



Name:

Klasse:

07.02.08

IB No.:

Pointer

oder

der Unterschied zwischen dem Inhalt und der Adresse einer Variablen

Sinn und Zweck:

Anschauliche Vorstellung:

Eine Bank oder auch ein Bahnhof hat Schließfächer zum Ablegen von Sachen.

Ein Schließfach hat zwei Eigenschaften, *Eigentümer* und *Nummer*. In einem Schließfach kann man Sachen ablegen. Wenn man an den *Inhalt* des Schließfaches will braucht man mindestens Eigentümer oder Nummer.

Computer-Realität:

Jede Variable hat einen **Namen** und eine Speicher**adresse**. Die **Adresse** gibt an, wo im Speicher diese Variable ist.

Der Wert einer Variablen steht genau an dieser Stelle. Zum Speichern dieser Adressen gibt es **Pointer-Variablen**.

MEKRE:

Der WERT einer Variablen ist nicht dasselbe wie die ADRESSE einer Variablen.

Syntax in C:

```
//          man spricht „Referenz deklarieren“ oder „Pointer deklarieren“
n
//Pointer auf eine Variable

Datentyp* Name_der_Pointer-Variablen;

//          man spricht „einen Pointer de-referenzieren“

//Inhalt einer Pointer-Variable mit dem * Operator

*Name_der_Pointer-Variablen;

//          man spricht „die Adresse einer Variablen verwenden“

//Adresse einer beliebigen Variablen mit dem & Operator

&Name_der_Variablen;
```

BEACHTEN:

Der Datentyp kann bspw. sein: void, float, int, char, double, byte, etc.

Beispiele:

I)

Aufgabe: Deklariere Variable „pMeinZeigerAufFloat“, die einen Pointer auf float enthält

Lösung:

```
float* pMeinZeigerAufFloat;          //beachte das * hinter dem Datentyp;
```

II)

Aufgabe: Deklariere eine Pointer-Variable „pEins“, die eine Adresse auf int speichern kann. Deklariere eine weitere Variable „iTest“ vom Datentyp int. Speichere die Adresse von iTest in pEins.

Lösung:

```
int* pEins;
int iTest;
```

```
pEins = &iTest;          // beachte auf der linken Seite steht eine Pointer-Variable
                        // auf der rechten Seite wird das & benutzt (für Adresse)
```