

UOT 372.0:002

Zəhra Rövşən qızı Həsənova
Lənkəran Dövlət Universiteti

C++ PROQRAMLAŞDIRMA DİLİNDƏ SİYAHILARIN YARADILMASI VƏ SİYAHILARLA MÜXTƏLİF ƏMƏLİYYATLARIN APARILMASI

Захра Ровшан гызы Гасанова
Ленкоранский Государственный Университет

СОЗДАНИЕ СПИСКОВ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ C++ И СПИСКАМИ ПРОВЕДЕНИЯ РАЗЛИЧНЫХ ОПЕРАЦИЙ

Zahra Rovshan Hasanova
Lenkoran State University

C++ PROGRAMMING LANGUAGE AND CREATE A LIST WITH THE LIST OF THE VARIOUS OPERATIONS

Xülasə: Məqalədə C++ dilində yazılmış proqramlarda çox geniş istifadə olunan yeni tiplər – siyahılar haqqında məlumat verilmişdir. Bundan əlavə, obyektlərin, siyahıların yaradılma qaydası da öz əksini tapmışdır. Siyahılarla işləmə bacarığını artırmaq məqsədilə nümunə proqramlar verilmiş və yerinə yetirilmişdir.

Açar sözlər: C++ proqramlaşdırma dili, siyahılar, siyahılarla əməliyyat, cərgələr, obyektlərin yaradılması

Резюме: В статье представлена информация о новых типах программ, которые широко используются в C++. Кроме того, также включены правила создания объектов и списков. Были выпущены и реализованы типовые программы для повышения их способности работать со списками.

Ключевые слова: язык программирования C++, списки, операции списка, строки, создание объектов.

Summary: The article provides information about new types of programs that are widely used in C++. In addition, the rules for creating objects and lists are also included. Exemplary programs have been issued and implemented to increase their ability to work with lists.

Key words: C++ programming language, lists, list operations, rows, objects creation

Siyahıların C++ dilində tətbiqi olduqca genişdir və siyahısız C++ dilində yazılmış proqramları təsəvvür etmək mümkün deyil. Praktiki cəhətdən siyahılar cərgələrə oxşardır, onlar da cərgələr kimi özündə verilmiş tiptən olan elementlər ardıcılığını saxlayır. Lakin siyahıların cərgələrdən fundamental üstün cəhətləri odur ki, cərgə elan edən zaman biz onun elementlərinin sayını əvvəlcədən elan etməliyiksə və sonra biz cərgəyə əlavə element yerləşdirə və ya onun elementlərinin sayını dəyişdirə bilməyiksə, siyahının elementləri ilə istədiyimiz kimi manipulyasiya edə bilərik. Yəni proqramın icrası boyu biz

siyahıya istənilən sayda yeni element əlavə edə və ya mövcud elementləri siyahıdan silə bilərik. Misal üçün, kompüterdə hal-hazırda icra olunan proqramlara nəzarət etmək üçün nüvə *task_struct* adlı siyahıdan istifadə edir. İstifadəçi yeni proqramlar yüklədikcə nüvə *task* -lar siyahısına yeni element əlavə edir və bu elementdə yeni proqram barədə müvafiq məlumatları (yükləndiyi yerin ünvanı, adı, təşkil olduğu hissələr, açdığı fayllar, v.s.) yerləşdirir. Daha sonra hər hansı proqram sona çatdıqda nüvə müvafiq elementi *task* -lar siyahısından silir. Əgər bu zaman siyahı əvəzinə cərgədən istifadə olunsaydı,

onda biz kompüterdə müəyyən saydan artıq proqram yükləyə bilməzdik. Cərgənin elementlərinin sayın böyük götürükdə isə lazım olmayan elementlər boş-boşuna yaddaşda yer tutar. Siyahının elan edilməsi int tipindən biz adi dəyişən, ünvan dəyişəni və cərgə elan etmək qaydalarını vermişik.

```
int x, *y, z[10];
```

Misalda biz *int* tipindən *x* dəyişəni, *y* ünvan dəyişəni və 10 elementdən ibarət *z* cərgəsi elan etdik. İndi isə hər bir elementində *int* tipli dəyişən olan siyahı elan edək. Bunun üçün əvvəlcə siyahını təşkil edən obyektlərin tipini yaratmalıyıq. Yəni, qeyd etdik ki, siyahı da cərgə kimi elementlər (obyektlər) ardıcılığıdır və bu elementlərin (obyektlərin) hər birində müxtəlif məlumatlar yerləşdirmək olar. İndi biz həmin 1 obyektı yaratmağa çalışırıq, daha sonra bir neçə bu cür obyektı bir-biri ilə əlaqələndirib siyahı yaratma qaydasını öyrənəcəyik. Baxdığımız sadə halda yaratmaq istədiyimiz obyekt özündə bir məlumat - *int* tipindən olan dəyişən saxlayır. Bu obyektı yaratmaq üçün biz *struct* tiptən istifadə edəcəyik.

