Praktikumsaufgabe 3, Black Jack

Sie sollen eine vereinfachte Variante von Black Jack programmieren. Black Jack ist ein Karten-Glücksspiel. Ziel des Spiels ist es, mit zwei oder mehr Karten näher an 21 Punkte heranzukommen als der Croupier, ohne dabei den Wert von 21 Punkten zu überschreiten.

Der Spieler darf spielen, bis sein Guthaben (Startguthaben: 10 Chips) verbraucht ist. Er muss mindestens einen und darf maximal alle Chips setzen. Der Croupier setzt jeweils die gleiche Menge.

Asse zählen entweder ein oder elf Punkte. Der Einfachheit halber werden sie hier immer mit elf Punkten gewertet. Zweier bis Zehner zählen entsprechend ihrer Augen. Alle Bildkarten zählen zehn Punkte.

Der Spieler ist der Nutzer. Der Croupier wird vom Computer simuliert. Der Computer zieht solange Karten, bis er mehr als 16 hat (oder genau 21 oder mehr).

Das Kartenziehen soll mit Zufallszahlen aus 2 bis 11 simuliert werden.

Dafür nutzen wir die Klasse *Random*. Random stellt einen Generator für Pseudozufallszahlen dar. *Random.Next* ist eine Methode, um zufällige ganze Zahlen zu erhalten.

```
Random zufall = new Random();
while (true)
{
      int eineKarte = zufall.Next(2, 12);
      // erster Wert inklusive, zweiter Wert exklusive
      Console.WriteLine(eineKarte);
}
```

Die (vereinfachten) Regeln:

- Vor Beginn des jeweiligen Spiels setzt der Spieler seine Anzahl an Chips.
- Dann beginnt zuerst der Spieler. Er muss mindestens zwei Karten nehmen.
- Dann darf er solange neue Karten ziehen, bis er 21 oder mehr hat oder meint, genug Punkte zu haben.
- Die aufsummierten Wertigkeiten dürfen 21 nicht überschreiten, sonst verliert er, und sein Einsatz geht an die Bank.
- Will der Spieler keine Karten mehr und hat nicht mehr als 21, zieht der Computer so lange, bis er mehr als 16 bzw. mehr als 21 hat.

- Hat der Computer mehr als 21 Punkte, hat er verloren und der Einsatz der Bank geht an den Spieler.
- Haben beide 21 Punkte oder weniger, gewinnt der Computer, wenn die Summe der Wertigkeiten seiner Karten größer oder gleich der des Spielers ist. Ansonsten gewinnt der Spieler.
- Gewinnt der Spieler, behält er seinen Einsatz und bekommt den Einsatz der Bank.

Das Spiel wird so lange wiederholt, bis der Spieler aufhören möchte oder keine Chips mehr besitzt.

Hinweise:

Gehen Sie bei der Entwicklung Ihres Programms schrittweise vor, zum Beispiel so:

1. Programmieren Sie eine Schleife für den Part des Spielers. Hier werden die Zufallszahlen so lange aufsummiert bis
    a. der Spieler aufhören möchte ODER
    b. der Spieler 21 oder mehr hat.
2. Programmieren Sie eine ähnliche Schleife für den Part des Computers. Diese Schleife wird nur ausgeführt, wenn der Spieler nicht mehr als 21 hat. Hier werden die Zufallszahlen so lange aufsummiert bis
    a. der Computer mehr als 16 hat ODER
    b. der Computer 21 oder mehr hat.
3. Nun programmieren Sie die Entscheidung, wer gewonnen hat (Regeln s. oben).
4. Programmieren Sie die nötigen Schritte für das Setzen der Chips vor der Schleife aus Schritt 1. Erhöhen bzw. vermindern Sie das Guthaben des Spielers, wenn er gewonnen bzw. verloren hat.
5. Abschließend betten Sie Ihr gesamtes Programm in eine weitere Schleife ein, die abbricht, wenn der Spieler aufhören möchte oder keine Chips mehr hat.

Testen Sie Ihr Programm nach jedem einzelnen Schritt ausführlich. Erst, wenn Sie ganz sicher sind, dass ein Teil funktioniert, gehen Sie zum nächsten über.

So könnten Programmabläufe aussehen:

```
Dein Guthaben: 10 Chips
Dein Einsatz: 2
Karte:  7, Gesamt:  7
Karte:  9, Gesamt: 16
Noch eine Karte (j/n)? j
Karte:  4, Gesamt: 20
Noch eine Karte (j/n)? n

Ich bin dran.
Karte:  4, Gesamt:  4
Karte:  4, Gesamt:  8
Karte:  4, Gesamt: 12
Karte:  2, Gesamt: 14
Karte:  5, Gesamt: 19

Du hast gewonnen.

Dein Guthaben: 12 Chips.

Nochmal (j/n)?
```

```
Dein Guthaben: 10 Chips
Dein Einsatz: 13
Du musst mindestens 1 und kannst max. 10 Chips setzen.
Dein Einsatz: 4
Karte:  7, Gesamt:  7
Karte:  9, Gesamt: 16
Noch eine Karte (j/n)? j
Karte: 10, Gesamt: 26

Ich habe gewonnen.

Dein Guthaben: 6 Chips.

Nochmal (j/n)?
```