



Magdeburg, 17.07.2012

Klausur „Grundlagen der Informatik“

Matrikelnummer:	
Name:	Vorname:
Studiengang: WMB / MB / MTK / PH (wöchentliche V/Ü im SoSe) (zutreffenden Studiengang unterstreichen)	
Anzahl Zusatzblätter:	
Unterschrift Student/in:	Unterschrift Aufsicht:

bitte **nicht** ausfüllen:

Aufg. 1: einfaches C++- Programm	Aufg. 2: statische Felder	Aufg. 3: OOP	Aufg. 4: Fragenblock WiSe	Aufg. 5: Fragenblock SoSe	Summe
(von 12)	(von 23)	(von 15)	(von 25)	(von 25)	(von 100)

Hinweise für die Klausurbearbeitung

- Überprüfen Sie die Klausur auf Vollständigkeit (18 Seiten / 9 Blätter). Füllen Sie das Klausurdeckblatt gewissenhaft mit Namen, Vornamen und Studiengang aus. Schreiben Sie auf jedes Blatt Ihren Namen.
- Legen Sie alle für die Klausur benötigten Dinge, insbesondere Lichtbildausweis (Studentenausweis zur Anwesenheitskontrolle), Stifte, Verpflegung auf Ihren Tisch. Schalten Sie Ihre Handys aus.
- Verwenden Sie für Ihre Antworten den freien Platz nach den Aufgaben und ggf. die Rückseiten der Blätter. Melden Sie sich, wenn Sie zusätzliche leere Blätter benötigen.
- Die Papierklammerung der Klausur darf **nicht** gelöst werden!
- Benutzung roter / grüner Stiftfarbe und Bleistift ist untersagt.
- Die Benutzung unerlaubter Hilfsmittel (z. B. Taschenrechner, Bücher, Folien zur Vorlesung, Mobiltelefone, Spickzettel) gilt als Täuschungsversuch und führt zu einer Bewertung der Prüfung mit „nicht ausreichend“ (5,0).
- Schreiben Sie deutlich! Unleserliche Passagen können nicht korrigiert werden.
- Beschränken Sie sich auf die geforderten Angaben und halten Sie Ihre Antworten kurz und präzise. Nicht geforderte Angaben ergeben keine zusätzlichen Punkte.

Viel Erfolg!

Name: _____

Aufgabe 1: einfaches C++-Programm (12 Punkte)

Bei diesem Programm sollen zunächst Werte eingelesen und anschließend ausgegeben werden. Dies soll dann noch einmal für alle Arten der Wiederholungsanweisungen programmiert werden. Gehen Sie dabei in folgenden Schritten vor.

- a) Schreiben Sie die Funktion (1,5 P)
`void read(char &c, double &d, int &i),`
die drei Variablen vom Typ *double*, *char* und *int* per Tastatur einlesen kann.
- b) Die Funktion (1 P)
`void output(char c, double d, int i)`
soll alle drei Variablen auf dem Bildschirm mit einem Hinweistext in der Form
Alle drei Werte: ..., ..., ...
ausgeben.
- c) In der main- Funktion sollen die drei Variablen mit der Funktion *read* eingelesen und mit der Funktion *output* auf dem Bildschirm ausgegeben werden. (2 P)
- d) Wiederholen sie das Einlesen und Ausgeben der drei Variablen 5mal. Zeigen Sie dies für jede der drei Arten von Wiederholungsanweisungen (Schleifen). (7,5 P)

Name: _____

Name: _____

Aufgabe 2: Statische Felder (23 Punkte)

Ihre Aufgabe ist es, ein Feld zu erzeugen, welches mit Zufallszahlen gefüllt wird. Die Werte im Feld sollen ausgegeben werden und eine Berechnung des Minimums, des Maximums und der Summe aller Feldelemente soll erfolgen.

