بسمه تعالی

گزارشکار تکلیف چهارم برنامه نویسی
Shared Pointerو Unique Pointer

استاد مربوطه : دکتر جهانشاهی

تدریسیار مربوطه: مهندس کیان بهزاد

نویسنده : محمدمهدی شریفیان(9823053)

مقدمه

Unique pointer و Shared pointer هر دو از کلاس های C++ برای کار کردن با پوینتر ها هستند در این تمرین به پیاده سازی این کلاس ها پرداختیم. این کلاس ها به ما این امکان را میدهند تا بدون مشکلات مختلفی که ممکن است در حین کارکردن با پوینتر ها داشته باشیم با استفاده از این کلاس ها از قابلیت پوینتر ها استفاده کنیم.

Unique Pointer

Unique pointer یک pointer یکتا به یک شی خاص میباشد که قابلیت کپی شدن و یا تغییر دادن را ندارد و اکثر عملگر هایی که پوینتر ها دارند مانند عملگر های مختلف مانند -> و * را دارد.

پیاده سازی Unique pointer بسیار سر راست بود و چالش خاصی نداشت یک متغیر برای ذخیره کردن اشاره گر نیاز داشتیم که به نام p_ آن را تعریف کردیم. نکته جالب درمورد پیاده سازی copy constructor و =operator این بود که باید در هنگام استفاده از این اعمال خطای کامپایل داده میشد که این مشکل با delete کردن این دو عمل حل شد و زمانی که یک Unique pointer بخواهد کپی شود و یا تغییر داده شود خطای کامپایل رخ میدهد.

```
UniquePtr(const UniquePtr<T>& obj)=delete; // delete for compile error
UniquePtr& operator=(const UniquePtr& obj)=delete;
```

از نکات جدید این تمرین میتوان به این بحث اشاره کرد که به دلیل تمپلیتی بودن کلاس های تعریف شده مانند قبل فایل h. و cpp. نداشتیم بلکه همه تعاریف عملا در header آمده اند و از فایل های h. و hpp. استفاده کردیم.

پیاده سازی بقیه توابع بسیار سرراست بود و نکته خاصی در پیاده سازی آنها وجود نداشت. همچنان تعریف عملگر های مختلف مانند <- و * نیز بسیار سرراست بود و با برگرداندن p_ و یا قسمتی از حافظه که p_ به آن اشاره میکند به راحتی انجام شد.

در نهایت تعریف عملگر bool را برای این کلاس داشتیم تعریف آن بسیار مشابه با تعریف دیگر عملگر ها بود با این تفاوت که تایپ خروجی را برای این عملگر تعریف نکردیم و کامپایلر به صورت خودکار میداند که خروجب این عملگر حتما به صورت bool خواهد بود.

Shared Pointer

در این تمرین قسمت مشکل تر کار پیاده سازی این کلاس بود زیرا بخاطر ویژگی های خاصی که دارد و همچنین امکان تغییر دادن و کپی کردن آن پیاده سازی را قدری مشکل تر میکند.

این کلاس سه متغیر دارد p_ برای ذخیره کردن اشاره گر به کار میرود، p___ اشاره گر به اشاره گر میباشد تا در صورت لزوم برای عوض کردن مقدار p_ بتوانیم با استفاده از اشاره گر به آن از آن استفاده کنیم و در آخر متغیر reference میباشد این متغیر از نوع *int تعریف شده است و به جایی از حافظه اشاره میکند که تعداد شی هایی که دارای p_ یکسان هستند در آن ذخیره شده است. اینگونه میتوان از هر شی از کلاس به این مقدار دسترسی یافت و مقدار آن را تغییر داد. از این قابلیت در destructor, copy constructor, operator= و ... استفاده شده است.

مزیت Shared Pointer نسبت به اشاره گر معمولی این است که زمانی که جایی از حافظه باقی بماند که اشاره گری از این نوع به آن وجود نداشته باشد به صورت خودکار آن قسمت از حافظه پاک میشود و اینگونه هرگز memory leak نخواهیم داشت.

پیاده سازی constructor اول و default constructor سر راست است و متغیر های اولیه assign میشوند. در copy constructor نیز متغیر های شی جدید برابر شی کپی

شده قرار میدهیم و reference را به مقدار یک افزایش میدهیم.عملکرد =operator نیز بسیار مشابه به copy constructor میباشد.

بقیه توابع و عملگر ها کاملا مشابه با Unique pointer میباشد.تنها نکته قابل توجه این بود که در پیاده سازی destructor و reset هر زمان که متغیر reference به عدد صفر برسد و آخرین Shared Pointer به یک جای خاص از حافظه نیز از بین برود آن قسمت از حافظه نیز پاک میشود. و در غیر این صورت متغیر reference یک واحد کم میشود.

لینک گیت هاب