

C++ Übungen Blatt 2

2.1) Aufrufmethoden

Erstellen Sie ein leeres Projekt mit Namen „RectCopy“. Importieren Sie alle notwendigen Dateien zur Verwendung der Klasse RectField. Erweitern Sie diese Klasse um eine Methode „print()“, die Position, Dimension und Flächeninhalt für alle gültigen Rechtecke ausgibt.

Definieren und Implementieren Sie für die Klasse „RectField“ eine Methode „AddByValue“, die dem Array ein neues Rechteck-Objekt hinzufügt. Der Methode soll ein komplettes Objekt übergeben werden, das an das Ende des bestehenden Arrays angehängt wird. Testen Sie die Methode und geben Sie das erweiterte Feld von Rechtecken auf dem Bildschirm aus.

Eine weitere Methode „AddByPointer“ soll die Prozedur nach dem Verfahren „call by pointer“ durchführen. Definieren, implementieren und testen Sie auch diese Methode.

Eine dritte Variante soll den gleichen Zweck erfüllen unter Verwendung des Methodenaufrufs „call by reference“. In der Methode soll der übergebene Wert für die Position des Rechtecks verändert werden, bevor das Rechteck dem Array hinzugefügt wird. Geben Sie den Wert der übergebenen Position nach dem Methodenaufruf im Hauptprogramm aus. Hat sich der Wert geändert? Ist das Verhalten wie erwartet?

2.2) Objekte kopieren

Erstellen Sie ein Objekt der Klasse RectField auf dem Heap, das 10 Rechtecke speichert, und belegen Sie Position und Dimension dieser Rechtecke mit Werten.

Erstellen Sie jeweils eine Kopie der Instanz von RectField

1. durch Verwendung des Copy Konstruktors
2. durch Verwendung des Assign Operators

Stellen Sie fest, was in den beiden Kopien passiert, wenn Sie in der Original-Instanz Werte für Position und/oder Dimension ändern.

Implementieren Sie einen Copy Konstruktor und einen Assignment Operator für die Klasse RectField, um sog. „tiefes Kopieren“ zu ermöglichen. Was passiert nun mit den Kopien, wenn Sie am Original Werteänderungen vornehmen?