- a) In der Funktion (3 P)
`void initRandom(int anz, int a[])`
sollen allen *anz* Elementen aus dem integer-Feld zufällige ganze Zahlen aus [0,...,99] (`rand() % 100`) zugewiesen werden.
- b) Die Funktion (2 P)
`void show(int anz, int a[])`
soll alle *anz* Elemente von Feld *a* auf dem Bildschirm ausgeben.
- c) Die Funktion (6,5 P)
`void calculate(int anz, int a[], int &min, int &max, int &sum)`
soll aus den Elementen von Feld *a*, das kleinste (*min*) Element, das größte (*max*) Element und die Summe (*sum*) aller Elemente berechnen.
- d) In der *main*- Funktion soll (11,5 P)
- ein statisches integer-Feld vereinbart werden, dass Platz für maximal 100 Elemente hat. Die aktuelle Feldgröße *anz* soll per Tastatur eingegeben werden. Überprüfen Sie, ob *anz* zulässig ist und wiederholen Sie gegebenenfalls die Eingabe von *anz*.
 - Danach sollen die Elemente des Feldes durch Aufruf der Funktion *initRandom* belegt und mit *show* auf dem Bildschirm ausgegeben werden.
 - Zum Schluss sollen noch die Berechnungen mit Hilfe der Funktion *calculate* ausgeführt werden. Alle drei Rechenergebnisse sind auf dem Bildschirm auszugeben.

Name: _____

Name: _____

Aufgabe 3: OOP (15 Punkte)

Bei dieser Aufgabe wird eine Klasse *Fahrrad* erzeugt, auf welche anschließend verschiedene Methoden zugreifen.

- a) Definieren Sie eine Klasse *Fahrrad* mit den Attributen (2,5 P)
- hersteller* (Datentyp *string*),
 - rg* (Datentyp *int* – Rahmengröße in Zoll),
 - modelljahr* (Datentyp *int*)
- b) Schreiben Sie innerhalb der Klasse (3,5 P)
- einem Standard-Konstruktor und
 - einem Konstruktor mit 3 Parametern (für die drei Attribute *hersteller*, *rg*, *modelljahr*)
- c) Schreiben Sie innerhalb dieser Klasse zwei Methoden (3,5 P)
- *void read()*
zum Einlesen der Daten von der Tastatur (ohne Kontrolle) und
 - *void show()*
zum Anzeigen der Daten auf den Bildschirm.
- d) Erzeugen Sie in der *main*-Funktion **zwei** Objekte: (2,5 P)
- *fahrrad1* mit dem Konstruktor, der die drei Attribute setzt und
 - *fahrrad2* mit dem Standard-Konstruktor.
 - Überschreiben Sie die Daten in *fahrrad1* mittels der Methode *read* und
 - zeigen Sie die Daten der beiden Fahrräder mit der Methode *show* an.
- e) (3 P)
- Legen Sie nun in der *main*-Funktion ein **dynamisches** Feld *fahrradStaender* an, in dem 5 Fahrräder Platz finden. Alle 5 Fahrräder sollen dabei die Werte des Standard-Konstruktors erhalten.
 - Geben Sie zur Kontrolle die Daten aller 5 Fahrräder aus.

Name: _____

Name: _____

Aufgabe 4: Fragenblock Wintersemester (25P)

- a) Wie werden die Zahlen 7, 8 und 9 im Dualsystem dargestellt? Was ist die besondere Bedeutung des Dualsystems in der Informatik? (3P)
- b) Die Darstellung von positiven und negativen Zahlen mit Nachkommastellen erfolgt durch eine Aufteilung von mehreren Bytes in 3 Bereiche. Nennen Sie diese Bereiche und erläutern Sie kurz deren Funktion bei der Darstellung. (3P)
- c) Was ist bei der Ausführung von Programmen der Unterschied von einem Compiler zu einem Interpreter? Nennen Sie wenigstens zwei Nachteile des Interpreteransatzes. (4P)
- d) Erläutern Sie die 3 Klauseln im Kopf einer for-Schleife. (3P)

