

TWEEDE DEELTENTAMEN IMPERATIEF PROGRAMMEREN - VERSIE 1
VRIJDAG 19 OKTOBER 2018, 17.00-19.00 UUR

- Schrijf op elk ingeleverd blad je naam.
- De lijst met standaardfuncties na afloop graag weer inleveren.
- Opgave 1 t/m 10 zijn meerkeuzevragen, die meetellen voor $10 \times 4 = 40$ punten.
Opgave 11 t/m 15 zijn tekstvragen, die meetellen voor $5 \times 6 = 30$ punten.
Opgave 16 en 17 zijn programmeervragen, die meetellen voor $2 \times 15 = 30$ punten.

Meerkeuzevragen: de letter van het goede antwoord volstaat.

Belangrijk: dit is versie 1 van het tentamen, vermeld dat boven je antwoorden.

1. Een *static* methode ...
 - (a) mag niet door methoden buiten de eigen klasse worden aangeroepen
 - (b) mag de variabelen die in de klasse zijn gedeclareerd niet gebruiken
 - (c) kan geen nieuwe objecten aanmaken
 - (d) levert geen resultaatwaarde op
2. De situatie die ontstaat door `class A : B { C D; }` kan beschreven worden door
 - (a) A heeft een member met type C
 - (b) B heeft een member D
 - (c) B erft alle members van A
 - (d) C heeft het type D
3. Wat is de waarde van '2' + 3? (Let goed op de aanhalingstekens!)
 - (a) 53
 - (b) 5
 - (c) '5'
 - (d) "23"
4. Na de declaratie `int[,] a = new int[1,1];` heeft de array `a`
 - (a) nog geen elementen
 - (b) 1 element met de waarde 0
 - (c) 2 elementen, elk met de waarde 1
 - (d) 4 elementen met een nog niet bepaalde waarde
5. Wat wordt er aangeduid met de expressie `base` als die in de body van een methode `M` van een klasse `K` gebruikt wordt?
 - (a) De superklasse van `K`
 - (b) De methode `M` in de superklasse van `K`
 - (c) Het object `this` met de superklasse van `K` als type
 - (d) De constructormethode van de superklasse van `K`

6. Wat is een *virtual* methode?
- (a) Een methode die alleen vanuit een subklasse kan worden aangeroepen
 - (b) Een methode die alleen kan worden aangeroepen vanuit de klasse waarin hij is gedefinieerd
 - (c) Een methode die niet wordt geërfd door de subklassen
 - (d) Een methode die in een subklasse opnieuw kan worden gedefinieerd
7. Wanneer is de declaratie `A b = new C();` toegestaan?
- (a) Als de constructormethode van klasse A virtual is
 - (b) Als de constructormethode van klasse C wordt *gecast* naar A
 - (c) Als de constructormethode van klasse C is gedefinieerd in de klasse A
 - (d) Als klasse C een subklasse is van klasse A
8. Een *null reference exception* treedt op
- (a) In het `try`-gedeelte van een `try-catch` opdracht waarin `catch(NullReferenceException e)` staat
 - (b) In het `catch`-gedeelte van een `try-catch` opdracht waarin `catch(NullReferenceException e)` staat
 - (c) Als je een membervariabele gebruikt van een objectvariabele die niet is gedeclareerd
 - (d) Als je een membervariabele gebruikt van een objectvariabele waaraan geen waarde is toegekend
9. Iemand schrijft een methode om een getal `x` tot een niet-negatieve macht `e` te verheffen:

```
int Macht(int x, int e)
{
    int res = 1;
    for (int t=e; t<0; t--)
        res *= x;
    return res;
}
```

Welk ongewenst effect heeft deze methode?

- (a) de herhaling gaat één stap te lang door
 - (b) de uitkomst is altijd 1
 - (c) de herhaling gaat oneindig lang door
 - (d) de herhaling stopt één stap te vroeg
10. Als je twee buttons dezelfde `Click` event-handler geeft, en je wilt in de event-handler bepalen welke button was ingedrukt, dan
- (a) heb je pech: dit kan niet
 - (b) gebruik je een property van de `EventArgs`-parameter van de event-handler
 - (c) gebruik je de `object`-parameter van de event-handler
 - (d) bekijk je voor welk van de twee buttons (die als member-variabele moeten worden gedeclareerd) de property `Clicked` de waarde `true` heeft

Tekstvragen: geef kort en duidelijk antwoord in woorden.

