

Aufgaben

1. Solution und Projekt unter Visual Studio anlegen:

- a. Erstellen Sie unter Visual Studio eine leere Solution mit dem Namen **OOP_LB_WH_Nachname.Vorname**.
- b. Erstellen sie im Projektmappenexplorer dieser Solution zwei Projekte, **Sporttracker** und **Diverses**.

2. Sporttracker

Erstellen Sie eine Sporttracker Software

- a. Erstellen sie die Klasse **Laufeinheit**, mit einer **gelaufenen Distanz** sowie einer **Trainingszeit** Variable. Achten Sie auf die Kapselung, diese Member sollen nur von der Klasse oder Ableitungen dieser Klasse zugänglich sein. Implementieren Sie daher eine Möglichkeit diese Variablen zu setzen. Verwenden Sie einen **geeigneten Container** um diese Laufeinheiten zu verwalten und **begründen** Sie ihre Auswahl in einem Satz.
- b. Erweitern Sie nun die Laufeinheit Klasse insofern, das mehrere Laufeinheiten über den **+** **Operator** addiert werden können (Zeit und Distanz werden addiert). Überladen Sie des Weiteren die **<<** und **>>** **Operatoren** um **Ein-** und **Ausgabe** zu ermöglichen (Format: *<Zeit> <Distanz>*).
- c. Leiten Sie von der Laufeinheiten Klasse nun die Klassen **Intervalltraining** (zusätzlicher Member: AnzahlWiederholungen) und **Ausdauertraining** (Gegenwind *<true/false>*) ab. Erstellen sie einen **Sporttracker Container**, welcher **Laufeinheiten**, **Intervalltraining** und **Ausdauertraining** beinhalten kann. Hierfür soll, unter anderem, der für diesen Fall Ressourcen sparende **Smart Pointer** verwendet werden.
- d. Ermöglichen Sie die Ausgabe *Trainingseinheit <TYP> <DISTANZ> <ZEIT> <ZUSÄTZLICHER PARAMETER>* über die Funktion **print** für alle drei erstellten Typen wobei der zusätzliche Parameter das Zusatzfeld von Intervall- und Ausdauertraining ist.
- e. Testen Sie den Code ausreichend und verwenden Sie jede implementierte Funktion mindestens einmal.

3. Diverses

- a. Verwenden Sie ein **Containertemplate**, um **Shared Pointer** auf **Integer** abzulegen (Vorsicht, über diesen Container wird später **iteriert**).
- b. Finden sie die **größten drei geraden integer** (im Container von 3a) über ein **Lambda** und geben sie diese **außerhalb** des Lambdas, **sortiert** aus
- c. Erstellen Sie eine von `std::exception` erbende **Exception** welche beim Aufruf der **what()** Funktion *"Exception occured: A negative Value was found in Templatecontainer"* zurückgibt. Diese Exception soll sich in einem eigens definiertem Namespace befinden (erste drei Buchstaben ihres Nachnamens). Werfen Sie diese Exception in **3b** wenn ein **negativer** Wert im Templatecontainer auftritt und ersetzen sie diesen negativen Wert mit der **positiven Repräsentation (*-1)** dieses Wertes.
- d. Erstellen Sie eine Template Funktion welche das **Quadrat** eines **beliebigen Typs** berechnet. Stellen Sie sicher das bei der Verwendung dieses Templates durch **Strings** eine Meldung auf **cerr** ausgegeben wird.
- e. Erstellen Sie eine eigene Template Klasse welche zwei Werte speichert und befüllen sie diese Werte über einen Konstruktor.
- f. Erstellen sie die Objekte **Rechteck** und **Quadrat** in der korrekten **is-a** Beziehung. Erstellen Sie nun einen **Copy** sowie einen **Move Konstruktor** welcher die Kopie bzw. das bewegen eines Objektes in ein anderes (in der is-a Beziehung abgebildet) ermöglicht.

Nachbearbeitung

- 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 Compile und temporären Dateien gelöscht sein.
- Löschen Sie das .vs Verzeichnis (Achtung: hidden folder!) im Verzeichnis der Solution, bevor Sie das zip-File der Solution erstellen.
- Für diese Aufgabe gibt es keine Abgabe! Sie ist freiwillig. Bei Feedbackwunsch bitte im Büro vorbeischauen oder Kontakt über Email.