|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **작성자** | **2014132002 구태균** | **팀명** | - |
| **주차** | **18 주차** | **기간** | **2020.05.11 ~ 2020.05.17** | **지도교수** | **정 내 훈** (서명) |
| **이번주 한일** |  | | | | |

shared\_ptr(const shared\_ptr& other)

{

if (!other.ctr) {

shared\_ptr();

return;

}

other.ctr->lock();

other.ctr->add\_ref\_copy();

ptr = other.ptr;

ctr = other.ctr;

other.ctr->unlock();

}

shared\_ptr(const weak\_ptr<Tp>& other)

{

if (!other.ctr) {

shared\_ptr();

return;

}

other.ctr->lock();

other.ctr->add\_ref\_copy();

ptr = other.ptr;

ctr = other.ctr;

other.ctr->unlock();

}

shared\_ptr& operator=(nullptr\_t)

{

if (ctr) {

ctr->lock();

if (!ctr->release())

ctr->unlock();

}

ctr = nullptr;

ptr = nullptr;

return \*this;

}

shared\_ptr& operator=(const shared\_ptr& other)

{

if (other.ptr == ptr)

return \*this;

ctr\_block<Tp>\* empty = ctr;

if (ctr)

ctr->lock();

if (other.ctr)

other.ctr->lock();

else {

if (ctr) {

ctr\_block<Tp>\* pred\_ctr = ctr;

ctr = nullptr;

ptr = nullptr;

if(!pred\_ctr->release())

pred\_ctr->unlock();

}

return \*this;

}

other.ctr->add\_ref\_copy();

ctr\_block<Tp>\* curr\_ctr = ctr;

ptr = other.ptr;

ctr = other.ctr;

other.ctr->unlock();

if (empty != curr\_ctr)

std::cout << "??" << std::endl;

if (curr\_ctr) {

if(!curr\_ctr->release())

curr\_ctr->unlock();

}

return \*this;

}

~shared\_ptr()

{

if (!ctr) return;

ctr->lock();

if (!ctr->release())

ctr->unlock();

}

weak\_ptr(const shared\_ptr<Tp>& other)

{

if (!other.ctr) {

weak\_ptr();

return;

}

other.ctr->lock();

other.ctr->weak\_add\_ref();

ptr = other.ptr;

ctr = other.ctr;

other.ctr->unlock();

}

weak\_ptr(const weak\_ptr<Tp>& other)

{

if (!other.ctr) {

weak\_ptr();

return;

}

other.ctr->lock();

other.ctr->weak\_add\_ref();

ptr = other.ptr;

ctr = other.ctr;

other.ctr->unlock();

}

~weak\_ptr()

{

ctr->lock();

if (!ctr->weak\_release())

ctr->unlock();

}

weak\_ptr& operator=(const shared\_ptr<Tp>& other)

{

if (other.ptr == ptr)

return \*this;

if (ctr)

ctr->lock();

if (other.ctr)

other.ctr->lock();

else {

if (ctr) {

ctr\_block<Tp>\* pred\_ctr = ctr;

ctr = nullptr;

ptr = nullptr;

if (!pred\_ctr->weak\_release())

pred\_ctr->unlock();

}

return \*this;

}

other.ctr->weak\_add\_ref();

ctr\_block<Tp>\* curr\_ctr = ctr;

ptr = other.ptr;

ctr = other.ctr;

other.ctr->unlock();

if (curr\_ctr) {

if (!curr\_ctr->weak\_release())

curr\_ctr->unlock();

}

return \*this;

}

weak\_ptr& operator=(const weak\_ptr& other)

{

if (other.ptr == ptr)

return \*this;

if (ctr)

ctr->lock();

if (other.ctr)

other.ctr->lock();

else {

if (ctr) {

ctr\_block<Tp>\* pred\_ctr = ctr;

ctr = nullptr;

ptr = nullptr;

if (!pred\_ctr->weak\_release())

pred\_ctr->unlock();

}

return \*this;

}

other.ctr->weak\_add\_ref();

ctr\_block<Tp>\* curr\_ctr = ctr;

ptr = other.ptr;

ctr = other.ctr;

other.ctr->unlock();

if (curr\_ctr) {

if (!curr\_ctr->weak\_release())

curr\_ctr->unlock();

}

return \*this;

}

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** |  | **해결 방안** |  |
| **다음 주차** | **19 주차** | **다음 기간** | **2020.05.18 ~ 2020.05.24** |
| **다음주 할 일** |  | | |
| **지도교수**  **Comment** |  | | |