Weak Pointers

std::weak_ptr is the last component of the smart pointers family. Its purpose is limited compared to the other smart pointers, and we will examine why in this lesson.

WE'LL COVER THE FOLLOWING

Introduction

To be honest, std::weak_ptr is not a classic smart pointer, since it supports no transparent access to the resource; it only borrows the resource from an `std::shared_ptr`.

Methods

The table provides an overview of the methods of `std::weak_ptr`.

Name	Description
`expired`	Checks if the resource was deleted.
`lock`	Creates a `std::shared_ptr` on the resource.
`reset`	Resets the resource.
`swap`	Swaps the resources.
	Returns the value of the reference

`use_count`	Returns the value of the reference counter.

Methods of std::weak_ptr

> 💡 There is one main reason for the existence and use of `std::weak_ptr`. It breaks the cycle of `std::shared_ptr`. We will discuss these cyclic references in detail in the next lessons.

Further information

- std::weak_ptr
- cyclic references

Let's see an example of this topic in the next lesson.