Fragen zu Strukturierter Prg.

Mittwoch, 25. Oktober 2017 21:10

Aufgaben

1. Es soll folgendes Spiel durch ein Programm simuliert werden:
   Der Spieler zahlt für jedes Spiel einen Einsatz von 1 €. Er darf einmal mit 3 Würfeln würfeln. Je nach gewürfelter Punktzahl bekommt der Spieler einen Gewinn entsprechend der Tabelle ausgezahlt. Benutzen Sie zum Simulieren der Zufallszahlen 1-6 die Random-Klasse des .NET-Frameworks.

Punktzahl	Gewinn in €
3-14	0
15	2
16	5
17	10
18	100

2. Das nachfolgende Programm ist syntaktisch korrekt und könnte somit übersetzt und ausgeführt werden. Es enthält jedoch vier Beispiele für logische Fehler, die bei einem Programmlauf teilweise zu einem Abbruch führen würden. Finden und korrigieren Sie die Fehler ohne Benutzung des Compilers.

```
public class Falsch
{
    public static void Main()
    {
        int x = 0, y = 4;
        //Beispiel A
        if (x < 5)
            if (x < 0)
                Console.WriteLine("x < 0");
        else
            Console.WriteLine("x >=5");

        // Beispiel B
        if (x > 0)
            Console.WriteLine("ok! x > 0");
            Console.WriteLine("1/x = " + (1/x));

        //Beispiel C
        if (x > 0);
            Console.WriteLine("1/x = " + (1/x));

        //Beispiel D
        if (y < x)
        { //vertausche x und y
            x = y;
            y = x;
```

```
        }
        Console.WriteLine("x = " + x + " y = " + y);
    }
}// Falsch
```

3. Warum ist nachfolgender Code ein Beispiel für schlechten Programmierstil.

   Erstellen Sie einen Schreibtischtest für die Ausgabe.

```
public class badSchleife
{
    public static void Main()
    {
        int i, j;
        for (i=1; i<=10; i++)
        { // Schleife A
            Console.WriteLine("A1: i = " + i);
            i = 5;
            Console.WriteLine("A2: i = " + i);
            for (i = 7; i<=20; i++)
            { // Schleife B
                Console.WriteLine("B1: i " + i);
                i = i + 2;
                Console.WriteLine("B2: i " + i);
            }
        }
    }
}
```

4. Welches Ergebnis erzeugt untenstehender Code

```
    class SwitchTest
    {
        const int EINS = 1;

        public void testSwitch (int zahl)
        {
            switch (zahl)
            {
                case EINS:
                {
                    Console.WriteLine("Testergebnis: "+EINS);
                    break;
                }
                case 2:
                {
                    Console.WriteLine("Testergebnis: "+2);
                    break;
                }
            }
        }

        public static void Main (String[] args)
        {
            SwitchTest test = new SwitchTest();
```

```
            test.testSwitch (1);
            test.testSwitch (2);
            test.testSwitch (EINS);
        }
    }
```

5. Der Algorithmus

    1. Lies den Wert von n ein.
    2. Setze i auf 3.
    3. Solange i < 2n, wiederhole
       - Erhöhe i um 1.
       - Gib {1}/{2i+1} aus

    soll auf drei verschiedene Arten implementiert werden.

    Schreiben Sie jeweils ein C#-Programmstück, das diesen Algorithmus als **while**, als **for** und als **do while** - Schleife realisiert. Sämtliche Programmstücke sollten die gleiche Ausgabe erzeugen

6. Erstellen Sie ein Programm, das Ganzzahlen beliebigen Wertes annimmt und in der binären Darstellung ausgibt. Benutzen Sie die **do-while**-Schleife zur Ermittlung der Binär-Zahl.

7. Ein Programm soll einen Automaten zur Geldrückgabe simulieren. Das Programm erhält einen Betrag in Form einer Gleitkommazahl als Parameter in der Kommandozeile. Dann soll der Betrag mit möglichst wenig Münzen (Euro) ausgegeben werden.

1. In einer Fabrik für Fahrrad-Nummernschlösser soll der neue Computer die Schließnummern festlegen. Jedes Schloss wird mit einer dreistelligen Nummer geöffnet.

    Der Computer soll alle Nummern ausgeben. Fahrradschlösser mit zwei oder drei gleichen Ziffern nimmt die Kundschaft nicht ab, also darf der Computer sie nicht aufschreiben. Damit nicht so viel Platz verbraucht wird, soll der Computer jeweils 10 Nummern nebeneinander schreiben, etwa so.

```
012 013 014 015 016 017 018 019 021 023
024 025 026 027 028 029 031 032 034 035
036 037 038 039 041 042 043 045 046 047
048 049 051 052 053 054 056 057 058 059
061 062 063 064 065 067 068 069 071 072
073 074 075 076 078 079 081 082 083 084
085 086 087 089 091 092 093 094 095 096
097 098 102 103 104 105 106 107 108 109
120 123 124 125 126 127 128 129 130 132
```