

Informationstechnik

Übungsblatt 05

Institut für Technik der Informationsverarbeitung, Karlsruher Institut für Technologie

Zu Übung05, Besprechung: 21.06.2012 14⁰⁰ – Neue Chemie

Aufgabe 5.01: Verständnisfragen

- a) Die Methoden einer abgeleiteten Klasse können auf die private-Elemente der Basisklasse zugreifen. Richtig / Falsch
- b) Eine abgeleitete Klasse erbt eine public-Methode der Basisklasse nicht, wenn sie selbst eine public-Methode mit gleichem Namen besitzt. Richtig / Falsch
- c) Polymorphe Klassen werden in C++ mit Hilfe von _____ Methoden implementiert.
- d) Bei der Ausführung der Anweisungen: `string name; cin >> name;` werden Zeichen von der Standardeingabe eingelesen und zwar
 - i. alle Zeichen einer Zeile ohne führende Zwischenraumzeichen
 - ii. genau ein Wort ohne führende Zwischenraumzeichen
 - iii. eine ganz Textzeile
- e) Zum Verketteten zweier Objekte vom Typ `string` kann man den Operator _____ verwenden.
- f) Das erste Zeichen in einem String hat die Position _____.
- g) Für den Zugriff auf die einzelnen Zeichen in einem `string`-Objekt kann der Operator _____ verwendet werden.
- h) Nennen Sie einen Vorteil und einen Nachteil einer verketteten Liste gegenüber einem dynamisch angelegten Array.
- i) Beim Einfügen und Löschen eines Elements in einer verketteten Liste werden keine Elemente verschoben, sondern lediglich _____ versetzt.

Aufgabe 5.02: OOP, Klassendefinition und Vererbung

- a) Angenommen die folgenden Klassen sind in der Header-Datei "myClasses.h" definiert:

```
class Polynom {
    int x;
public:
    Polynom();
    virtual int calc( int a );
};
```

```
class Add : public Polynom {
    int y;
public:
    Add();
    int calc();
};
```

Für ein Objekt `obj` der Klasse `Add` ist dann folgende Anweisung zulässig:

```
int res = obj.calc( 2 );
```

Richtig / Falsch

- b) Was gibt das folgende Programm auf dem Bildschirm aus?

```
#include <iostream>

using namespace std;

class B {
public:
    B() { cout << "Konstruktor der Klasse B \n"; }
    ~B() { cout << "Destruktor der Klasse B \n"; }
};

class D : public B {
public:
    D() { cout << "Konstruktor der Klasse D \n"; }
    ~D() { cout << "Destruktor der Klasse D \n"; }
};

class X {
private:
    D d;
public:
    X() { cout << "Konstruktor der Klasse X \n"; }
    ~X() { cout << "Destruktor der Klasse X \n"; }
};

int main() {
    X xObj;
    cout << "Bye, bye!" << endl;
    return 0;
}
```

Aufgabe 5.03: Stringmanipulation

Schreiben Sie ein Programm, welches in einen String alle Vorkommen eines (anderen) Strings **findstring** sucht und die jeweiligen Positionen auf dem Bildschirm ausgibt. In nächsten Schritt soll dieser String durch einen anderen **replaceString** ersetzt werden. Die Länge der Strings können dabei beliebig (auch unterschiedlich) sein. Wenn **findstring** nicht im String gefunden wird soll eine Fehlermeldung ausgegeben werden. Tipp: Sie können zur Lösung der Aufgabe die folgenden Manipulator-Funktionen verwenden:

a) Die Methode **find()** liefert die Position an der ein Teilstring zu ersten Mal in einem String vorkommt. Als Argumente erhält Sie den gesuchten Teilstring und als optionales zweites Argument die Startposition zur Suche. Als Rückgabewert liefert die Methode das erste Auffinden des gesuchten Strings.

Beispiel:

```
string kaffee = "Der Kaffee ist kalt, sehr kalt";  
int pos = kaffee.find( "kalt", 20 ); //pos = 26
```

b) Die Methode **replace()** kann einen String durch einen anderen ersetzen, dabei dürfen die Strings auch eine unterschiedliche Länge haben. Sie erhält als erstes Argument die Startposition und als Zweites die Länge des zu ersetzenden Strings. Das dritte Argument ist der Ersatz-String.

Beispiel:

```
kaffee.replace( 4, 6, "Tee" );  
//Ergebnis: "Der Tee ist kalt, sehr kalt"
```

Beispielausgabe zum Programm:



```
C:\WINDOWS\system32\cmd.exe
Text eingeben : Mein roter Koffer und meine rote Tasche
findstring    : rot
replacestring  : blau

String rot gefunden an Pos: 5
String rot gefunden an Pos: 28
Ergebnis: Mein blauer Koffer und meine blaue Tasche
Drücken Sie eine beliebige Taste . . . _
```