|  |  |
| --- | --- |
|  | **Datum**:  **Opleiding**:  **Lesmodule**:  **Test**: |
| **Resultaat**: **/ 30** | **Naam**: |

**Instructies:**

Deze test is een gesloten boek test. Gesloten computer

De puntenverdeling is gegeven, verspil niet al je tijd aan één opgave.

**Afspraak:**

Je hebt max. 100 min om de test op te lossen.

Veel succes.

Meerkeuzevragen ( /5):

* Je implementeert een methode berekenKorting die een korting berekent voor verkochte goederen. de klasse waarin de methode gedefinieerd staat een variabele percentWaarde van het type double. Je moet ervoor zorgen dat percentWaarde enkel toegankelijk is binnen berekenKorting methode. Welke access modifier zou je gebruiken voor de percentWaarde variabele?
* private
* protected
* public
* geen access modifier
* Stel dat je code schrijft voor een klasse met de naam Product. Je moet ervoor zorgen dat de instance variabelen van de klasse worden geïnitialiseerd naar hun juiste waarden zodra een object van de klasse Product wordt gemaakt. De initialisatie code moet altijd worden uitgevoerd. Wat moet je doen?
* Een static methode voorzien in de Product klasse om instance variabelen te initialiseren.
* initialiseren.
* Een static variabele voorzien in de Product klasse om instance variabelen te initialiseren.
* Een gewone methode voorzien in de Product klasse om de instance variabelen te initialiseren.
* Welk van de volgende beweringen is NIET waar?
* Een klasse is een concrete instantie van een object
* Een klasse definieert een blauwdruk voor een object
* Een static variabele is gemeenschappelijk voor de gehele klasse
* Een constructor wordt gebruikt om data members van het object te initialiseren
* Wat is de output van onderstaand codevoorbeeld?

int [] scores = {10,5,8,6,9};

System.*out*.println("Length: " + scores.length);

* Length: 4
* Length: 5
* Length: 6
* Compile error

* Wat is de output van onderstaande codevoorbeeld?

public class Developer {

public static void main(String[] args) {

boolean isDeveloper = false;

if (isDeveloper = true)

System.out.println("Je bent een goede developer");

else

System.out.println("Je bent een slechte developer");

}

}

* Je bent een goede developer wordt afgeprint
* Je bent een slechte developer wordt afgeprint
* Er wordt niets afgeprint
* Compile error

Vul aan: ( /10)

* Een klasse waarvan geen instanties aangemaakt kunnen worden moet gedeclareerd worden met het sleutelwoord
* Drie basis principes van object-georiënteerd programmeer talen zijn , \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_en .
* Je kan gebruik maken van \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ om gerelateerde klassen te groeperen om naam conflicten te vermijden.
* Het \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ sleutelwoord verwijst naar de huidige instantie van de klasse.
* Je kan gebruik maken van het \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ sleutelwoord om een variabele te declareren die behoort tot de klasse zelf ipv een specifieke object.
* Gegevens van een bepaald type forceren naar een ander type noemt men \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
* Met de access modifier “\_à\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_” heb ik toegang tot de eigen klasse, de subklassen en de pakketklassen, en niet tot andere klassen!
* Een \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_variabele krijgt automatisch de waarde 0, null of false mee wanneer deze niet geïnitialiseerd is.

Codevoorbeelden: ( /5)

**Wat is de uitvoer van volgende codevoorbeelden?**

1. **public** **static** **void** main(String[] args) {

**int** i = 0;

**int** j = 0;

**if** (i++ == ++j) {

System.*out*.println("True: i=" + i + ", j=" + j);

} **else** {

System.*out*.println("False: i=" + i + ", j=" + j);

}

}

Antwoord:

2. **public** **static** **void** main(String[] args) {

**float** f = 2.2f;

**int** i = 10;

**long** l;

f= i + 3;

f= f + l;

System.*out*.println(f);

}

Antwoord:

3. System.*out*.println(!!!!!**true**);

Antwoord:

4**. public** **class** ClassA {

**private** **int** size;

**public** ClassA(**int** size){

size = size;

}

**public** **static** **void** main(String[] args) {

ClassA a = **new** ClassA(5);

}

}

Wat is de waarde van de instance variabele **size** nadat de code is uitgevoerd?

Antwoord:

Codevoorbeeld ( /2)

**public** **class** Sbuild {

**public** **static** **void** main(String[] args) {

StringBuilder sb1 = **new** StringBuilder("Hello");

StringBuilder sb2 = sb1.append(" World");

System.***out***.println(sb1);

System.***out***.println(sb2);

System.***out***.println(sb1 == sb2);

}

}

Wat print de console uit als je die programma uitvoert?

Codevoorbeeld ( /2)

**public** **class** StringDaBuid {

**public** **static** **void** main(String[] args) {

StringBuilder sb1 = **new** StringBuilder("Alfred");

StringBuilder sb2 = sb1.append(" J. Kwak");

System.***out***.println(sb1);

}

}

Wat print de console uit als je die programma uitvoert?

Codevoorbeeld ( /2)

**public** **static** **void** main(String[] args) {

String s1 = "De Eend";

String s2 = "De Eend";

String s3 = s1;

s1 = **new** String("De Eend");

System.***out***.println(s3==s1);

System.***out***.println(s3==s2);

System.***out***.println(s1==s2);

}

}

Wat print de console uit als je dit programma uitvoert?

Codevoorbeeld ( /2)

**public** **class** StringenBuild {

**public** **static** **void** main(String[] args) {

String s1, s2;

StringBuilder sb1, sb2;

s1 = **new** String("Hello World");

sb1 = **new** StringBuilder("Hello World");

s2 = s1;

sb2 = sb1;

s1 = **new** String("Hello Mars");

sb1 = **new** StringBuilder("Hello Mars");

System.***out***.println(s2);

System.***out***.println(sb2);

}

}

Hoeveel objecten worden er aangemaakt in dit programma?

Wat print de console uit als je dit programma uitvoert ?

Codevoorbeeld ( /2)

**public** **class** StringTing2 {

**for**(**int** x = 0; x<2; x++){

String s1 = **new** String(" ");

String s2 = s1;

String s3;

}

}

Hoeveel objecten worden er aangemaakt als je dit programma uitvoert?