

به نام خدا

آموزش برنامه نویسی (کتابخانه set)

نویسنده : محمدصادق دهقان نیری

تهیه شده توسط وبلاگ رقابت

...: Rivalry.blog.ir ...

Set چیست؟

ست (Set) در واقع مجموعه ظروفی است که چیز های تکراری را قبول نمی کند برای مثال اگر به ست اعداد ۱۰ و ۲۰ و ۳۰ و ۱۰ را بدهیم ، چیزی که در ست ذخیره می شود برابر خواهد بود با ۱۰ و ۲۰ و ۳۰ . پس در ست هیچ دو عضو دقیقا عین هم وجود ندارد.

خب اگه تا اینجا یه خورده گیج شدید و فکر می کنید ست چیز عجیب غریبه طرز فکرتون کاملا غلته D: . هنوز اول کاره بابا.

تعریف یک Set

فرم کلی ست در زیر آمده است. در برنامه زیر ما یک ست تعریف کرده ایم به اسم myset . همون طور که می بینید بین علامت های < > نوع خونه های ست نوشته می شه (همون متغیر های خودمون!!!)

```
1 #include <iostream>
2 #include <set>
3 Using namespace std;
4 int main ()
5 {
6     set<int> myset;
7
8     return 0;
9 }
```

اضافه کردن یک عضو به Set

برای اضافه کردن یک عضو به ست از insert استفاده میشه. فرم کلی آن به این شکله:

YourSetName . insert (Value or Variable);

برای مثال به کد زیر توجه کنید.

```
1 #include <iostream>
2 #include <set>
3 Using namespace std;
4 int main ()
5 {
6     std::set<int> myset;
7     int a = 10;
8     int b = 20;
9     int c = 30;
10    myset.insert(a);
11    myset.insert(b);
12    myset.insert(c);
13    myset.insert(20);
14    return 0;
15 }
```

اعضای ست برابر است با ۱۰ و ۲۰ و ۳۰

البته به صورت های دیگه هم می تونید عضو تعریف کنید. مثل این:

```
int myints[]= {5,10,15};  
myset.insert (myints,myints+3);
```

توجه : همیشه اعضای ست به صورت مرتب شده اند(sorted).

به دست آوردن آدرس خانه های اول و آخر Set

برای به دست آوردن اولین خانه و آخرین خانه ست از دو دستور begin و end استفاده می کنیم برای مثال:

```
YourSetName . begin();
```

```
YourSetName . end();
```

این دو تا دستور آدرس اولین و آخرین خونه ست رو به شما بر می گردونه.

توی چاپ اعضا خیلی به درد می خوره D :

چاپ اعضای Set

بر عکس آرایه ها ما نمی تونیم برای دسترسی به خونه های ست یه همچین چیزی رو بنویسیم:

```
myset[5];
```

برای همین به شکل زیر عمل می کنیم:

برای چاپ اعضای ست از iterator استفاده می شه . فرم کلی تعریف یک iterator :

```
Set <Variable type> :: iterator YourIteratorName ;
```

برای مثال به کد زیر توجه کنید:

```
1 #include <iostream>  
2 #include <set>  
3 Using namespace std;  
4 int main ()  
5 {  
6     set<int> myset;  
7     set <int> :: iterator it ;  
8     myset.insert(200);  
9     myset.insert(100);  
10    cout << "myset contains:";  
11    for (it=myset.begin(); it!=myset.end(); ++it)  
12        cout << *it<<endl;  
13  
14    return 0;  
15 }
```

خروجی کد زیر برابر است با ۱۰۰ و ۲۰۰

دقت کنید که iterator یک اشاره گره(pointer). درباره پوینتر ها قبلا سامان توضیح مختصری

داده.ایشالله اگه وقت کنیم مفصل درباره اشاره گر ها هم توضیح می دیم.

دستور size

دستور سایز تعداد عضو های ست شما رو بر می گردونه و به صورت زیر تعریف میشه:

```
YourSetName.size ();
```

برای این دستور فکرنمی کنم مثال نیاز باشه. بدیهه!!!

