

Указатели и псевдоними

УП Практикум, 2ра група
Богомил Стоянов
Виолета Кастрева

Въведение

Указателите (пойнтъри) и псевдонимите (референции) са основни части от C++.

Те осигуряват ниво на контрол и ефективност при управление на паметта и данните.

Разбирането на тези концепции е от решаващо значение за писането на ефективен и стабилен C++ код.

I heard he knows
how to use pointers



Пойнтъри

Променлива, която
съхранява адреса на
паметта на друга
променлива.



```
1 // декларация на пойнтьър
2 int* ptr;
3
4 =====
5
6 int num = 42;
7 // инициализация на пойнтьър
8 int* ptr = &num;
```




```
1 // взимане на стойността в паметта
2 int value = *ptr;
3
4 =====
5
6 // поинтърна аритметика
7 ptr++; ptr += 2;
8
9 =====
10
11 // обхождане на масив
12 int thirdElement = *(ptr + 2);
13
14 =====
15
16 int* ptr = nullptr;
```



Динамична памет

new, delete

∀ new ∃ delete



```
1 int* dynamicPtr = new int;  
2 *dynamicPtr = 42;  
3 delete dynamicPtr;
```

Пойнтьър към функции



```
1 #include <iostream>
2
3 int multiply(int a, int b) { return a * b; }
4
5 int main()
6 {
7     int (*func)(int, int);
8     func = multiply;
9
10    int prod = func(15, 2);
11    std::cout << "The value of the product is: " << prod << std::endl;
12
13    return 0;
14 }
```




```
1 #include <iostream>
2
3 int apply(int a, int b, int (*operation)(int, int)) {
4     return operation(a, b);
5 }
6
7 int add(int x, int y) {
8     return x + y;
9 }
10
11 int main() {
12     int num1 = 10;
13     int num2 = 5;
14
15     std::cout << "Addition: " << apply(num1, num2, add) << std::endl;
16
17     return 0;
18 }
```

C isn't that hard:

`void (*(*f[])())()` defines `f` as an array of unspecified size, of pointers to functions that return pointers to functions that return `void` .

Shuntz

Референции

Псевдоним за
съществуваща променлива.



```
1 int num = 10;
2 // декларация на референция
3 int& ref = num;
4
5 // променя num=30
6 ref = 30;
7
8 =====
9
10 void modifyValue(int& x)
11 {
12     x = x * 2;
13 }
```



```
1 int& getReference() {  
2     int num = 10;  
3     return num;  
4 }  
5  
6 int main() {  
7     int& ref = getReference();  
8     // 'ref' е референция към променлива,  
9     // която вече не съществува => ОПАСНО!  
10    return 0;  
11 }
```

Пойнтър vs. референция

Използвайте указатели, когато е необходимо директно манипулиране на паметта или незадължителни параметри.

Използвайте псевдоними за по-четлив и по-безопасен код, особено във параметрите на функции и когато не е необходимо повторно присвояване.

В много случаи псевдонимите се предпочитат заради тяхната безопасност и синтактично удобство, но указателите осигуряват повече гъвкавост в определени сценарии. Изборът зависи от конкретните изисквания на поставената задача.

Край