

Test : C++ 3 - Vererbung

Benutzer : Treffler, Tanja

Standort : Hannover

Ergebnis : sehr gut - 96% (273 von 283 Pkt)

Benutzte Zeit : 00:49:50

Datum : 06/21/2019 11:14:09

1. Frage (ID 7 tf: Richtig oder Falsch)

Wenn der Programmierer keinen Konstruktor anlegt, legt der Compiler einen default-Konstruktor implizit an.

» **X Richtig** (ID 1)

X Falsch (ID 2)

Ergebnis : Richtig

Punkte : 3

2. Frage (ID 8 tf: Richtig oder Falsch)

Ein Konstruktor kann überladen werden.

» **X Richtig** (ID 1)

X Falsch (ID 2)

Ergebnis : Richtig

Punkte : 3

3. Frage (ID 9 tf: Richtig oder Falsch)

Wenn der Programmierer keinen Destruktor anlegt, legt der Compiler einen default-Destruktor implizit an.

» **X Richtig** (ID 1)

X Falsch (ID 2)

Ergebnis : Richtig

Punkte : 3

4. Frage (ID 10 tf: Richtig oder Falsch)

Ein Destruktor kann überladen werden.

X Richtig (ID 1)

» **X Falsch** (ID 2)

Ergebnis : Richtig

Punkte : 3

5. Frage (ID 15 tf: Richtig oder Falsch)

Funktionen sind überladen, wenn sie bei gleichem Bezeichner und gleichen Parametern nur unterschiedliche Eigenschaften der Klasse ändern.

☐ Richtig (ID 1)

» ☒ Falsch (ID 2)

Ergebnis : Richtig

Punkte : 3

6. Frage (ID 11 tf: Richtig oder Falsch)

Da der Aufruf von const-Member-Funktionen keine Eigenschaften verändern darf, dürfen const-Member-Funktionen auch wiederum nur const-Member-Funktionen aufrufen.

» ☒ Richtig (ID 1)

☐ Falsch (ID 2)

Ergebnis : Richtig

Punkte : 10

7. Frage (ID 5 mr: Multiple Choice - Multiple Answer)

Es bedeutet einen Unterschied, ob ich eine Methode einer Klasse innerhalb oder außerhalb der Klasse definiere?

» ☒ stimmt (ID 1)

☐ stimmt nicht (ID 4)

Ergebnis : Richtig

Punkte : 10

8. Frage (ID 12 tf: Richtig oder Falsch)

Funktionen sind überladen, wenn sie bei gleichem Bezeichner und gleichen Parametern nur unterschiedliche Rückgabetypen haben.

☐ Richtig (ID 1)

» ☒ Falsch (ID 2)

Ergebnis : Richtig

Punkte : 3

9. Frage (ID 13 tf: Richtig oder Falsch)

Funktionen sind überladen, wenn sie bei gleichem Bezeichner unterschiedliche Parametertypen haben.

» ☒ Richtig (ID 1)

☐ Falsch (ID 2)

Ergebnis : Richtig

Punkte : 3

10. Frage (ID 19 tf: Richtig oder Falsch)

Den Funktionsoperator () kann man grundsätzlich überladen.

» **X Richtig** (ID 1)

X Falsch (ID 2)

Ergebnis : Richtig

Punkte : 3

11. Frage (ID 21 tf: Richtig oder Falsch)

Den Gültigkeitsbereichsoperator :: kann man grundsätzlich überladen.

X Richtig (ID 1)

» **X Falsch** (ID 2)

Ergebnis : Richtig

Punkte : 3

12. Frage (ID 17 tf: Richtig oder Falsch)

Den Punktoperator . kann man grundsätzlich überladen.

X Richtig (ID 1)

» **X Falsch** (ID 2)

Ergebnis : Richtig

Punkte : 3

13. Frage (ID 20 tf: Richtig oder Falsch)

Den Indexoperator [] kann man grundsätzlich überladen.

» **X Richtig** (ID 1)

X Falsch (ID 2)

Ergebnis : Richtig

Punkte : 3

14. Frage (ID 16 tf: Richtig oder Falsch)

Der Zuweisungsoperator = wird als in der Basisklasse überladener Operator an die abgeleitete Klasse vererbt.

X Richtig (ID 1)

» **X Falsch** (ID 2)

Ergebnis : Richtig

Punkte : 10

15. Frage (ID 27 tf: Richtig oder Falsch)

"<<" ist ein überladener Operator in der Klasse ostream.

» **X Richtig** (ID 1)

X Falsch (ID 2)

Ergebnis : Richtig

Punkte : 10

16. Frage (ID 6 mr: Multiple Choice - Multiple Answer)

```
class Box
{
    int x, y, z;
public:
    Box(int a, int b, int c): x(a), y(b), z(c) {}
    draw();
};
```

Eigenschaften der Objekte von Klassen können initialisiert werden ...

» **X ... als Strukturobjekte außerhalb der Klasse. Box myBox = {0x00, 0x80, 0xc0};** (ID 2)

» **X ... durch ein anderes Objekt dieser Klasse mittels Kopier-Konstruktor.** (ID 3)

» **X ... durch Anweisungen in einem Konstruktor der zugehörigen Klasse.** (ID 1)

X ... bei Einsatz des Schlüsselwortes static vor Angabe des Typs des Elementes. (ID 4)

Ergebnis : Richtig

Punkte : 30

17. Frage (ID 22 tf: Richtig oder Falsch)

```
class KlasseA {
    protected: int wert;
};
class KlasseB {
    protected: int wert;
};
class KlasseC : public KlasseA, public KlasseB {
    public: void setwert(int value) {
        wert=value;
    }
};
```

Gleichnamige Member in den Basisklassen können bei einer Mehrfachvererbung in der abgeleiteten Klasse problemlos genutzt werden.

X Richtig (ID 1)

» **X Falsch** (ID 2)

Ergebnis : Richtig

Punkte : 10

18. Frage (ID 24 tf: Richtig oder Falsch)

Die Member-Funktionen einer Klasse werden auch als Methoden der Klasse bezeichnet.

» ☒ **Richtig** (ID 1)

☐ **Falsch** (ID 2)

Ergebnis : Richtig

Punkte : 10

19. Frage (ID 26 mr: Multiple Choice - Multiple Answer)

Wann benötige ich einen "zweistufigen" Zugriff auf Member?

Objekt_A.Objekt_B.Member();

☐ Bei ungekapselten Klassen (ID 2)

☐ bei Eigenschaften (Elementen) von Klassen (ID 4)

☐ Bei vererbten Klassen (ID 3)

» ☒ **Bei eingeschlossenen Klassen** (ID 1)

Ergebnis : Richtig

Punkte : 10

20. Frage (ID 18 tf: Richtig oder Falsch)

Enthält die Klasse, für die der Operator = überladen wird, dynamische Eigenschaften, so ist eine Zuweisung auf sich selbst immer abzufangen.

```
myWin = myWin;
```

» ☒ **Richtig** (ID 1)

☐ **Falsch** (ID 2)

Ergebnis : Richtig

Punkte : 10

21. Frage (ID 25 mr: Multiple Choice - Multiple Answer)

Betrachten Sie den Kopf der folgenden Klassendeklaration:

```
class Tasse : public Behaelter::Geschirr {.....};
```

Wofür ist der Bezeichner vor dem Operator :: notwendig?

☐ Geschirr ist abgeleitet von der Klasse Behaelter. (ID 2)

☐ Geschirr und Behaelter sind Basisklassen einer Mehrfachvererbung. (ID 4)

» ☒ **Geschirr steht im Namensraum Behaelter.** (ID 1)

☐ Geschirr wurde von Behaelter vererbt. (ID 3)

Ergebnis : Richtig

Punkte : 10

22. Frage (ID 46 tf: Richtig oder Falsch)

Der Zugriff auf Eigenschaften der Basisklasse ist in abgeleiteten Klassen möglich, wenn die Zugriffsrechte auf public: gesetzt wurden.

