BTH001: Assignment 2

This assignment focus on:

- Inheritance
  - abstract base class
  - polymorphism(多态)
  - dynamic binding（动态绑定）
- Composition using container class
- Deep copying and dynamic memory allocation
  - copy constructor
  - assignment operator
  - destructor

Runner handler

In a small competition where both professional and amateur runners will compete there is a need of a program to handle the registration of the runners. The program will later on be developed further by other system developers, so remember to be clear about the naming and comments in the source code.

All runners will be registered by name and start number. Professional runners will also be registered by club and sponsor. Amateurs will instead be registered by age.

For all runners it must be possible to get all the information contained in the object as a string (toString()-function). It must also be possible to get the name and to get the start number.

Concerning the professional runners it must also be possible to change the sponsor. Furthermore it must be possible to get the sponsor and to get the club.

When it comes to amateurs it must be possible to get the age.

All runners should be handled in a management system that have the functionality given below:

1) Add a professional runner using user input
2) Add an amateur runner using user input
3) Remove a runner given the start number where the user inputs the start number
4) Present all runners by name and start number (both professionals and amateurs)
5) Present all professional runners (name, start number, club and sponsor) and the total number of professional runners
6) Present all amateur runners (name, start number and age) and the total number of amateur runners.
7) Change the sponsor for professional runners. The user inputs start number and the new sponsor.
8) Present all amateurs in a given interval(间距) of ages. The user inputs the lower limit of the age and the upper limit of the age.

You shall create an **inheritance hierarchy(继承结构)** for runners. Abstract base class, polymorphism and dynamic binding are required (hint: virtual). You shall also create a **class containing the runners** in which one (1) dynamically allocated array is used (declared as a "double-pointer") to keep / store all the runners. This array shall expand if necessary. In the runner container class you shall also implement copy constructor, assignment operator and a virtual destructor – deep copying is required!

Keyboard input and screen output in the functions within the class containing the runners are not allowed. All runners must always be placed consecutive in the array.

Finally you shall implement a file containing the main function which corresponds(对应) to a menu-based system for handling furniture with the functionality described above in 1) – 8).

Other requirements:

- No memory leaks are allowed!
  Use:
  _CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF | _CRTDBG_LEAK_CHECK_DF);
  at the top of main function and run in debug-mode to detect memory leaks.
- Only private member variables are allowed
- All classes shall be divided into header- and cpp-files.
- The possibility to implement constant member functions shall be used
- Global variables are not allowed
- The menu based system shall handle separate part problems in separate functions