

Notizen AIK ProgTech 3: new und delete

Klaus Kusche

Funktion

- **new** *erzeugt ein neues Objekt* oder ein *Array von Werten / Objekten* im Speicher.
- **new** liefert einen *Pointer* auf das neu angelegte Objekt / Array.
- **delete** *gibt* ein mit **new** angelegtes Objekt / Array wieder *frei*.

Varianten

- **myClass *p;**
p = new myClass;

Es wird *ein* neues **myClass-Objekt** angelegt und mit dem *Standard-Konstruktor* initialisiert.

- **p = new myClass(x, y, 0);**

Es wird *ein* neues **myClass-Objekt** angelegt und mit dem *angegebenen Konstruktor-Aufruf* initialisiert.

- **p = new myClass[arrSize];**

Es wird ein *Array* mit **arrSize** vielen **myClass-Objekten** angelegt. Für jedes Objekt im Array wird der *Standard-Konstruktor* aufgerufen (der Aufruf eines anderen Konstruktors ist bei Arrays *nicht* möglich).

arrSize kann eine *Konstante oder Variable* sein, d.h. die benötigte Größe des Arrays kann zur Laufzeit ausgerechnet werden.

Dynamisch große *mehrdimensionale* Arrays sind *nicht* möglich.

- **int *ip;**
ip = new int[arrSize];

Es wird ein *Array* mit **arrSize** vielen **int-Elementen** angelegt. Die Elemente bleiben *uninitialisiert* (zufällige Werte).

```
myClass **pp;  
pp = new myClass* [100];
```

Analog für ein *Array* von 100 *Pointern* (auch *uninitialisiert!*).

- **delete p;**

Das *einzelne Objekt*, auf das **p** zeigt, wird *freigegeben*, der *Destruktor* wird ausgeführt.

p darf auch ein **NULL-Pointer** sein: Dann macht **delete p;** nichts.

- **delete [] p;**

Das *Array*, auf das **p** zeigt, wird *freigegeben*. Falls es ein Array von Objekten ist, wird für jedes Element der *Destruktor* aufgerufen.