Aşağıdakı kimi:

```
struct syh_el {  
    int x;  
}
```

Yuxarıda biz özündə *int* tipindən olan dəyişən, *x* -i saxlayan yeni strukt tipi aratdıq. İndi bu tiptən olan bir neçə dəyişən elan edək.

```
syh_el dey1, dey2, dey3, dey4;
```

Hal-hazırda yaddaşda vəziyyət aşağıdakı kimidir:

Biz isə siyahı almaq üçün bu obyektləri bir-biri ilə əlaqələndirməliyik. Şəkildə çəkmək asandır, amma praktikada obyektlər arasında bu əlaqəni yaratmaq elə də ilk cəhddə asan deyil. Siyahının yaradılması üçün aşağıda daxil ediləcək qayda bir az çətin ola bilər. Siyahılarla işləmə vərdişi yarandıqdan sonra mahiyyəti qavramaq da asanlaşır.

Siyahı yaratma qaydası:

Əsas məsələ bir obyektı başqa obyekt ilə əlaqələndirməkdir. Obyektlərin bir-biri ilə əlaqələndirilməsi ardıcıl aparılır. Yəni dey 1, dey 2 ilə, dey 2, dey 3 ilə və s ardıcılıqla əlaqələndirilir. Beləliklə, siyahıya istənilən sayda yeni obyekt əlavə edib siyahımızı istədiyimiz qədər

uzada bilərik. Bəs bir obyektı digəri ilə necə əlaqələndirək? Bunun üçün proqramlaşdırmada ilk baxışda elə də anlaşılmayan aşağıdakı qaydadan istifadə edirlər:

Siyahıda hər bir obyekt özündən sonra gələn obyektin yaddaşdakı ünvanın bilməlidir. Bunun üçün obyektin daxilində onun öz tipindən olan ünvan dəyişəni yerləşdirirlər və bu ünvan dəyişəninə siyahının növbəti obyektinin ünvanını mənimsədirlər.

Aşağıdakı elana nəzər salaq:

```
struct syh_el {  
    int x;  
    syh_el *novb_el;  
}
```

Bu elanın yuxarıdakı elandan yeganə fərqi o oldu ki, biz burada *struct syh_el* tipinin daxilində bu tiptən olan **novb_el* ünvan dəyişəni yerləşdirdik. Artıq bu bizə siyahı yaratmağa imkan verir. Əvvəlcə biz siyahımızı elan etməliyik. Bu zaman biz prinsip etibarı ilə *struct syh_el* tiptən olan ünvan dəyişəninə aşağıdakı kimi elan etmiş oluruq:

```
syh_el *menim_syh;
```

Siyahı yaratmaq üçün *syh_el* tiptən olan daha iki ünvan dəyişəninə ehtiyacımız olacaq. Bunları *p* və *q* ilə işarə edək. Bunlardan biri - *p*, yeni obyektlərin yaradılması və inisializasiyası (ilk qiyətin mənimsədilməsi), digəri - *q* isə iterasiya üçündür (siyahı boyu hərəkət etmək). Gəlin bu dəyişənləri də elan edək:

```
syh_el *p, *q;
```

İndi isə siyahının ilk elementini yarada bilirik.

Bunun üçün yazırıq:

```
p = new syh_el;
```

Bu zaman yaddaşın vəziyyəti dəyişir. Növbəti addım *menim_syh* dəyişəninə bu yeni yaratdığımız obyektə mənimsətməkdir. Bunun üçün sadəcə yazırıq:

```
menim_syh = p;
```

Yaddaşın vəziyyəti yenə dəyişir.

Artıq *menim_syh* dəyişənindən istifadə etməyəcəyik. O, siyahının ilk elementinə istinad edir və siyahıya müraciət etmək üçün bu dəyişəndən istifadə edəcəyik.

