

Objektorientierte Programmierung (Prog2) Probeklausur SoS 2023

Aufg. Nr.	Text	Pkt
1.	<p>a) Was ist die vollständige Überdeckung einer Klasse?</p> <p>b) Was ist der Unterschied zwischen einer konkreten Klasse und einer abstrakten Klasse?</p> <p>c) Was charakterisiert die 3 Stufen des objektorientierten Vorgehens nach Wegner? Stellen Sie 6 Verbindungslinien her, so dass korrekte Aussagen zusammenpassen:</p> <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="width: 45%;"> <p>Objekt-basiertes Vorgehen <input type="radio"/></p> <p>Klassen-basiertes Vorgehen <input type="radio"/></p> <p><u>Objekt-orientiertes Vorgehen</u> <input type="radio"/></p> </div> <div style="width: 50%;"> <p><input type="radio"/> Geheimnisprinzip</p> <p><input type="radio"/> <u>Polymorphismus</u></p> <p><input type="radio"/> Klassenbildung</p> <p><input type="radio"/> Daten- und Funktionskapsel</p> <p><input type="radio"/> Abstraktion + Instanziierung</p> <p><input type="radio"/> <u>Vererbung</u></p> </div> </div>	6
2.	<p>Multiple Choice Aufgaben, es können keine, eine oder mehrere Aussagen der Auswahlen richtig sein. Kreuzen Sie nur die richtigen Aussagen an.</p> <p>a) Wegner-Stufen:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Stufe 2 nach Wegner ist durch die Begriffe Vererbung und Polymorphismus bezeichnet. <input type="checkbox"/> Das Geheimnisprinzip ist, dass alle Methoden der Schnittstelle keinen Rückgabewert liefern. <input checked="" type="checkbox"/> Die „Kapsel“ bedeutet, dass nur Mitgliedsfunktionen von Objekten auf die Mitgliedsvariablen zugreifen. <input checked="" type="checkbox"/> Die „Kapsel“ bedeutet, dass nur Methoden von Objekten auf die Attribute zugreifen. <input checked="" type="checkbox"/> Durch die Abstraktion zu Klassen aus Wegner Stufe 2 ist die Instanziierung von Objekten möglich. <input type="checkbox"/> Keine der Aussagen trifft zu. <p>b) Referenzen:</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Eine Referenz ist ein Alias-Name auf ein Objekt und bleibt ständig mit ihm verbunden. <input type="checkbox"/> Die Einführung von Referenzen führt unmittelbar zu endlosen Rekursionen. <input checked="" type="checkbox"/> Die Einführung von Referenzen reduziert den Laufzeitaufwand für Methodenaufrufe. <input type="checkbox"/> Werden Attributswerte in einer Referenz auf ein Objekt verändert, so wird damit auch das Objekt verändert. <input checked="" type="checkbox"/> Gibt eine Funktion eine Referenz auf ein Objekt zurück, dann muss dieses Objekt über das Verlassen der Funktion hinaus bestehen. <input type="checkbox"/> Keine der Aussagen trifft zu. 	6

3.	<p>Zeigen Sie den Unterschied zwischen</p> <p>a) Überladen und</p> <p>b) Überschreiben</p> <p>von Funktionen, indem Sie jeweils ein Beispiel erstellen. Verwenden Sie für Ihre Beispiele folgende Klasse als Ansatz:</p> <pre> class cls1 { double y; public: void func1 (int i) { y = i*1.19; } }; </pre>	6
4.	<p>a) Finden Sie mindestens 8 Fehler im folgenden Programm und Beschreiben Sie jeden mit wenigen Worten.</p> <p>b) Beschreiben Sie die Funktion des Programms (ignorieren Sie dabei die Fehler).</p> <pre> #include <iostream> using namespace std; class cSchiff { public: virtual void anlegen () == 0; }; class cTanker : protected Schiff { public: virtual void anlegen () { cout << "Tanker legt an mit Hilfe von Schleppern" << endl; } }; class cUboot : public cBoot { public: virtual void anlegen (cSchiff& s) { cout << "Uboot taucht auf und legt dann an" endl; } } int main () { cSchiff * sp; cTanker t1; cUboot u1; sp = &t1; sp->anlegen(); sp = &u1; sp.anlegen(); return 0; } </pre>	12

5. Programmieraufgabe am Rechner. Verwenden Sie MS Visual Studio und laden Sie das **Projektverzeichnis** und die **sln-Datei** als Zip-Archiv hoch. Alternativ können Sie auch eine einzelne **cpp-Datei** in einem Zip-Archiv hochladen. Bitte beachten Sie die Anleitung für die Programmieraufgabe im Anhang der Klausur.

40

Zur Vervielfältigung von Zeichen bei der Ausgabe („klonieren“) soll Folgendes programmiert werden:

Erstellen Sie eine Klasse **cZeichenKlon** mit den notwendigen Mitgliedern:

Attribute (privat):

- **zeichen** zur **Angabe des zu klonenden Zeichens**
- **anzahlKlone** zur Angabe der Anzahl der Klone (Wiederholungen) des Zeichens (beim Wert 0 in **anzahlKlone** wird das Zeichen nur einmal ausgegeben)

Methoden:

- Konstruktor (Vorgabewerte '*' für den Buchstaben und 0 für die Anzahl)
- Ausgabemethode „**ausgabe()**“, die die beiden Attribute ausgibt (dient zur Kontrolle, ob der überladene Ausgabeoperator funktioniert)
- Überladung des Inkrement-Operators ++
- Überladung des Dekrement-Operators --
- Überladung des Ausgabeoperators <<
- ggf. private Hilfsmethoden

Der Inkrement-Operator ++ erhöht die Anzahl der Klone um 1. Implementieren Sie nur die rechtsseitige Stellung des Operators (Postfix, es muss also der Wert vor der Erhöhung zurückgegeben werden).

Der Inkrement-Operator -- erniedrigt die Anzahl der Klone um 1. Implementieren Sie nur die rechtsseitige Stellung des Operators (ebenfalls Postfix). Beachten Sie die Vermeidung des Null-Unterlaufes.

Der Ausgabeoperator << soll folgende Funktionalität besitzen: Das Zeichen **zeichen** wird ausgegeben und danach **anzahlKlone** mal wiederholt.

Erstellen Sie ein Hauptprogramm:

- Definieren Sie eine Instanz k1 der Klasse **cZeichenKlon**.
- Geben Sie die Werte der Attribute durch Aufruf der Methode **ausgabe()** aus.
- Geben Sie das Objekt k1 danach mit dem überladenen <<-Operator aus.
- Führen Sie die Operation **k1++** dreimal aus.
- Geben Sie das Objekt k1 danach mit dem überladenen <<-Operator erneut aus.
- Führen Sie die Operation **k1--** aus.
- Geben Sie das Objekt k1 mit dem überladenen <<-Operator erneut aus.

Bewertungsschema:

Punktzahl

- Programm lässt sich ohne Fehler/Warnungen übersetzen	5P
- Codeformatierung und sinnvolle Kommentare	3P
- Klassendefinition und Konstruktor	4P
- Ausgabemethode ausgabe()	3P
- Überladung des Inkrement-Operators ++	5P
- Überladung des Dekrement-Operators --	5P
- Überladung des Ausgabeoperators <<	10P
- Hauptprogramm	5P

Summe: 40P

Handskizzen können mit bis zu 15 Punkten berücksichtigt werden. (max. Gesamtpunktzahl 40P)

Summe

70

Zeitschema: 60 Minuten, 70 Punkte. Bestanden mit 32 Punkten; Note 1,0 ab 64 Punkte