Name: _____

Name: _____

Aufgabe 4

- e) Die Ausführung von Schleifen kann im Schleifen-Code durch die Befehle *break* und *continue* gesteuert werden. Erläutern Sie die Funktion der beiden Befehle. (2P)
- f) Was ist der Unterschied der Übergabe von Parametern an eine Funktion als Wert (Pass/Call by Value) bzw. als Referenz (Pass/Call by Reference)? (4P)
- g) Was ist der Unterschied zwischen abweisenden und nicht-abweisenden Schleifen in den Programmiersprachen C/C++? Nennen Sie für jeden Schleifentyp je eine Steueranweisung aus C/C++ (4P)
- h) Welcher Unterschied besteht in C++ zwischen statischen Feldern (als Variablen auf dem Stack) und dynamischen Feldern (mit Operator *new* erzeugt auf dem Heap)? (2P)

Name: _____

Name: _____

Aufgabe 5 : Fragenblock Sommersemester (25P)

- a) Was versteht man in der Programmierung unter *Rekursion*? (2P)
- b) Erläutern Sie das objektorientierte Konzept der Spezialisierung/Vererbung am Beispiel der Klassen *Person* (mit den Eigenschaften *Name* und *Alter*) und Studenten (mit den Eigenschaften *Matrikelnummer* und *Studiengang*). (2P)
- c) Was ist in einer C++-Klasse die Aufgaben von *Konstruktoren*? (2P)
- d) Erläutern Sie im Sinne der Computergrafik kurz den Begriff der *Bildsynthese*. (2P)

Name: _____

Name: _____

Aufgabe 5

e) Nennen Sie 4 grafische Primitive, die in OpenGL unterstützt werden. (2P)

f) Welche zwei Operationen sind grundlegend für die Arbeit mit der Datenstruktur *Stack* (Stapelspeicher)? Beschreiben Sie kurz das Ergebnis und die Veränderung, die diese Operationen am *Stack* bewirken. (4P)

g) Wie werden entsprechend dem *Relationenmodell* Daten dargestellt? Was sind Schlüssel und Fremdschlüssel und wozu werden sie verwendet? (6P)

Name: _____

Name: _____

Aufgabe 5

h) Wahr oder falsch?

(5P, wenigstens 0 Punkte)

Kreuzen Sie jeweils (ohne Begründung!) an, ob die Aussage “*wahr*“ oder “*falsch*“ ist. Für jede richtige Antwort gibt es einen halben Punkt, für jede falsche Antwort wird ein halber Punkt abgezogen. Beantworten Sie also lieber nur Fragen, bei denen Sie sich sicher sind!

	wahr	falsch
1. Die Bestandteile der zentralen Verarbeitungseinheit (CPU) der Von-Neumann-Architektur sind der Speicher und der Systembus.		
2. Die Abkürzung RAM als Hauptspeicher steht für Random-Access Memory.		
3. Mit dem Zweierkomplement können negative und positive ganze Zahlen binärkodiert werden.		
4. Der ASCII Code ist eine Kodierungsvorschrift für reelle Zahlen.		
5. Das Hexadezimalsystem verwendet ausschließlich die Ziffern 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C und D.		
6. Der Präprozessor untersucht Quelltext vor dem Übersetzen auf semantische Korrektheit.		
7. Für die Deklaration einer Funktion muss keine Implementierung (Code) angegeben werden.		
8. Als <i>protected</i> definierte Attribute können in der Klasse selbst und in abgeleiteten Klassen zugegriffen werden.		
9. Ein terminierender Algorithmus liefert für jede Eingabe nach einer endlichen Anzahl von Berechnungsschritten ein Ergebnis.		
10. Die Laufzeitkomplexität der Berechnung des Maximums in einem unsortierten Feld der Größe n ist $O(n)$.		

Name: _____

Name: _____