

1. Deze opgave bestaat uit een aantal vragen. Houd het antwoord kort: één of twee zinnen per onderdeel kan al genoeg zijn.

(a) (2 punten) Wat is het verschil tussen een abstracte klasse en een interface? Geef een voorbeeld van wat niet mogelijk is met een interface en wel met een abstracte klasse.

**Antwoord:** Een interface defineert een lijst van methoden en properties, die een klasse kan implementeren. Een abstracte klasse heeft een of meerdere methoden of properties die geen implementatie hebben. In tegenstelling tot een interface kun je membervariabelen, methoden en properties definiëren in een abstracte klasse.

(b) (2 punten) Wat is de betekenis van de operator %? Geef een expressie die voor positieve getallen  $x$  en  $y$  dezelfde waarde heeft als  $x\%y$ , zonder daarbij de operator % te gebruiken.

**Antwoord:** De operator % geeft de rest bij deling. Een equivalente expressie voor  $x\%y$  is  $x - (x/y)*y$ .

(c) (2 punten) Wat betekent het woord **virtual** als het in de definitie van een methode gebruikt wordt? Mag **virtual** ook gebruikt worden voor properties?

**Antwoord:** Dit betekent dat de methode (of property) vervangen (overridden) mag worden in een subklasse. **virtual** mag gebruikt worden voor methoden en properties.

(d) (2 punten)

**Antwoord:**

```
1 // aanpak 1: vooraf controleren
2 if (waarde<0)
3     s = "onmogelijk";
4 else
5     s = "De wortel is " + Math.Sqrt(waarde);
6
7 // aanpak 2: foutsituatie afvangen
8 try
9 {
10     s = "De wortel is " + Math.Sqrt(waarde);
11 }
12 catch (Exception e)
13 {
14     s = "onmogelijk";
15 }
```

2. (12 punten) Geven is de volgende membervariabele in een klasse:

```
1 int[] array = { 3, 1, 5, 3, 8, 2, -1, 1, 1 };
```

Hieronder staan zes methoden uit diezelfde klasse. Geef in elk van de gevallen aan wat de return-waarde van de methode-aanroep is. Licht het antwoord kort toe.

1	<pre>int Methode1() {     int result = 0;     for (int i = 0; i &lt; array.Length; ++i)         result += array[i];     return result; }</pre>	23: de som van alle elementen in de array.
2	<pre>int Methode2() {     int result = 0, i = 0;     while (i &lt; array.Length) {         result += array[i];         i += array[i];     }     return result; }</pre>	9: 3 + 3 + -1 + 2 + 1 + 1.
3	<pre>int Methode3() {     int result = 0;     for (int i = 0; i &lt; array.Length; ++i)         if (i % 2 == 0)             result += array[i];     return result; }</pre>	16: de som van alle elementen met even index.
4	<pre>int Methode4() {     int result = 0, i = 0;     while (i &lt;= array[i]) {         result += array[i];         ++i;     }     return result; }</pre>	20: 3 + 1 + 5 + 3 + 8. Hierna stopt de loop want 5 > 2.
5	<pre>int Methode5() {     int result = 0;     for (int i = 0; i &lt; array.Length - 1; ++i)         if (array[i] &gt; array[i + 1])             result += array[i];     return result; }</pre>	18: 3 + 5 + 8 + 2.
6	<pre>int Methode6() {     int result = -1000;     for (int i = 0; i &lt; array.Length; ++i)         if (array[i] &gt; result)             result = array[i];     return result; }</pre>	8: de hoogste waarde in de array.

3. Gegeven zijn de volgende twee klassedefinities:

```
1  class BaseClass {
2      private int var1;
3      protected int var2;
4      public int var3;
5      int var4;
6  }
7
8  class DerivedClass : BaseClass {
9      public int var5;
10
11     public void Method1() {
12         int i = 0;
13     }
14 }
```

(a) (2.5 punten) Kruis aan welke van de volgende opdrachten mogen worden uitgevoerd in de body van Method1 in de klasse DerivedClass, na de declaratie van variabele i:

- i = var1;
- i = var2;
- i = var3;
- i = var4;
- i = var5;

(b) (7.5 punten) De volgende opdrachten staan in een methode van een andere (ongelateerde) klasse:

```
1  BaseClass obj = new BaseClass();
2  DerivedClass obj2 = new DerivedClass();
3  BaseClass obj3 = obj2;
4  int i = 0;
```

Kruis aan welke van de volgende opdrachten mogen worden uitgevoerd in de body van die methode:

- i = obj.var1;
- i = obj.var2;
- i = obj.var3;
- i = obj.var4;
- i = obj2.var1;
- i = obj2.var2;
- i = obj2.var3;
- i = obj2.var4;
- i = obj2.var5;
- i = obj3.var1;
- i = obj3.var2;
- i = obj3.var3;
- i = obj3.var4;
- i = obj3.var5;
- obj3.Method1();

4. Gegeven is de volgende klasse:

```
1 class TetrisBlok
2 {
3     private bool[,] configuratie;
4     private const int size = 4;
5
6     public TetrisBlok() {
7     }
8 }
```

Deze klasse gebruiken we om een blok voor te stellen in de game Tetris. We gaan er in deze klasse vanuit dat elk tetrisblok in een configuratie van  $\text{size} * \text{size}$  deelblokjes beschreven wordt, waarbij **true** staat voor bezet en **false** staat voor vrij. In de tweedimensionale array staat de eerste dimensie voor de  $x$ -positie en de tweede dimensie voor de  $y$ -positie van elke deelblokje in de configuratie.

- (a) (2 punten) Er mist nog iets in deze klasse. Wat is dat? Schrijf de missende opdracht(en) op en geef aan waar ze toegevoegd moeten worden.

**Antwoord:** De array moet nog geïnitieerd worden. Voeg in de constructormethode toe:

```
1 this.configuratie = new bool[size, size];
```

- (b) (4 punten) We willen dat andere objecten kunnen opvragen of een bepaald deelblokje in de configuratie bezet is. Schrijf een methode **IsBezet** die berekent of een gegeven positie binnen het tetrisblok bezet is. Indien de indices gegeven als parameter buiten het bereik van de array vallen, dan dient de methode een **IndexOutOfRangeException** object te werpen.

**Antwoord:**

```
1 public bool IsBezet(int x, int y)
2 {
3     if (x < 0 || x >= size || y < 0 || y >= size)
4         throw new IndexOutOfRangeException("error: index should be between 0 and " + size);
5     return configuratie[x, y];
6 }
```

- (c) (4 punten) Schrijf een methode genaamd **DraaiRechtsom**. Deze methode draait het tetrisblok 90 graden naar rechts.

**Antwoord:**

```
1 public void DraaiRechtsom()
2 {
3     bool[,] nieuweconfiguratie = new bool[size, size];
4     for (int i = 0; i < size; ++i)
5         for (int j = 0; j < size; ++j)
6             nieuweconfiguratie[size - j - 1, i] = configuratie[i, j];
7     configuratie = nieuweconfiguratie;
8 }
```