Aufgaben

1. Gehen Sie die einzelnen Beispiele der Folien durch und probieren Sie das eine oder andere Programm in Visual Studio selbst aus. Setzen Sie Break Points, starten Sie die Programme und werfen Sie während der Laufzeit jeweils einen Blick auf den Typ und den Inhalt der Variablen. Experimentieren Sie etwas mit den Programmen, Visual Studio und dem Debugger.
   Versuchen Sie sich zu erklären, warum etwas so ist, wie es ist.
   Lesen Sie dazu jeweils auch in den angegebenen Quellen nach (färbiger Kasten jeweils rechts auf jeder Folie), um Ihre beim Testen gemachten Erfahrungen auch erklären zu können.
   Zudem bieten die Quellen weit mehr Anwendungsbeispiele und Informationen, als wir im Unterricht aus zeitlichen Gründen durchmachen könnten.

2. Erstellen Sie unter Visual Studio für jedes Aufgabenblatt (jede Abgabe) eine eigene Solution, in die Sie jeweils die Programme (Projekte) einfügen. Die Solution sollte die Bezeichnung für
   Teamabgaben: OOP21_<Gruppe>_[AufgabenNr]_[Nachname1]_[Nachname2]
   zB: OOP21_A_04_Mustermann_Musterfrau

   Die Namen der Projekte werden bei den einzelnen Unteraufgaben jeweils angegeben.

Hinweis: Eine sinnvolle Erweiterung und Adaption der Anforderung Ihrerseits ist durchaus erwünscht und erlaubt.

3. Umbau **Kontoverwaltung** (Erzeugen Sie eine Kopie des Projekts Kontoverwaltung aus LB03):

   a. Die Klasse **Konto** soll einen Konstruktor bekommen, der Kontoinhaber (Name) und auch den Kontostand (Betrag) aufnimmt. Jedes Konto soll eine eindeutige Kontonummer (ID) haben. Diese wird im Konstruktor gesetzt, jedoch NICHT als Parameter übergeben. **(2)**
      Hinweis: static

b. Wird ein Konto kopiert, so soll es eine neue Kontonummer bekommen und der Kontostand auf 0 gesetzt werden. Es darf nur der Kontoinhaber kopiert werden. **(2)**
Hinweis: static, copy constructor

c. Implementieren Sie eine Methode zum Schließen eines Kontos. Nachdem ein Konto geschlossen wurde, darf keine Transaktion auf dieses Konto mehr möglich sein. Prüfen Sie dabei auch, ob der Kontostand zum Zeitpunkt des Schließens tatsächlich genullt ist. **(2 Bonuspunkte)**

d. Fügen Sie in Ihrer Klasse **Konto** bei jeder Methode, die das Konto manipuliert, ein **Logging** (zumindest mit Zeitstempel, Text) ein. Das Logging sollte für alle Konten gleich auf einen *std::ostream* erfolgen, der definiert werden kann. **(2 Bonuspunkte)**

e. Führen Sie eine Methode zum Überweisen eines Betrags von einem Konto zu einem anderen Konto ein. Zudem sollten alle Transaktionen mit einem zusätzlichen Text versehen werden können. **(2)**

f. Das endgültige Löschen eines Kontoobjekts sollte auch immer geloggt werden. **(2)**
Hinweis: destructor

g. Testen Sie ihren Code in main(). Erzeugen Sie dafür einige Kontos und mindestens 200 Transaktionen welche zufällig zwischen den Kontos durchgeführt werden. Testen Sie die weiteren Funktionen (a-f) ausreichend.
Schreiben Sie alle Änderungen der Aufgaben a-f in die Klasse **Konto**. **(2)**

Abgabe

- Für jedes Team muss nur eine einzige Abgabe erfolgen! Aus dem Datennamen können ja, wenn er korrekt ist, auch die Teammitglieder ermittelt werden.

- Klicken Sie unter Visual Studio mit der **rechten Maustaste im Solution Explorer** auf den Solutionname (erste Zeile). Wählen Sie den Menüpunkt **Clean Solution** aus. Damit sollen (fast) alle Compilate und temporären Dateien gelöscht sein.
Wechseln Sie mit dem Windows Explorer in das Solution-Verzeichnis und löschen Sie dort auch das .vs Verzeichnis (Achtung: hidden folder!) und alle x64 bzw. Debug-Verzeichnisse in den jeweiligen Projektunterverzeichnissen.

  Achtung: Liefern Sie aber etwaige Datendateien der Programme mit, damit man beim Testen auch gleich einen Datenbestand hat.

- Die Abgabe erfolgt im Moodle unter Assignment 04
Beachten Sie, dass es für jede Gruppe einen bestimmten Upload Link gibt.

- bis spätesten **vor Beginn der nächsten Laboreinheit**