دستور max_size

دستور مکس سایز تعداد حد اکثر تعداد عضوی که می تونه ست شما داشته باشه رو برمی گردونه. به صورت زیر تعریف میشه:

```
YourSetName.max_size ();
```

به مثال زیر دقت کنید:

```
1 #include <iostream>
2 #include <set>
3 Using namespace std;
4 int main ()
5 {
6     int i;
7     set<int> myset;
8
9     if (myset.max_size()>1000)
10    {
11        for (i=0; i<1000; i++) myset.insert(i);
12        cout << "The set contains 1000 elements.\n";
13    }
14    else cout << "The set could not hold 1000 elements.\n";
15
16    return 0;
17 }
18
```

دستور erase

این دستور برای پاک کردن اعضای ست به کار میره و فرم کلیش اینجوریه:

```
YourSetName.erase(iterator);
```

```
YourSetName.erase(value);
```

```
YourSetName.erase(start,end);
```

همون طوری که می بینید می تونید به این دستور یک آدرس یا ایتريتور بدید تا اونو پاک کنه همچنین می تونید به جای دادن آدرس خود اونو بدید مثلا اگه می خواد عدد ۰ رو پاک کنید

بجای عدد ۴۰ رو بزارید. علاوه بر این ها می تونید آدرس اولین عضو و آخرین عضوی رو که می خواهید پاک کنید رو بدید تا این دستور اون بازه رو به طور کامل پاک کنه.
برای درک کردن بهتر به مثال زیر توجه کنید:

```
1 #include <iostream>
2 #include <set>
3 Using namespace std;
4 int main ()
5 {
6     set<int> myset;
7     set<int>::iterator it;
8
9     // insert some values:
10    for (int i=1; i<10; i++) myset.insert(i*10); // 10 20 30 40 50 60 70 80 90
11
12    it = myset.begin();
13    ++it; // "it" points now to 20
14    myset.erase (it);
15    myset.erase (40);
16    ++it;
17    ++it;
18    ++it;
19    ++it; // just for fun :D
20    myset.erase (it, myset.end());
21
22    std::cout << "myset contains:";
23    for (it=myset.begin(); it!=myset.end(); ++it)
24        std::cout << ' ' << *it;
25    std::cout << '\n';
26
27    return 0;
28 }
```

خروجی برنامه بالا برابر با: ۱۰ و ۳۰ و ۵۰

همون طور که می بینید ما اول ، اولین عضو رو پاک می کنیم بعد عدد ۴۰ رو و بعد ایتريتور رو به عدد ۶۰ می رسونیم و بعد از عدد ۶۰ تا ۹۰ رو پاک می کنیم.

دستور swap

این دستور جای دو تا ست رو با هم عوض می کنه

YourSetName.swap(a set);

برای مثال:

```
1 // swap sets
2 #include <iostream>
3 #include <set>
4 Using namespace std;
5 main ()
6 {
7     int myints[]={12,75,10,32,20,25};
8     set<int> first (myints,myints+3); // 10,12,75
9     set<int> second (myints+3,myints+6); // 20,25,32
10
11    first.swap(second);
12
13    return 0;
14 }
```

همون طور که می بینید جای دوتا ست دقیقا با هم عوض میشه یعنی اعضای ست اولی برابره با: ۲۰، ۲۵، ۳۲ و اعضای ست دومی برابره با: ۱۰، ۱۲، ۷۵

دستور empty

این دستور چک می کنه که ست شما خالیه یا نه به عبارتی دیگه ست شما دارای عضوی هست یا نه. دقت کنید که این دستور ست شما رو پاک نمی کنه و کارش چک کردنه.

YourSetName.empty();

مثال:

```
1 #include <iostream>
2 #include <set>
3 Using namespace std;
4 int main ()
5 {
6     set<int> myset;
7
8     myset.insert(20);
9     myset.insert(30);
10    myset.insert(10);
11
12    cout << "myset contains:";
13    while (!myset.empty())
14    {
15        cout << ' ' << *myset.begin();
16        myset.erase(myset.begin());
17    }
18    std::cout << '\n';
19
20    return 0;
21 }
22
```

تو این دستور به یه حلقه تکرار میرسه که تا زمانی که ست دارای عضو هست کار می کنه. بعد توی حلقه اولین عضو رو چاپ می کنه بعد اولین عضو رو پاک می کنه و در واقع عضو دوم برابر میشه با اولین عضو و همین جور ادامه پیدا می کنه.

دستور clear

این دستور کل اعضای ست رو پاک می کنه. خیلی ساده.

YourSetName.clear();

فکر نکنم نیاز به مثال داشته باشه.

دستور find

این دستور یک ورودی می گیرد و اونو تو ست سرچ می کنه اگه وجود داشت آدرس اون ورودی رو بر می گردونه وگرنه مقدار غلط (false) رو بر می گردونه.

YourSetName.find(value);

مثال:

```
1 #include <iostream>
2 #include <set>
3 Using namespace std;
4 int main ()
5 {
6     set<int> myset;
7     set<int>::iterator it;
8
9     // set some initial values:
10    for (int i=1; i<=5; i++) myset.insert(i*10);    // set: 10 20 30 40 50
11
12    it=myset.find(20);
13    myset.erase (it);
14    myset.erase (myset.find(40));
15
16    return 0;
17 }
```

پایان

نویسنده : محمد صادق دهقان نیری

تهیه شده در وبلاگ رقابت

... Rivalry.blog.ir ...