Əlbəttə, gələcəkdə siyahı üzərində əməliyyat aparsaq, əvvəlki obyektlərdən bəzilərinə sil-

sək, *onda menim_syh* parametri üzərində də çevrilmələr aparmalıyıq.

Hələlik isə siyahının yaradılmasını davam etdirək. İkinci elementi yaratmaq üçün hazırlıq işləri görək. Bunun üçün iterasiya dəyişənini siyahının ilk elementinə (*p* -yə) , ilk elementin ikinci elementlə əlaqələndirmə həddini (*novb_el*) isə NULL qiymətinə mənimsətməliyəm. NULL adətən ünvan dəyişənlərinin yaddaşda heç bir yerə ünvanlanmadığını bildirmək üçün istifadə olunur. Bu bizə siyahının sonun müəyyənləşdirmədə lazım olacaq. Eyni ilə birinci elementi yaratdığımız kimi siyahının ikinci elementini yaradaq:

```
p = new syh_el;
```

İlk element ilə yeni yaratdığımız elementi birləşdiririk. Bu əsas məqamlardan biridir. Bunun üçün sadəcə yazırıq:

```
q->novb_el = p;
```

Bununla da əlaqə yarandı.

Növbəti 3-cü elementi yaratmaq üçün hazırlıq işləri görək:

```
q = p;
```

```
p->novb_el = NULL;
```

Yaddaş yenə dəyişir, artıq bu şəkildə davam edərək siyahıya istədiyimiz qədər yeni element əlavə edə bilərik. Tutaq ki, siyahıya müəyyən qədər element əlavə etmişik və siyahımız var, sonda *p* və *q* dəyişənlərinin siyahıya istinadlarını ləğv etməliyik.

```
p=NULL;
```

```
q=NULL;
```

Bununla da siyahı yaratma prosesini tamamlamış olduq. Çətin görünə bilər, amma siyahılar proqramlaşdırmada əvəz edilməzdir və çox praktik məsələ olmaqla həyatda tez-tez rast gəlinir. Aşağıdakı nümunə proqramları yerinə yetirdikdən sonra siyahılarla işləmə bacarığımız bir qədər də artmış olar.

Proqram nümunəsi ilə işimizi tamamlayaq:

Gəlin yuxarıda daxil etdiyimiz nümunəyə uyğun proqram hazırlayaq. Yəni, sadə bir proqram tərtib edək, bu proqramda özündə ancaq bir hədd, *int x*; saxlayan obyektlərdən ibarət siyahı tərtib edək, bu siyahının obyektlərinin hədlərinə (*x*) qiymətlər mənimsədək, daha sonra siyahının elementlərini çap edək. Hələlik biz hər şeyi əl qaydası ilə aparacağıq, bir az sonra isə prosesi

avtomatlaşdırmaq məqsədilə funksiyalardan istifadə edəcəyik.

Proqram 1.

```
#include <iostream>
```

```
#include <string.h>
```

```
struct syh_el{
```

```
int x;
```

```
syh_el *novb_el};
```

```
int main (int argc, char *argv[]){
```

// istifadə edəcəyimiz dəyişənləri elan edirik

```
syh_el *menim_syh, *p, *q;
```

```
int dey;
```

// ilk elementi yaradaq

```
p = new syh_el;
```

// lazımı mənimsətmələri aparaq

```
menim_syh = p;
```

```
q = p;
```

```
p->novb_el = NULL;
```

// her şey hazırdır ikinci elementi yaradıırıq

```
p = new syh_el;
```

// siyahı ilə ikinci elementin əlaqəsini

qururuq

```
q->novb_el = p;
```

// lazım mənimsətmələri edirik

```
q = p;
```

```
p->novb_el = NULL;
```

// artıq siyahıda iki obyekt var, 3-nu yaradaq

// 3-cu elementi yaradıırıq

```
p = new syh_el;
```

// siyahının sonu ilə üçüncü elementin əlaqəsini qururuq

```
q->novb_el = p;
```

// lazımı mənimsətmələri edirik

```
q = p;
```

```
p->novb_el = NULL;
```

// siyahıda hazırda 3 element var, hələlik bəsdir.

