

Merkblatt zu Dynamischen Datenstrukturen

Referenzen

- Deklaration: `datentyp &ref_name = var_name;`
- Dereferenzierung: `cout << ref_name;`

Zeigervariablen

- Deklaration: `datentyp *ptr_name = &var_name;`
- Dereferenzierung: `cout << *ptr_name;`
- Änderung: `ptr_name = &var_name;`

Zeigervariablen und Felder

- Deklaration eines Feldes: `datentyp feld_name[anzahl];`
- Zeiger auf ein Feld: `datentyp *ptr_name = feld_name;`
`datentyp *ptr_name = &feld_name[0];`
- Zeiger auf ein Feldelement i: `ptr_name = feld_name + i;`
`ptr_name = &feld_name[i];`

Speicherverwaltung für elementare Datentypen

- Speicher erzeugen: `datentyp *mem_ptr = new dententyp;`
- Speicher verwenden: `*mem_ptr = wert;`
`cout << *mem_ptr;`
- Speicher freigeben: `delete mem_ptr;`

Speicherverwaltung für Felder

- Speicherfeld erzeugen: `datentyp *mem_ptr = new dententyp[anzahl];`
- Speicherfeld verwenden: `mem_ptr[i] = wert;`
`cout << *(mem_ptr+i);`
- Speicherfeld freigeben: `delete[] mem_ptr;`

Zeiger und Referenzen als Return-Wert von Funktionen

- Zeiger als Return-Wert: `datentyp* fkt_name(argumente){`
`return &var_name;`
`}`
- Referenz als Return-Wert: `datentyp& ftk_name(argumente){`
`return var_name;`
`}`