**9. Aufgabenblatt**

|  |  |  |  |
| --- | --- | --- | --- |
| **Vorname** | **Nachname** | **Matrikelnummer** | **Fachrichtung** |
| Philipp | Huber | 63326 | Mabb |
| Marius | Grumer | 63284 | Mabb |

|  |  |
| --- | --- |
| **Punkte** | xx/yy |
| **Bestanden** | ja/nein |

*1. Aufgabe:*

Eine „Hat-Beziehung“ („has a“) entsteht zwischen zwei verschiedenen Klassen, wenn

1. eine Klasse von der anderen Klasse abgeleitet ist
2. ein Datenelement einer Klasse vom Typ der anderen Klasse ist
3. eine Klasse innerhalb der anderen Klasse definiert ist

**Lösung**: b

*Hinweis: Bei der Komposition hat eine Klasse eine oder mehrere andere Klassen. Klasse Auto „hat ein“ (Klasse) Lenkrad.*

*2. Aufgabe:*

Eine abgeleitete Klasse erbt eine public-Methode der Basisklasse nicht, wenn sie selbst eine public-Methode mit gleichem Namen besitzt

1. wahr
2. falsch

**Lösung**: a

*Hinweis: Die Methode der Elternklasse existiert nicht mehr im Bereich der Kinderklasse, wenn diese überschrieben wurde. Es kann aber auf die Elemente der Elternklasse mit dem Bereichsoperator :: zugegriffen werden.*

*CA: Elternklasse*

*myB: Objekt der Kindklasse CB*

*void fkt(): Funktion der Eltern- und Kindklasse*

myB.CA::fkt(); //Zugriff auf fkt() im Bereich der Elternklasse

*3. Aufgabe:*

In einer abgeleiteten Klasse

1. können nur die Methoden der Basisklasse redefiniert (überschrieben) werden
2. können nur die Datenelemente der Basisklasse redefiniert (überschrieben) werden
3. kann jedes Element der Basisklasse redefiniert (überschrieben) werden

**Lösung**: c

*Hinweis: Alle Elemente der Elternklasse können in der Kindklasse überschrieben werden.*

*4. Aufgabe:*

Durch public-Vererbung ist die Klasse Z aus Y und die Klasse Y aus der Klasse X entstanden. Die Klasse X enthält eine public-Methode calc(), die in der Klasse Y aber nicht in der Klasse Z als public-Methode redefiniert (überschrieben) ist. Beim Aufruf der Methode calc() für ein Objekt der Klasse Z wird dann

1. die Methode calc() der Klasse X aufgerufen
2. die Methode calc() der Klasse Y aufgerufen
3. der Compiler eine Fehlermeldung ausgeben, da die Methode in der Klasse Z nicht redefiniert ist

**Lösung**: b

*Hinweis: Z erbt von Y, daher wird diese Funktion aufgerufen.*

*5. Aufgabe:*

Wenn eine Methode der Basisklasse in einer abgeleiteten Klasse redefiniert (überschrieben) wird, müssen die Signaturen der beiden Methoden übereinstimmen.

1. wahr
2. falsch

**Lösung**: a

*Hinweis: stimmen die Signaturen nicht überein wird eine neue Methode erstellt oder es findet eine Überladung der Methode statt.*

*6. Aufgabe:*

Beim Erzeugen eines Objektes einer abgeleiteten Klasse

1. wird der Konstruktor der abgeleiteten Klasse zuerst ausgeführt
2. wird der Konstruktor der Basisklasse zuerst ausgeführt
3. ist nicht definiert, welcher Konstruktor zuerst ausgeführt wird

**Lösung**: b

*Hinweis: zuerst wird der Konstruktor der Basisklasse ausgeführt, diese wird auch als letztes zerstört.*

*7. Aufgabe:*

Wenn ein Konstruktor einer abgeleiteten Klasse mit Parametern definiert wird, können Basisinitialisierer und Elementinitialisierer in einer Liste durch Kommas getrennt angegeben werden.

1. wahr
2. falsch

**Lösung**: a

*Hinweis: es ist möglich den nötigen Basisinitialisierer in der Liste zu initialisieren (Ausprobiert).*

class Eltern

{

public:

Eltern(int i) :mData(i) { ; } //Elementinitalisierung

~Eltern();

protected:

int mData;

};

class Kind :public Eltern

{

public:

Kind(int a, int b) :x(a), y(b), Eltern(5) { ; }

//Elementinitialisierung mit Basisinitialisierer

~Kind();

private:

int x, y;

};

*8. Aufgabe:*

Wenn kein Konstruktor für eine abgeleitete Klasse definiert ist, muss in der Basisklasse ein Default-Konstruktor vorhanden sein.

1. wahr
2. falsch

**Lösung**: a

*Hinweis: Damit die abgeleitete Klasse erstellt werden kann, wird ein Konstruktor benötigt. Ist dieser nicht in der Klasse selbst definiert, muss in der Elternklasse ein Basiskonstruktor definiert sein.*

*9. Aufgabe:*

Um allen Methoden einer abgeleiteten Klasse den direkten Zugriff auf die geschützten Elemente der Basisklasse zu ermöglichen, können diese Elemente in der Basisklasse als ................................. deklariert werden.

**Lösung**: protected

*Hinweis*: *durch das Verwenden von protected anstatt private, können Methoden der Kinderklassen auf die dort definierten Elemente zugreifen. Ein anderweitiger Zugriff (z.B. von Objekten) ist nicht möglich.*

*10. Aufgabe:*

Der Zugriff auf ein protected-Element einer Klasse X ist möglich

1. über jedes Objekt der Klasse X
2. innerhalb jeder Methode einer von X abgeleiteten Klasse
3. über jedes Objekt einer von X abgeleiteten Klasse

**Lösung**: b

*Hinweis*: *Der Zugriff ist nur innerhalb von Methoden möglich. Durch protected wird dieser erweitert auch auf Methoden abgeleiteter Klassen.*