# Јесен у мом крају



Миша (Miodrag.Djukic@rt-rk.uns.ac.rs) ФТН, Нови Сад (ftn.uns.ac.rs)

NIT-Institute, Нови Сад (sr.nit-institute.com)

Када С++ стандард дође у мој крај, сви се радују опадању старих, пожутелих фичера...

arr[i, j]

Када С++ стандард дође у мој крај, сви се радују опадању старих, пожутелих фичера...

```
NT operator,(MT x, MT y);
OT operator[](NT x);
arr[i, j]
```

Када С++ стандард дође у мој крај, сви се радују опадању старих, пожутелих фичера... и увођењу нових...

```
NT operator,(MT x, MT y);
OT operator[](NT x);

arr[ivi]
arr[ivi]
arr[i, j]
```

arr[(i, j)]

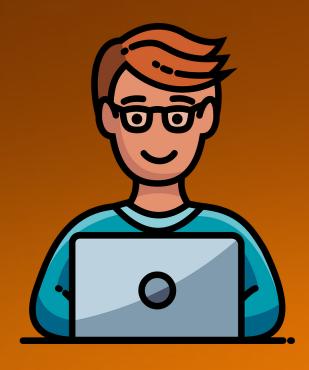
# …нови елементи библиотеке шуштећи прекривају употребу макроа…

```
___FILE______func____
__LINE____
__PRETTY_FUNCTION___
```

## ...нови елементи библиотеке шуштећи прекривају употребу макроа...

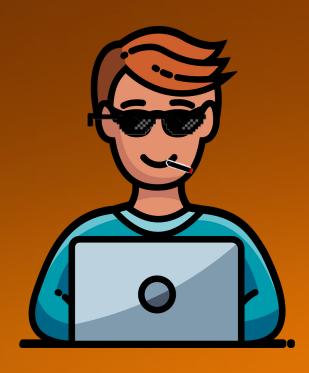
...док уведене новине у својим крошњама крију корисне детаље...

```
#define TWO(...) \
    foo(__VA_ARGS___); \
    bar(x_g, __VA_ARGS__)
TWO(a, b);
  foo(a, b);
  bar(x g, a, b);
TWO();
  foo();
  bar(x_g,);
```



...док уведене новине у својим крошњама крију корисне детаље...

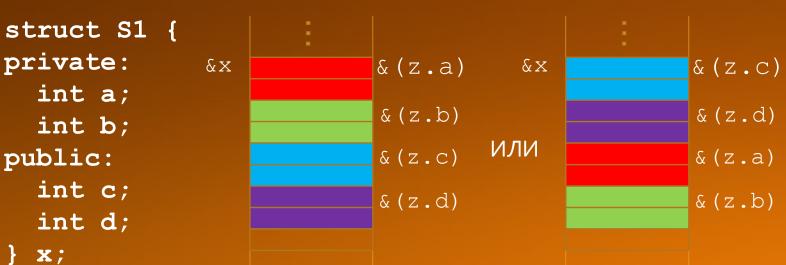
```
#define TWO(...) \
    foo(__VA_ARGS___); \
   bar(x g VA OPT (,) VA ARGS )
TWO(a, b);
  foo(a, b);
 bar(x g, a, b);
TWO();
  foo();
 bar(x_g);
```



...оголели проблеми које нисмо ни знали да имамо...

```
struct S1 { &x
    int a;
    int b;
    int c;
    int d;
} x;
& (z.a)
& (z.b)
& (z.c)
& (z.c)
& (z.d)
```





### ...оголели проблеми које нисмо ни знали да имамо...

```
struct S1 {
                            & (z.a)
  int a;
                            &(z.b)
  int b;
  int c;
                            &(z.c)
  int d;
                            & (z.d)
} x;
                                              C++23
struct S1 {
private:
                                                     (Z.C)
                            & (z.a)
                                       &X
               &Χ
  int a;
                                                     & (z.d)
                            &(z.b)
  int b;
                                     ИЛИ
                            &(z.c)
                                                     &(z.a)
public:
  int c;
                                                      (z.b)
                            & (z.d)
  int d;
  X;
```

...сваким следећим стандардом наш језик постаје старији и зрелији...

```
struct S {
  int a;
  int b;
  int c;
};
S \times = \{.a = 5, .b = 6, .c = 7\};
S x = \{.b = 8, .a = 7\};
S x = \{5, c = 7\};
int arr[1000] = {[567] = 1};
```

...сваким следећим стандардом наш језик постаје старији и зрелији...

```
struct S {
  int a;
  int b;
  int c;
S x = \{.a = 5, .b = 6, .c = 7\};
    <del>= {.b = 0, .a = 7};</del>
```



```
struct MyClass {
  void foo(int x);
};
MyClass a;
a.foo(i);
                Колико параметара прима функција foo?
```

```
struct MyClass {
  void foo(int x);
};
MyClass a;
a.foo(i);
                 Колико параметара прима функција foo?
                 Два!
```



```
struct MyClass {
                         struct MyClass {
  void foo(int x);
                           friend foo(MyClass&, int);
                         };
                         void foo(MyClass& t, int x)
MyClass a;
                         MyClass a;
a.foo(i);
                         foo(a, i);
movl $i, %esi
                         movl $i, %esi
movl $a, %edi
                       movl $a, %edi
call MyClass::foo(int) call func(MyClass&, int)
```

```
struct MyClass {
                              void foo(int x);
  void operator()(int x);
};
MyClass foo;
foo(i);
                              foo(i);
// foo.operator()(i);
// operator()(foo, i);
```



```
struct MyClass {
    ...
    //void foo(int x);
    void foo(this MyClass& self, int x);
    ...
};

MyClass a;
a.foo(i);
```



```
struct MyClass {
    ...
    //void operator()(int x);
    void operator()(this MyClass& self, int x);
    ...
};

MyClass foo;

foo(i);
```



```
struct MyClass {
  template<typename T>
  auto&& foo(this T&& self, int x);
};
const MyClass a;
MyClass b;
a.foo(i);
b.foo(j);
```

Поред методичког доприноса <sup>©</sup> ово је важно јер омогућава закључивање типа тог првог аргумента код шаблона функција чланица.

```
struct MyClass {
   void operator()(this MyClass& self, int x);
};

MyClass foo;

foo(i);
```

```
struct MyClass {
   static void operator()(this MyClass& self, int x);
};

MyClass foo;

foo(i);

// MyClass::operator()(i);
```

...неке могућности се почињу успављивати и припремати за дуги међустандардизацијски сан....

// Ulaz ne sme biti INT MIN

```
int abs(int x) {...}

// Ulazni opseg mora biti uredjen neopadajuce
template<typename It, typename T >
bool binary search(It b, It e, const T& val) {...}
```

...неке могућности се почињу успављивати и припремати за дуги међустандардизацијски сан....



```
// Ulazni opseg mora biti uredjen neopadajuce
template<typename It, typename T >
bool binary_search(It b, It e, const T& val)
   pre{std::is_sorted(b, e)}
{...}
```

# ...али стандард доноси и много шарених, ситних радости.

```
// Ako je suma >= 1, bice suma na izlazu
// Tokom jeseni suma je pozutela
// Kolac ti je spremljen!
// C++
```

# ...али стандард доноси и много шарених, ситних радости.

```
// Ако је сума >= 1, буће шума на излазу
// Током јесени шума је пожутела
// Колач ти је спремљен!
// Ч++
```



И то је зашто ја волим када Це++ стандард дође у мој крај.