

تمرین سری 4 برنامه نویسی پیشرفته

علیرضا طباطبائی

تمامی دستورات دارای کامنت میباشند.

: Unique Pointer

ابتدا باید قبل تمامی توابع و در شروع تعریف کلاس ، `template <typename T>` را قرار دهیم که بداند `T` منظور یک تایپ خاص از متغیر است. همچنین باید فایل های `.hpp` را نیز در `.h` ، `include` کنیم.

Constructor : جهت تعریف کانستراکتور اصلی میبایست ورودی را بصورت پوینتر بگیریم زیرا دستور `new int{10}` دارای خروجی آدرس میباشد.

سپس `default constructor` را نیز تعریف میکنیم. (بکمک `nullptr`)

برای بستن دسترسی به `copy constructor` و اپراتور مساوی = میبایست آنها را برابر `delete` بگذاریم.

جهت تعریف `destructor` : میبایست متغیر هایی که در طول برنامه مقدار دهی شده اند مثل `p_` را ابتدا محتوای پشت آنها را `delete` کرده و سپس خود آنها را برابر `nullptr` کنیم.

تابع **get** : متغیر **p_** را به خروجی میفرستند.

اپراتور ***** : همانند **get** است که خروجی آن **dereference** شده است.

اپراتور میخ **<** : عینا همانند **get** است .

Reset : دارای 2 حالت با ورودی و بدون ورودی است که در حالت اول ، هم **delete** و هم **nullptr** داریم.

در حالت دوم بجز **delete** ، متغیر ورودی را نیز که بصورت رفرنس است به **p_** میدهیم و همان آبجکت را **return** میکنیم.

اپراتور **bool** : برای استفاده از **ptr** در ورودی **if** ، میبایست تعریف شود که در صورت **nullptr** بودن **ptr** میبایست **false** و در غیر اینصورت **true** برگردانده شود.

تابع **release** : باید از متغیر واسط استفاده کنیم که مقدار **p_** را در آن بریزیم و سپس **p_** را **nullptr** کرده و متغیر واسط را **return** نماییم .

Make unique : در فایل **.h** و بیرون از کلاس تعریف میشود و کاربرد آن ساخت آبجکت با استفاده از یک تابع است.

: Shared Pointer

توابع یکسان همانند بخش **unique pointer** میباشد بجز توابع زیر :

در **constructor** میبایست برای **counter** نیز مقدار تعیین کنیم که در حالت اصلی برابر رفرنسی به عدد 1 و در حالت **nullptr** برابر 0 و در سایر حالت ها برابر یکی بیشتر از حالت قبل است.

به عبارتی باید **counter** را بصورت پوینتر تعریف کنیم که بتوان از یک آجکت به آن دسترسی داشت و در آجکتی دیگر آن را تغییر داد.

در حالت **copy constructor** نیز میبایست آن را ابتدا **dereference** نماییم و یک عدد به آن اضافه کنیم و آدرس این متغیر را بازگردانیم. برای یک واحد اضافه کردن نیز میبایست طوری عمل کنیم که خروجی بصورت **L Value** شود وگرنه در **constructor** کار نمیکند.

در **destructor** نیز میبایست مقدار آن را یک واحد یک واحد کم کنیم تا وقتی که برابر 1 شد ، متغیر **p_** را برابر **nullptr** نماییم . اگر در هر مرحله برابر **nullptr** قرار دهیم ، ارور **double free** میگیریم که به دلیل آن است که یک متغیر را نمیتوان 2 مرتبه یا بیشتر **delete** کرد.

تابع `get` همانند قبل است.

اپراتور `=` مساوی ابتدا برابر نبودن ورودی و آبجکتی که در آن هستیم را بررسی میکند و سپس همانند `copy constructor` عمل میکند.

`Use_count` مقدار متغیر `counter` را برمیگرداند.

اپراتور ستاره و میخ و `bool` همانند قبل است.

در `reset` باید دقت شود که `counter` نیز بروز شود. اگر در `reset` ورودی نداشتیم، `counter` برابر پوینتری به عدد 0 و اگر ورودی داشتیم، پوینتری به عدد 1 شود.

آدرس گیت هاب
