Spickzettel C / C++

Ausgabe:

```
clrscr();
cout << "Text" << endl;
```

Eingabe:

```
cin >> nZahl;
chZeichen = getch();
chZeichen = getche();
gets(chSatz);
```

Kontrollstrukturen:

a) Verzweigungen

```
if(Bedingung)
{
   Anweisungen;
}
else
{
   Anweisungen;
}
```

```
switch(Ausdruck)
{
   case Wert1:
      Anweisungen;
      break;
   case Wert2:
      Anweisungen;
      break;
   default:
      Anweisungen;
}
```

b) Schleifen

```
for(i=iMin; i<=iMax ; i++)
{
   Anweisungen;
}
```

```
do
{
   Anweisungen;
}while(Bedingung);
```

```
while(Bedingung)
{
   Anweisungen;
}
```

Speicherplatz:

```
nAnzahl = sizeof(Variable);
```
Anzahl der reservierten Bytes

Mathematik:

```
fErgebnis = sqrt(fZahl);
fErgebnis = pow(a,b);
```
Quadratwurzel

a^b

Zufallszahlen:

```
randomize();
nZufall = rand()%(nMax–nMin+1) + nMin
```
Zufallsgenerator "mischen"

Zufallszahl von nMin bis nMax

Farben:

```
window(1,1,80,25);
textcolor(YELLOW + BLINK);
textbackground(RED);
clrscr();
```
kompletter Bildschrm

Schriftfarbe

Hintergrundfarbe

Farben übertragen

Koordinaten:

```
gotoxy(2,3);
int x = wherex();
int y = wherey();
```
2. Spalte, 3. Zeile im aktuellen window

aktuelle Spalte

aktuelle Zeile

Adressoperator &

```
double fVariable;
```

Adresse von fVariable: `&fVariable`

Zeiger:

Deklaration eines Zeigers auf double: `double* pZeiger;`
auf fVariable zeigen lassen: `pZeiger = &fVariable;`
auf Inhalt von fVariable zugreifen: `*pZeiger = 1.2;`

Funktionen ohne Rückgabewert

a) ohne Parameter

```
//Prototyp
void funktion();

// Aufruf
funktion();

// Definition
void funktion()
{
      ...
}
```

b) mit Parametern

```
//Prototyp
void funktion(int, double);

// Aufruf
funktion(3, 1.2);

// Definition
void funktion(int n1, double f2)
{
      ...
}
```

Funktionen mit Rückgabewert

a) ohne Parameter

```
//Prototyp
double funktion();

// Aufruf
double f = funktion();

// Definition
double funktion()
{
      double fR;
      ...
      return fR;
}
```

b) mit Parametern

```
//Prototyp
double funktion(int, double);

// Aufruf
double f = funktion(3, 1.2);

// Definition
double funktion(int n1, double f2)
{
      double fR;
      ...
      return fR;
}
```