

Технологии разработки алгоритмов решения инженерных задач

лекция №4

Преподаватель: Дреев Александр Николаевич

4. Динамические массивы, списки.

4.1. Тип массива, списка. Разделение кода программы на управление данными и функциональность.

4.2. Оценивания сложности и среднего времени выполнения основных операций.

4.3. Списки. Основные операции над списком.

4.4. Оценка времени выполнения основных операций.

4.5. Сортировка массива, вектора и списка. Алгоритмы сортировки и их сложность.

[illegible]

Правила оформления программы

Массив, список

список - участок памяти, содержащая записанные однородные данные в произвольном расположении.

```
#include <stdio.h>
```

```
class Slist { int
```

```
data;
```

```
int month;
```

```
int year;
```

```
Slist * nextData; };
```

Правила оформления программы

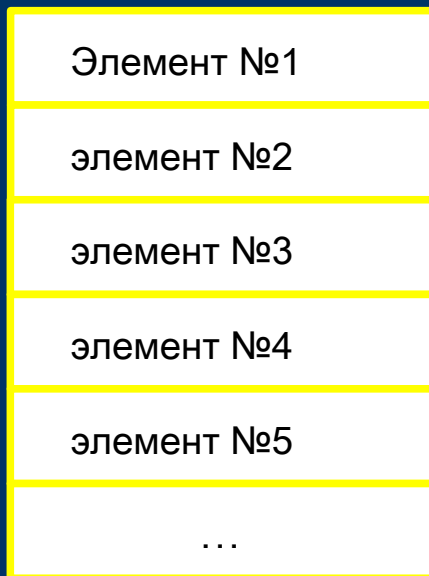
Массив, список

```
int main ( int N, char ** Param) {Slist  
  
    fierst;  
  
    fierst.data = 24;  
    fierst.month = 9;  
    fierst.year = 2012;  
    fierst.nextData = new Slist;  
    printf ( "Data: day% i, month% i, year% i \ n",  
    fierst.data, fierst.month, fierst.year)  
    delete fierst.nextData;  
    return 0; }
```

