

## C++ Übungen Blatt 1

### 1.1) Die Klasse „Rectangle“

Erstellen Sie ein neues Projekt mit dem Titel „Rectangle“. Das Programm soll zunächst leer sein.

Implementieren Sie die Klasse „Rectangle“ mit ihren Attributen und Methoden, bestehend aus Header- und Implementierungsfile.

Die Klasse repräsentiert ein Rechteck und ist fähig, den Flächeninhalt des Rechtecks zurück zu geben. Sie hält Attribute für die x/y-Position sowie Länge und Breite des Rechtecks vor. Ebenso sind die folgenden Zugriffsmethoden zu implementieren:

- void SetPos(double x, double y)
- void SetDim(double width, double height)
- double GetArea() // Gibt den Flächeninhalt des Rechtecks zurück

Programmieren Sie ein Hauptprogramm zur Klasse „Rectangle“. Das Programm soll x-Position, y-Position, Breite und Höhe eines Rechtecks vom Benutzer abfragen und die gesamte Information zusammen mit dem Flächeninhalt des Rechtecks auf dem Bildschirm ausgeben.

### 1.2) Zeiger und Objekte

Verwenden Sie die Klasse „Rectangle“ von Teil 1 in einem neuen Projekt „Zeiger“ und schreiben Sie dazu ein kleines Hauptprogramm, das von dieser Klasse ein Objekt „rect“, einen Zeiger „pRect1“ und ein dynamisch allokiertes Objekt „pRect2“ erzeugt.

Lassen Sie den Zeiger „pRect1“ auf das Objekt „rect“ zeigen. Anschließend lassen Sie den Zeiger „pRect2“ auf die gleiche Adresse wie „pRect1“ zeigen. Geben Sie nach jeder Zuweisung die Adresse für „rect“, sowie die Adressen, auf die „pRect1“ und „pRect2“ zeigen, aus. Was passiert mit dem Objekt auf dem Heap?

### 1.3) „delete“: Rückgabe an die Speicherverwaltung

Erzeugen Sie noch einmal von der Klasse „Rectangle“ ein Objekt „rect“ und einen Zeiger „pRect1“.

Allokieren Sie für den Zeiger „pRect1“ auf dem Heap ein Objekt der Klasse „Rectangle“. Geben Sie den für „pRect1“ auf dem Heap allokierten Speicher wieder frei.

Lassen Sie „pRect1“ auf das Objekt „rect“ zeigen. Versuchen Sie nun wieder, den Speicher, auf den „pRect1“ zeigt, freizugeben. Was geschieht beim Übersetzen/beim Ablauf des Programms?

### 1.4) Zugriff auf Objekte über Zeiger

Setzen Sie in dem obigen Programm bei beiden Instanzen beliebige Werte für Position und Dimension ein. Geben Sie für beide Objekte den Flächeninhalt aus.

### 1.5) Array von Zeigern auf Objekte

Programmieren Sie eine Klasse „RectField“ mit einem Feld von Zeigern des Typs „Rectangle“ (definieren sie dieses Feld zunächst als ‚public‘). Die Größe des Feldes soll maximal 50 Einträge betragen. Implementieren Sie für die Klasse einen Konstruktor, dem Sie die Anzahl der zu instanzierenden Feldelemente übergeben.

Schreiben Sie ein Hauptprogramm dazu, das eine Instanz einer solchen Klasse mit 25 Einträgen anlegt und diese Einträge mit Werten für Position und Dimension belegt. Was passiert, wenn Sie in diesem Hauptprogramm beide Header-Files „Rectangle.h“ und „RectField.h“ inkludieren? Welche Gegenmaßnahmen bieten sich an?

### **1.6) Auslesen der Werte des Feldes**

Lesen im gleichen Programm für alle Instanzen des Feldes den Flächeninhalt aus und stellen Sie ihn am Bildschirm dar. Was geschieht, wenn Sie den 26. Wert des Arrays lesen?

### **1.7) Verbesserungen an den beiden Klassen**

Nehmen Sie an den beiden Klassen die folgenden Verbesserungen vor:

- Geben Sie der Klasse „Rectangle“ einen Konstruktor, mit dem Sie alle Werte übergeben können
- Setzen Sie im Default Konstruktor der Klasse „Rectangle“ alle Attribute auf einen sinnvoll definierten Wert
- Definieren Sie das Array der Klasse „RectField“ als private. (Kapselung)
- Programmieren Sie zum Zugriff die beiden Methoden
  - void SetRect (int index, double x, double y, double width, double height)
  - double GetArea (int index)
- Bauen Sie Prüfungen ein,
  - ob der Index im erlaubten Bereich des Feldes liegt
  - ob der Zeiger unter dem angegebenen Index gültig ist