Hochschule Deggendorf Dr. Peter Jüttner	
Vorlesung: Objektorientierte Programmierung	SS 2019
Übung 3	Termin 4.4.19

Klassen

1. **Aufgabe: Ergänzung der Klasse Complex um selbstgeschriebene Konstruktoren und Destruktoren**

   a.) Ergänzen Sie Ihre Klasse Complex um einen parameterlosen Konstruktor, der
   - So wohl Real- aus auch Imaginärteil mit 0.0 initialisiert
   - eine zu vereinbarende globale Variable um 1 erhöht

```
unsigned int zahl_complex = 0;

…

complex()
 { real = 0.0;
   imag = 0.0;
   zahl_complex++;
   printf("parameterloser Konstruktor aufgerufen\n");
   printf("Anzahl Complex Objekte: %d\n",zahl_complex);
 };
```

   b.) Ergänzen Sie Ihre Klasse Complex um einen weiteren Konstruktor, der zwei float-Parameter r und i besitzt. Der Konstruktor soll
   - den Realteil mit r und den Imaginärteil mit i initialisieren.
   - und die oben erwähnte Variable um 1 erhöhen.

```
complex(float r, float i)
{ real = r;
  imag = i;
  zahl_complex++;
  printf("Konstruktor mit 2 Parametern aufgerufen\n");
 printf("Anzahl Complex Objekte: %d\n",zahl_complex);
 };
```

   c.) Erstellen Sie den Standarddestruktor so, dass er die oben erwähnte globale Variable um 1 reduziert.

```
~complex()
 { zahl_complex--;
   printf("Destruktor aufgerufen\n");
   printf("Anzahl Complex Objekte: %d\n",zahl_complex);
 };
```

d.) Initialisieren Sie die globale Variable, die die Objekte zählt bei Programmstart mit 0, geben Sie die Variable nach Anlegen oder Löschen eines Objekts aus.

e.) Ergänzen Sie Ihr main, so dass Objekte auf dem Stack und dem Heap angelegt werden.

```
…
complex c1(); // parameterlos
complex c2(4.0, 5.0); // mit Parametern
complex *cp1 = new complex; // parameterlos
complex *cp3 = new complex();// parameterlos
complex *cp3 = new complex(3.0, 6.0); // mit Parametern
…
delete cp1;
delete cp2;
delete cp3;
```

2. **Führen Sie die Bearbeitung der Aufgabe Fuhrparkverwaltung vom letzten Mal weiter.**