» **X Richtig** (ID 1)

X Falsch (ID 2)

Ergebnis : Richtig

Punkte : 10

23. Frage (ID 47 tf: Richtig oder Falsch)

Der Zugriff auf Eigenschaften der Basisklasse ist in abgeleiteten Klassen möglich, wenn die Zugriffsrechte auf protected: gesetzt wurden.

» **X Richtig** (ID 1)

X Falsch (ID 2)

Ergebnis : Richtig

Punkte : 10

24. Frage (ID 48 tf: Richtig oder Falsch)

Der Zugriff auf Eigenschaften der Basisklasse ist in abgeleiteten Klassen möglich, wenn die Zugriffsrechte auf private: gesetzt wurden.

X Richtig (ID 1)

» **X Falsch** (ID 2)

Ergebnis : Richtig

Punkte : 10

25. Frage (ID 49 tf: Richtig oder Falsch)

Der Zugriff auf Eigenschaften der Basisklasse ist in abgeleiteten Klassen möglich, wenn die Zugriffsrechte gar nicht gesetzt wurden.

X Richtig (ID 1)

» X Falsch (ID 2)

Ergebnis : Falsch

Punkte : 0

26. Frage (ID 38 tf: Richtig oder Falsch)

Man sollte eine Klasse vererben, wenn die Unterklasse zu den eigenen Methoden, die der abgeleiteten Klasse mit benutzen soll.

» **X Richtig** (ID 1)

X Falsch (ID 2)

Ergebnis : Richtig

Punkte : 10

27. Frage (ID 39 tf: Richtig oder Falsch)

Man sollte eine Klasse vererben, wenn die Unterklasse zu den eigenen Elementen, die der vererbenden Klasse mit benutzen soll.

» **X Richtig** (ID 1)

☐ Falsch (ID 2)

Ergebnis : Richtig

Punkte : 10

28. Frage (ID 40 tf: Richtig oder Falsch)

Man sollte eine Klasse vererben, wenn die Unterklasse einem völlig neuen, eigenständigen Projekt zugeordnet werden soll.

☐ Richtig (ID 1)

» **X Falsch** (ID 2)

Ergebnis : Richtig

Punkte : 10

29. Frage (ID 41 tf: Richtig oder Falsch)

Man sollte eine Klasse vererben, wenn die Unterklasse keine eigenen Elemente und Methoden enthält.

☐ Richtig (ID 1)

» **X Falsch** (ID 2)

Ergebnis : Richtig

Punkte : 10

30. Frage (ID 31 mr: Multiple Choice - Multiple Answer)

Was muss der Präprozessor-Anweisung `#if` (.....) in einer späteren Zeile unbedingt folgen?

☐ `#then` (ID 4)

☐ `#end` (ID 3)

☐ `#else` (ID 2)

» **X `#endif`** (ID 1)

Ergebnis : Richtig

Punkte : 10

31. Frage (ID 42 tf: Richtig oder Falsch)

Die Präprozessor-Anweisung `#include <cmath>` steht sinnvoll nur bei konkretem, dortigen Bedarf in der h-Datei.

» **X Richtig** (ID 1)

☐ Falsch (ID 2)

Ergebnis : Richtig

Punkte : 10

32. Frage (ID 43 tf: Richtig oder Falsch)

Die Präprozessor-Anweisung `#include <cmath>` steht sinnvoll nur bei konkretem, dortigen Bedarf in der `cpp`-Datei.

» **X Richtig** (ID 1)

X Falsch (ID 2)

Ergebnis : Richtig

Punkte : 10

33. Frage (ID 44 tf: Richtig oder Falsch)

Die Präprozessor-Anweisung `#include <cmath>` steht sinnvoll immer in der `cpp`-Datei.

X Richtig (ID 1)

» **X Falsch** (ID 2)

Ergebnis : Richtig

Punkte : 10

34. Frage (ID 45 tf: Richtig oder Falsch)

Die Präprozessor-Anweisung `#include <cmath>` steht sinnvoll immer in der `h`-Datei.

X Richtig (ID 1)

» **X Falsch** (ID 2)

Ergebnis : Richtig

Punkte : 10