

Funktionen und Prozeduren

Definition:

- „Teil einer Software“
- Kann an verschiedenen Stellen aufgerufen werden
- Liefert ein Ergebnis zurück

Funktionen und Prozeduren

- Kann beliebig oft aufgerufen werden
- Sprung und Rücksprung geschieht automatisch
- Eindeutiger Funktionsname für den Aufruf
- Aufrufer ist nur noch verantwortlich für die richtigen Parameter und „das Abholen“ des Ergebnisses
- Ergebnistyp legt fest, was die Funktion als Ergebnis zurückgibt

Funktionen und Prozeduren

Ergebnistyp Funktionsname (Parameterliste)

```
{ /* Funktionsrumpf */ }
```

Beispiel:

```
float quadrat (float x)
```

```
{return x * x;};
```

Funktionen und Prozeduren

Aufruf einer Funktion:

```
float p = 1.2;
```

```
float q = quadrat (p);
```

Aktuelle Parameter werden auf den Stack kopiert

Funktionen und Prozeduren

Funktionsparameter:

- Beim Aufruf werden formale Parameter durch aktuelle „ersetzt“
- „Call by Value“ (Konstante, z.B. `int m = min(5,10)`)
- „Call by Name“ (Variable, z.B. `int m1 = 5`)
- „Call by Reference“ (Pointer)

Funktionen und Prozeduren

- Werden grundsätzlich auf dem Stack übergeben
- Analog zu Variablen
- Beim beenden wird der Bereich wieder freigegeben

Funktionen und Prozeduren

Rückgabewerte:

- Als Konstante (z.B. return 7)
- Als Variable (int, float, char, ...)
- Als Ausdruck (z.B. return m+5)

Es wird immer ein Wert zurückgegeben und nicht die Variable oder die Konstante!!!

Funktionen und Prozeduren

Im Unterschied zur Funktion liefert eine Prozedur keinen Rückgabewert auf direktem Weg!

Wird leer übergeben, hat keinen Funktionsparameter (void)

Funktionen und Prozeduren

Aufgabe:

Schreiben Sie eine Funktion, die 3 Buchstaben nach dem Alphabet sortiert!

Funktionen und Prozeduren

Lösung: