns "W"

4. Zastosuj logikę decyzyjną dla wyników rozdań z mniej niż 21 punktami.

Zaimplementuj funkcję, która zwraca ciąg reprezentujący decyzję podaną na Twoich kartach. Ta funkcja jest wywoływana tylko wtedy, gdy handScore jest mniejszy niż 21. Otrzyma 2 argumenty: handScore i dealerScore. Powinna implementować warunki w kategorii "Mała ręka" powyżej.

handScore = 15

dealerScore = 12

SmallHand(handScore, dealerScore)

// returns "H"

5. Niech gra działa!