// p ve q -nu siyahıdan ayıraq

```
p=NULL;
```

```
q=NULL;
```

/ vəssalam indi mənim_syh dəyishəni yeni yaratdığımız siyahının ilk elementinə istinad edir və onun vastəsilə siyahını bütün obyektlərinə müraciət edə bilərəm*/*

C++ proqramlaşdırma dilində siyahıların yaradılması və siyahılarla müxtəlif əməliyyatların aparılması

```
/* Siyahının elementlərinə qiymətlər
mənimsədik, daha sonra bu
Qiymətləri çap edəcəyik*/
// Yenə p yə ehtiyacımız olacaq
p = mənim_syh;
// indi p dayanıb siyahının əvvəlində ,
aşağıdakı koda diqqət eləyin
std::cout<<"Siyahının həddlərinin
qiymətlərini daxil edin.\n";
std::cout<<"Siyahının birinci həddinin
qiymətini daxil edin. \n";
std::cin>>dey;
/*Siyahının ilk obyektinin x həddinə
istifadəsinin daxil etdiyi
Qiyməti mənimsədirəm */
p->x = dey;
/* Siyahının ikinci obyektini
sürüşmək üçün aşağıdakı
qaydadan istifadə olunur */
p = p->novb_el;
/* Artıq p siyahının ikinci obyektinə
istinad edir , onun x həddinə
qiymət mənimsədik*/
std::cout<<"Siyahının ikinci həddinin
qiymətini daxil edin. \n";
std::cin>>dey;
p->x = dey;
p = p->novb_el və s.
Proqramı yerinə yetirək:
C:\cpp\prog2\Debug>
C:\cpp\prog2\Debug>prog2.exe
Siyahının həddlərinin qiymətlərini daxil
edin.
Siyahını birinci həddinin qiymətini daxil
edin.
65
Siyahını ikinci həddinin qiymətini daxil
edin.
```

345
Siyahını üçüncü həddinin qiymətini daxil
edin.
78
Siyahının elementləri aşağıdakılardır:
Birinci element 65
İkinci element 345
Üçüncü element 78
Siyahının elementləri çap olundu
C:\cpp\prog2\Debug>
C:\cpp\prog2\Debug>

Bu nümunədə biz sadə siyahı yaratdıq, 3 elementdən ibarət, siyahının həddlərinə qiymətlər mənimsətdik və bu qiymətləri çap etdik. Praktikiada bu cür məsələlərin həllində funksiyadan istifadə etmək zəruridir. Yuxarıdakı proqramın funksiyaların tətbiqi ilə olan variantını nəzərdən keçirək. Funksiyalardan istifadə etməklə yuxarıda baxdığımız proqram aşağıdakı kimi olar:

Bizə mahiyyət etibarilə 2 funksiya lazımdır: biri siyahını yaratmaq və ona elementləri yerləşdirmək, digəri isə siyahının elementlərini çap etmək.

Bu əməliyyatları davam etdirməklə istənilən bazanı yaratmaq, siyahıları tərtib etmək və onun üzərində müxtəlif əməliyyatlar aparmaqla çap etmək olar.

Problemin aktuallığı. İstənilən bazanı yaratmaq, siyahıları tərtib etmək və onun üzərində istənilən əməliyyatları aparmaq olar.

Problemin yeniliyi. Problemin yeniliyi ondan ibarətdir ki, hər bir bazanı məhz müasir proqramlaşdırma dili üzərində yaratmaq olur.

Problemin praktiki əhəmiyyəti. Problemin praktiki əhəmiyyəti ondan ibarətdir ki, digər dillərdə yaradılması çətinlik təşkil edilən baza yaratmaq və onun üzərində müxtəlif əməliyyatlar mümkün olur.

Ədəbiyyat

1. Əliyev A.Y. İnformatika və proqramlaşdırma. Bakı, 2008.
2. Окулов С.М. Задачи по программированию. М., 2006.
3. Кərimov S.Q., Həbibullayev S.B., İbrahimzadə T.İ. İnformatika. B., 2003.
4. C++ proqramlaşdırma dili: elektron dərslik. Bakı, 2011.
5. Князева М.Д. Алгоритмика от алгоритма к программе. М., 2006.
6. Hübətəliyev R.Z., Tağıyev H.N., Bayramova E.N. Proqramlaşdırma dilləri. Bakı, 2017.

E-mail: hasanovazahra@gmail.com

Rəyçilər: ped.ü.elm.dok., prof. Hümeyir Hüseyn oğlu Əhmədov

Redaksiyaya daxil olub: 18.04.2018