Tasks

Example 1. Inheritance

You are given an example of a class `Student` derived from a class `Person`.

Task 1.1

Define a new class `Employee` derived from the class `Person`. Instead of the field `unsigned int _groupId` for `Student`, define a field `std::string _position` for `Employee`. Otherwise, copy all the functionality of `Student`.

Task 1.2

Modify the class `Person` so that it can be constructed implicitly from `std::string` but only explicitly from string literals (`const char*`).

Example 2. Virtual functions

Task 2.1

Add class `Employee` to the second example as well.

Task 2.2

Modify the function `createSample` so it adds `Employee` objects to the resulting vector.

Task 2.3

Modify the test functions to use smart pointers (`std::unique_ptr`) instead of deleting data manually. Check that the destructors get called correctly.

Task 2.4 (complex)

Make `Professor` a class derived from `Employee` instead of `Person`. Check if it changes how containers of pointers behave.
Check that the constructors and destructors are called correctly. In this case there should be three constructor calls and three destructor calls for each object.

Example 3. Abstract classes

Task 3.1

Add a test function that uses a container of smart pointers to the abstract class to store objects of derived classes.