Chapter I. Boost.SmartPointers

Table of Contents

The library Boost.SmartPointers provides various smart pointers. They help you manage dynamically allocated objects, which are anchored in smart pointers that release the dynamically allocated objects in the destructor. Because destructors are executed when the scope of smart pointers ends, releasing dynamically allocated objects is guaranteed. There can't be a memory leak if, for example, you forget to call `delete`.

The standard library has included the smart pointer `std::auto_ptr` since C++98, but since C++11, `std::auto_ptr` has been deprecated. With C++11, new and better smart pointers were introduced in the standard library. `std::shared_ptr` and `std::weak_ptr` originate from Boost.SmartPointers and are called `boost::shared_ptr` and `boost::weak_ptr` in this library. There is no counterpart to `std::unique_ptr`. However, Boost.SmartPointers provides four additional smart pointers – `boost::scoped_ptr`, `boost::scoped_array`, `boost::shared_array`, and `boost::intrusive_ptr` – which are not in the standard library.