Programowanie

Dariusz Wardecki, wyk IX

Powtórzenie

Znajdź błąd

```
struct A
{
  int n;
  char t[10];
};

int main()
{
  A obiekt;
  obiekt->n = 10;
  obiekt->t = "Hello!!!";

  ...
}
```

Powtórzenie

Co zostanie wypisane?

```
int main()
{
  char tablica[6];
  tablica = "Ala ma kota.";

  cout << tablica << endl;
  return 0;
}
```

Rozmiar struktury

W przypadku tablicy:

```
int t[10];
cout << sizeof(t) << endl;
```

W przypadku struktury:

```
struct A{
  float x, y;
  char z[5];
};
...
cout << sizeof(A) << endl;
```

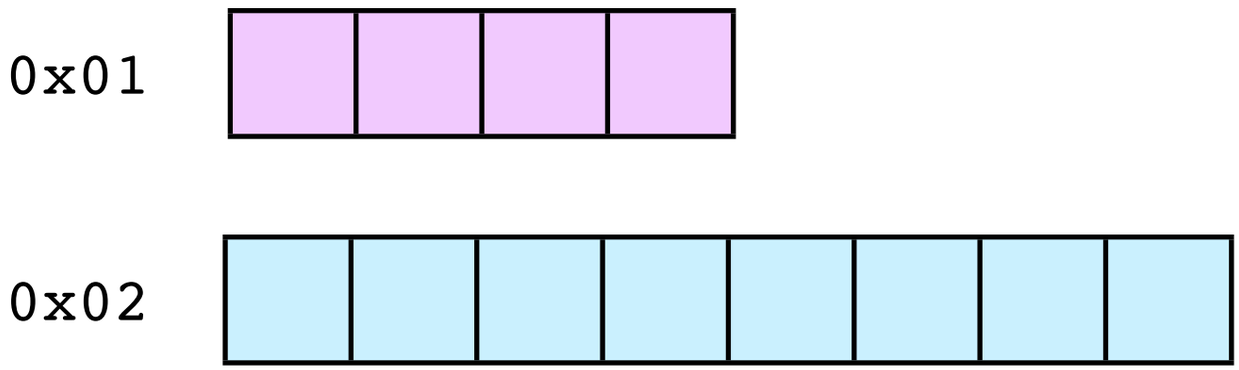
Rzutowanie struktury

```
struct B{
  int i, j;
};
int main(){
  char tab[8] = {0,0,0,0,1,0,0,0};
  B *b = (B*)(tab);
  cout << b->i << " " << b->j << endl;
  return 0;
}
```

Unie

```
struct U{
  int x;
  double y;
};
```

```
union U{
  int x;
  double y;
};
```



Klasa vector

```
#include <vector>

vector<int> first;                 // empty vector of ints
vector<int> second (4,100); // four ints with value 100
vector<int> third (second); // kopia second
```

Klasa vector

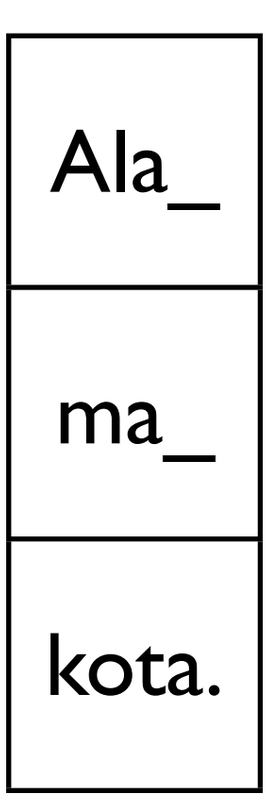
Typy zmiennych dla wektorów

```
#include <vector>

vector<char> first;
vector<int> second;
vector<long> third;
vector<string> third;

… itp
```

Klasa vector

`vector<string>` vs C-string

Ala_ma_kota.



Klasa vector

operator []

```
vector<int> myvector (10);
cout << myvector[2];
```

Klasa vector

push_back()

```
int main ()
{
  vector<int> myvector;
  int myint;

  cout << "Please enter some integers (enter 0 to end):\n";

  do {
    cin >> myint;
    myvector.push_back (myint);
  } while (myint);

  cout << "myvector stores " << int(myvector.size()) << " numbers.\n";

  return 0;
}
```

Klasa vector

pop_back()

```
int main ()
{
  vector<int> myvector;
  int sum = 0;
  myvector.push_back (100);
  myvector.push_back (200);
  myvector.push_back (300);

  while (!myvector.empty())
  {
    sum+=myvector.back();
    myvector.pop_back();
  }

  std::cout << "The elements of myvector add up to " << sum << '\n';

  return 0;
}
```

Klasa vector

size()

```
int main ()
{
  vector<int> myints;
  cout << "0. size: " << myints.size() << '\n';

  for (int i=0; i<10; i++) myints.push_back(i);
  cout << "1. size: " << myints.size() << '\n';

  myints.insert (myints.end(),10,100);
  cout << "2. size: " << myints.size() << '\n';

  myints.pop_back();
  cout << "3. size: " << myints.size() << '\n';

  return 0;
}
```

```
0. size: 0
1. size: 10
2. size: 20
3. size: 19
```

Klasa vector

empty()

```
int main ()
{
  vector<int> myvector;
  int sum (0);

  for (int i=1;i<=10;i++) myvector.push_back(i);

  while (!myvector.empty())
  {
     sum += myvector.back();
     myvector.pop_back();
  }

  cout << "total: " << sum << '\n';

  return 0;
}
```

Klasa vector

swap()

```
int main ()
{
  vector<int> foo (3,100);   // three ints with a value of 100
  vector<int> bar (5,200);   // five ints with a value of 200

  foo.swap(bar);

  cout << "foo contains:";
  for (unsigned i=0; i<foo.size(); i++)
    cout << ' ' << foo[i];
  cout << '\n';

  cout << "bar contains:";
  for (unsigned i=0; i<bar.size(); i++)
    cout << ' ' << bar[i];
  cout << '\n';

  return 0;
}
```

Klasa vector

data()

```
int main ()
{
  std::vector<int> myvector (5);

  int* p = myvector.data();

  *p = 10;
  ++p;
  *p = 20;
  p[2] = 100;

  std::cout << "myvector contains:";
  for (unsigned i=0; i<myvector.size(); ++i)
    std::cout << ' ' << myvector[i];
  std::cout << '\n';

  return 0;
}
```

```
myvector contains: 10 20 0 100 0
```