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

**Instructies:**

Deze test is een theoretische test, dit wil zeggen dat je de computer en je boek niet mag gebruiken.

* Meerkeuzevragen **/4**

1) Een abstracte klasse en een interface kunnen statische methoden bevatten: (1pt)

* Juist
* Fout

2) Wat wordt uitgeprint als je volgend stukje code uitvoert ? (het is uitvoerbaar ). (1pt)

String s = new String("Bicycle");

int iBegin = 1;

char iEnd = 3;

System.out.println(s.substring(iBegin, iEnd));

* Bic
* ic
* icy
* error

3) Wat is de output van volgende code: (1pt)

|  |
| --- |
| String kevin = "Kevin";  String copyKevin = **new** String("Kevin");  String copy2Kevin = kevin;  **if** (kevin == copyKevin)  System.*out*.println("EQUAL");  **else**  System.*out*.println("NOT EQUAL");  **if** (copyKevin == copy2Kevin)  System.*out*.println("EQUAL");  **else**  System.*out*.println("NOT EQUAL");  **if** (kevin == copy2Kevin)  System.*out*.println("EQUAL");  **else**  System.*out*.println("NOT EQUAL"); |

* EQUAL - EQUAL - EQUAL
* NOT EQUAL - NOT EQUAL - EQUAL
* EQUAL - EQUAL - NOT EQUAL
* NOT EQUAL - NOT EQUAL - NOT EQUAL

4) Als je naar onderstaande situatie kijkt: welke code werkt dan? (Meerdere antwoorden mogelijk) (1pt)

public abstract class Boek implements Voorwerp

public class WoordenBoek extends boek

* Boek boek = new Boek();
* WoordenBoek woordenBoek = new Boek();
* Boek boek = new WoordenBoek();
* Voorwerp voorwerp = new Boek();
* Voorwerp voorwerp = new WoordenBoek():
* WoordenBoek woordenBoek = new WoordenBoek();
* Open vragen **/12**

1) Ik wil 4 uitprinten. Hoe doe ik dit ? (2pt)

**int** [] [] array = **new** **int**[][]{{6,1,3},{9,8,7},{4,5},{2}};

2) Welke klassen hebben toegang tot een **protected** variabele? (2 pt.)

3) Leg uit wat een wrapper klasse is en waar ze worden gebruikt. ( 2 pt.)

4) Geef voordeel en een nadeel van het gebruik van het Collections frameWork? ( 2 pt.)

5) Wat is het verschil tussen een LinkedList en een HashSet? (Zo gedetailleerd mogelijk) (2pt)

6 Leg het verschil uit tussen CHECKED EN UNCHECKED Exceptions: (2pt)

* Code voorbeelden **/9**

1) Los volgende vragen op voor elk codevoorbeeld: (3pt)

* Wordt de code wel degelijk uitgevoerd of krijg je een error?
* Wat is de uitvoer van volgende codes? Welke error krijg je ?
* Moet de gebruiker iets invoeren?



2)Wat zal de output zijn als je de applicatie runt? ( 2pt)

|  |
| --- |
| **package** innerclasstesttokeep;  **public** **class** Mix {  **int** getal = 0;  Mix(){}  Mix(**int** i){  getal = i;  }    //inner class    **class** SubMix **extends** Mix {    SubMix(){  getal=Mix.**this**.getal\*2;  }  };  } |
| **package** innerclasstesttokeep;  **public** **class** MixApp {    **public** **static** **void** main(String[] args) {  Mix mix = **new** Mix(4);  Mix.SubMix sub = mix.**new** SubMix();  System.***out***.println(sub.getal);  System.***out***.println(mix.getal);  }  } |

3) Wat zal de output zijn als je de applicatie MixApp runt? (2pt.)

|  |
| --- |
| **package** newpak;  **public** **class** Mix {  **int** getal;  **public** Mix() {  getal = 45;  }  **public** Mix(**int** i) {  **this**.getal = getal \*2;  }  } |
| **package** newpak;  **public** **class** SubMix **extends** Mix{} |
| **package** newpak;  **public** **class** MixApp {    **public** **static** **void** main(String[] args) {    Mix submix = **new** SubMix(15);  }  } |

4) Wat is de output ? (2pt)

**public** **class** Person **implements** Comparator<Person> {

**public** String name;

**public** Person(String name) {

**this**.name = name;

}

@Override

**public** **int** compare(Person person1, Person person2) {

**return** person1.name.compareTo(person2.name);

}

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

List<Person> personen = **new** ArrayList<>();

personen.add(**new** Person("Luk"));

personen.add(**new** Person("Alfred"));

personen.add(**new** Person("Zorro"));

personen.add(**new** Person("Batman"));

**for** (Person person : personen) {

System.*out*.println(person.name);

}

}

}

* Kruiswoord :bijna gratis punten **/5**



* Je hebt 2 soorten programmeertalen: Gecompileerde en ………………………. Java probeert de voordelen van beide te verenigen.
* Deze operatoren voeren een bewerking uit op 3 operanden.
* Een complete programmaregel afgesloten met een “;”
* Het standaard toegangsniveau van een variabele/ constructor, als we dit niet expliciet toewijzen.
* “**private** **int** number = 10;” is een voorbeeld van een …………………………… variabele.
* Deze component moeten we gebruiken als we van onze klasse geen subklassing willen toelaten.
* Elk primitief datatype heeft ook een object variant. Hoe noemt dit?
* Fundamentele fouten (meervoud) die optreden in de Java Virtual Machine, deze mogen normaal niet voorkomen.
* Een ……………… klasse is een klasse waar we geen objecten van kunnen aanmaken.
* Een klasse die slechts een beperkt aantal instanties bevat.