

Klausur C++ #1

Jahr: 2001; Dozent: Dipl.Ing. Sorber

1) Welche Antworten zur Operatorüberladung sind richtig? (1)

- a) C++ - Operatoren können zusammen mit Objekten funktionieren
- b) C++ - Operatoren wird mehr aufgeladen, als diese handhaben können.
- c) C++ - Operatoren wird neue Bedeutung verliehen

Antwort: a,c

2) Wieviele Argumente sind in der Definition eines unären überladenen Operators erforderlich? (1)

Antwort: In der Definition eines unären überladenen Operators sind keine Argumente erforderlich.

3) Nennen Sie zwei Operatoren in C++ die nicht überladen werden dürfen. (1)

Antwort:

- . Elementzugriff
- .* Zugriff auf Elementzeiger
- sizeof Size Off Operator
- ?: Bedingungsoperator
- :: Scope-Operator

4) Erläutern Sie die Bedeutung des this-Zeigers. (2)

Antwort: Bei jedem Aufruf einer Methode zeigt der this-Zeiger auf das zu nutzende Objekt bzw. auf die zu nutzenden Daten.

5) Realisieren Sie für eine Klasse Preis mit den privaten Daten

unsigned long int mark;

unsigned short pfennig;

die folgenden Operatorüberladungen als Methoden der Klasse. Gehen Sie davon aus, dass vor der jeweiligen Operation für den pfennig-Anteil gilt: $0 \leq \text{pfennig} < 100$.

a) + -Operator zur Addition zweier Preis-Instanzen (Mark und Pfennig) (5)

b) ++ -Operator als Präfixoperator zur Erhöhung des pfennig-Anteils um 1 (5)

Antwort:

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
class Preis
```

```
{private:
```

```
unsigned long int mark;
```

```
unsigned long int pfennig;
```

public:

void Eingabe ()

```
{cout<<endl<<"Eingabe in Mark und Pfennig: ";  
cin>>mark>>pfennig;  
}
```

Preis operator + (Preis a)

```
{Preis erg;  
erg.mark=mark+a.mark;  
erg.pfennig=pfennig+a.pfennig;
```

```
if (erg.pfennig>=100)
```

```
{erg.mark++;  
erg.pfennig-=100;  
}  
return erg;  
}
```

void Ausgabe ()

```
{cout<<endl<<"Addition ergibt: "<<mark<<" Mark und "<<pfennig<<" Pfennig!";}  
/*
```

void operator ++ ()

```
{pfennig++;  
if (pfennig>=100)  
{mark++;  
pfennig-=100;  
}  
}
```

void Anzeige ()

```
{cout<<endl<<"Nach Erhoehung des Pfenniganteils: "<<mark<<" Mark und "  
cout<<pfennig<<" Pfennig!";}  
*/  
};
```

int main ()

```
{Preis p1, p2, p3;  
cout<<"P1: ";  
p1.Eingabe ();
```

```
cout<<endl;
```

```
cout<<"P2: ";  
p2.Eingabe ();
```

```
cout<<endl;
```

```
p3=p1+p2;  
p3.Ausgabe ();
```

```
// p1++;  
// cout<<endl<<endl<<"P1: ";  
// p1.Anzeige ();  
// p2++;  
// cout<<endl<<endl<<"P2: ";  
// p2.Anzeige ();
```

```
getch ();  
}
```

Die auskommentierten Teile sind Aufgabe b.

6) Was verstehen Sie unter einer Konstruktorliste? Schreiben Sie eine Konstruktorliste für die Datenelemente der folgenden Klasse auf. (3)

```
class Beispiel
{ unsigned int variable 1;
  float variable 2;
  long double variable 3;
  public:
    .....
};
```

Antwort:

Konstruktorliste:

```
Beispiel :: Beispiel ( ) : variable 1 (2), variable 2 (4.6), variable 3 (8.9) { }
```

7) Es ist der + - Operator für die Verknüpfung von Zeichenketten zu überladen. Schreiben Sie eine Klasse string, die als Datenelement einen char-Vektor für maximal 100 Zeichen enthält, sowie eine entsprechende Operatorfunktion, die zwei Zeichenketten zu einer dritten in der Form $s_3 = s_1 + s_2$; miteinander verknüpfen kann. Prüfen Sie außerdem, ob die Länge der beiden zu verknüpfenden Zeichenketten den Wert 100 nicht übersteigt. Geben Sie eine entsprechende Meldung aus. (6)

Antwort:

```
#include <iostream.h>
#include <conio.h>
#include <string.h>

class String
{ private:
  char vektor [100]; //Vektor mit 100 Elementen

  public:
  String () //Konstruktor
  {strcpy (vektor, "");} //Kopieren des Vektors

  String (char x[]) //Konstruktor
  {strcpy (vektor,x);} //Kopieren des Vektors x in Vektor

  String operator + (String str) //Operatorfunktion (mit neuem Element)
  {if (strlen (vektor)+strlen(str.vektor)<100) //Prüfen auf Stringlänge (siehe Anweisung)
  {String temp; //temporäres Objekt wird erstellt
  strcpy (temp.vektor, vektor); //Kopierfunktion (vektor nach temp.vektor)
  strcat (temp.vektor, str.vektor); //Anhängen zweier Strings
  return temp; //Rückgabe von temp
  }
}
```

```

else
  cout<<"Summenvektor zu gross!";      //Ausgabe für übergroße Stringlänge
}
void Ausgabe ()                        //Ausgabefunktion
{cout<<vektor;}

};

int main ()
{String s1("Du bist");                  //1. String
String s2(" wirklich nett!");          //2. String
String s3;                              //3. String

cout<<"1. String: "<<endl;
s1.Ausgabe ();
cout<<endl<<endl;

cout<<"2. String: "<<endl;
s2.Ausgabe ();
cout<<endl<<endl;

s3=s1+s2;                               //Strings aneinanderhängen
cout<<"Ergebnis"<<endl;
s3.Ausgabe ();

getch ();
}

```

8) Ermitteln Sie von den angegebenen Programmen erzielte Bildschirmausgabe. (4)

a)

```

#include <iostream.h>
#include <conio.h>

int main ()
{int x=0;
for (int i=1; i<10; i*=2)
{x++;
cout<<endl<<x;
}
cout<<"\nx= "<<x;
getch ();
}

```

Antwort:

[leere Zeile]

1
2
3
4
x=4

b)

```
#include <iostream.h>
#include <conio.h>
```

```
int main ()
{int x=0;
for (int i=1;i<10;++i)
{if (i%2==1)
x+=i;

else
x--;

cout<<endl<<x;
}

cout<<endl<<"\nx= "<<x;
getch ();
}
```

Antwort:

[leere Zeile]

1

0

3

2

7

6

13

12

21

[leere Zeile]

x=21