11. (a) Beschrijf de syntax van de opdrachtvorm die met `if` begint.
(b) Beschrijf de semantiek van deze opdrachtvorm.
(NB: Geef de beschrijvingen in algemene termen, dus *niet* aan de hand van een voorbeeld.)
12. In de body van een property kunnen een `get` en een `set` mini-methode worden gedefinieerd.
 - (a) In welke situatie wordt de `get`-minimethode aangeroepen, en in welke situatie de `set`-minimethode?
 - (b) Welke rol speelt het keyword `value` daarbij?
13. Een `string`-object is *immutable*.
 - (a) Wat houdt dat in?
 - (b) Hoe is er bij de definitie van de *indexer-property* van `string` voor gezorgd dat een `string` inderdaad *immutable* is?
14. In een library kan de structuur van een nieuw type objecten worden beschreven. De auteur van de library heeft daarbij de keus om een `class` of een `struct` te definiëren.
 - (a) Wat is een voordeel van een `class` boven een `struct`?
 - (b) Wat is een gevaar bij een keuze voor `class`, dat je niet hebt bij een `struct`?
15. Bekijk een programma waarin alle aangeklikte punten op een `Form` moeten worden gemarkeerd. Iemand doet dit door in de handler van het `MouseClicked`-event de methode `this.CreateGraphics()` aan te roepen, en daarmee vervolgens een markering op het aangeklikte punt te tekenen.
 - (a) Waarom is dat geen goed idee?
 - (b) Hoe moet het dan wel?
(omschrijf dit in woorden, een programmavoorbeeld is niet nodig)

Programmeervragen: hier schrijf je een stukje programma

16. In de klasse `String` zitten onder andere de volgende methodes:

```
string Substring(int start, int lengte)
bool Contains (string s)
```

De hier bedoelde versie van `Substring` levert het deel van de `string` op vanaf het meegegeven startpunt, met de meegegeven lengte. Bij te grote lengte geeft de methode zo veel letters als hij heeft. Voorbeelden:

```
"Utrecht".Substring(2,4) geeft "rech"
"Utrecht".Substring(3,10) geeft "echt"
```

De methode `Contains` geeft aan of de parameter ergens voorkomt in het totaal. Voorbeelden:

```
"Utrecht".Contains("ech") geeft true
"Utrecht".Contains("trh") geeft false
"Utrecht".Contains("Utrecht") geeft true
```

Stel dat je de auteur van de klasse `String` bent. Enkele methoden en properties zijn al geschreven: de `indexers-property` om een losse letter te pakken, de `Length-property`, de operator `+`, `==`, of als je wilt de methodes `Concat` en `Equals` zijn al beschikbaar (die mag je dus gebruiken). De andere methoden ontbreken nog en mag je dus niet aanroepen.

De opgave: Schrijf de twee hierboven beschreven methoden.
(De zelf-geschreven methodes mogen wel elkaar aanroepen).

17. In onderstaande klasse wordt een array gedeclareerd met daarin de uitslagen van een tentamen. Opgave: schrijf de body van de methode `Teken`, die wordt gebruikt als event-handler om een staafdiagram van deze uitslagen op het scherm te tekenen, zoals in de afbeelding.

De uitslagen worden daarbij naar beneden afgerond, dus bijvoorbeeld een 7.5 wordt meegeteld in de staaf behorend bij 7. Het programma moet ook gebruikt kunnen worden als, later, de array wordt aangepast (waarna het programma opnieuw wordt gecompileerd). De breedte van de balkjes is 10 beeldpunten voor elke corresponderende uitslag.

```
public class Staafdiagram : Form
{
    double [] cijfers
    = { 10, 8, 9, 7, 10, 6.5, 8, 7.5, 9, 7.5, 8, 8.5, 4
      , 6.5, 8.5, 8, 8, 8, 7, 8.5, 9, 8.5, 4, 5.5, 9, 9
      , 8.5, 9.5, 4, 9, 9, 8, 7.5, 6.5, 8, 7, 8, 9.5, 8
      , 8, 8, 6, 7.5, 9, 8, 9, 6, 8.5, 7, 9, 6, 8.5, 8.5
      , 9, 8, 6, 8, 7, 8.5, 7.5, 8, 7.5, 8, 8, 9, 9, 8.5
      , 8.5, 9.5, 9, 8.5, 0, 5.5, 8.5, 8, 7
    };

    Font f = new Font("Tahoma", 16);

    public Staafdiagram()
    {
        this.ClientSize = new Size(400, 230);
        this.Text = "Cijfers";
        this.Paint += Teken;
    }

    public void Teken(object o, PaintEventArgs pea)
    {
        // TODO
    }
}
```