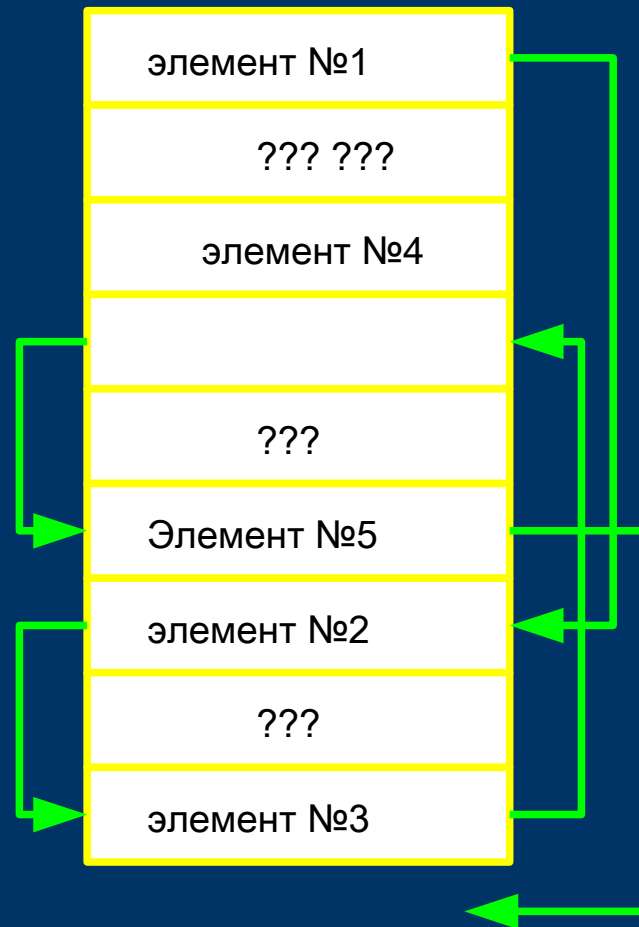
Правила оформления программы

Массив, список

МАССИВ



СПИСОК



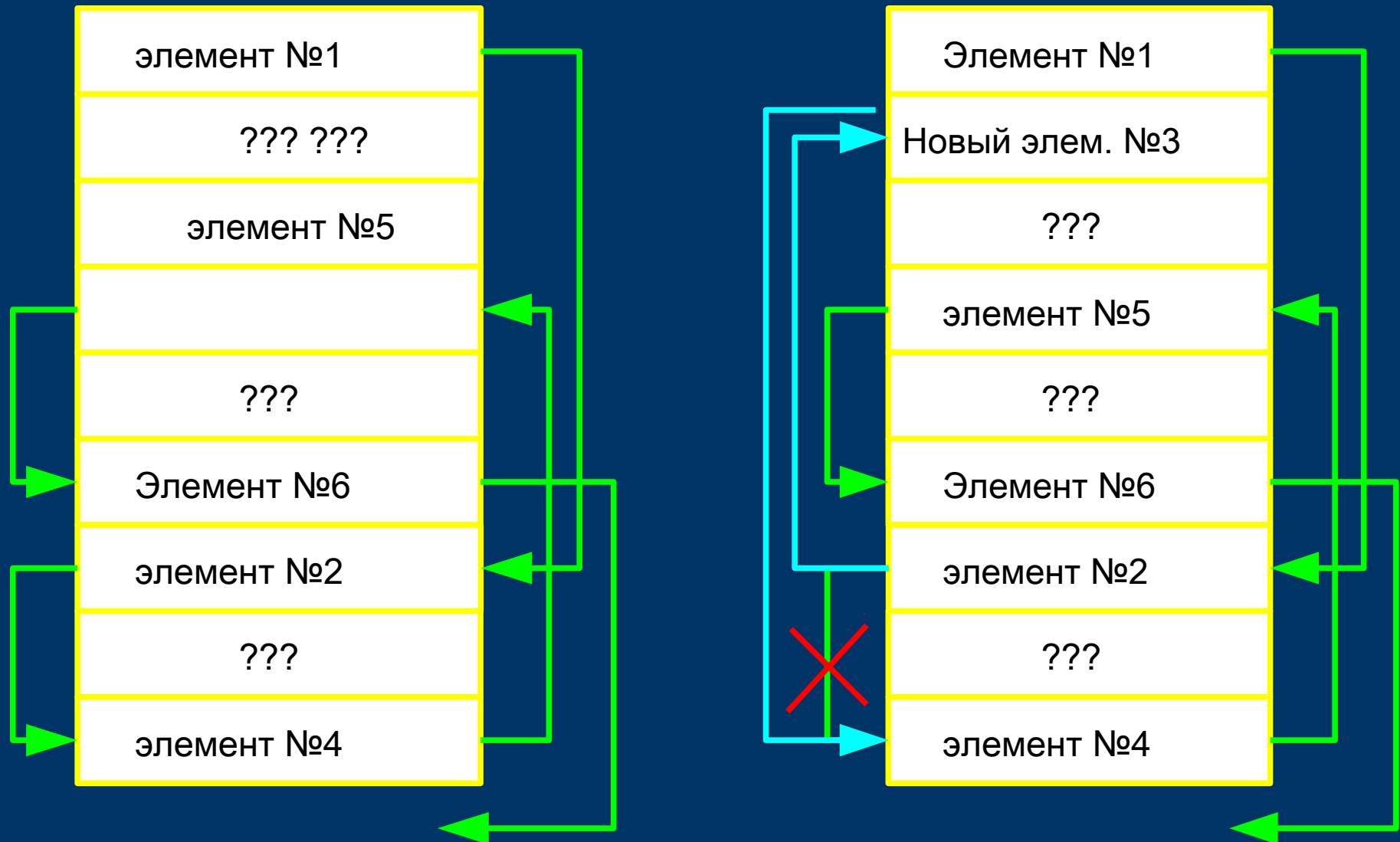
Правила оформления программы

Массив, список

МАССИВ	Список
Сплошной блок памяти	Каждый элемент имеет свой блок памяти
Быстрый доступ к каждому элементу	Для доступа нужно пройти всю цепочку
Элемент не добавляется, нужно делать запас	Элементы легко вставляются и удал.
Очень простое пользование	Более сложное пользования
быстро работает	Гибкий в использовании

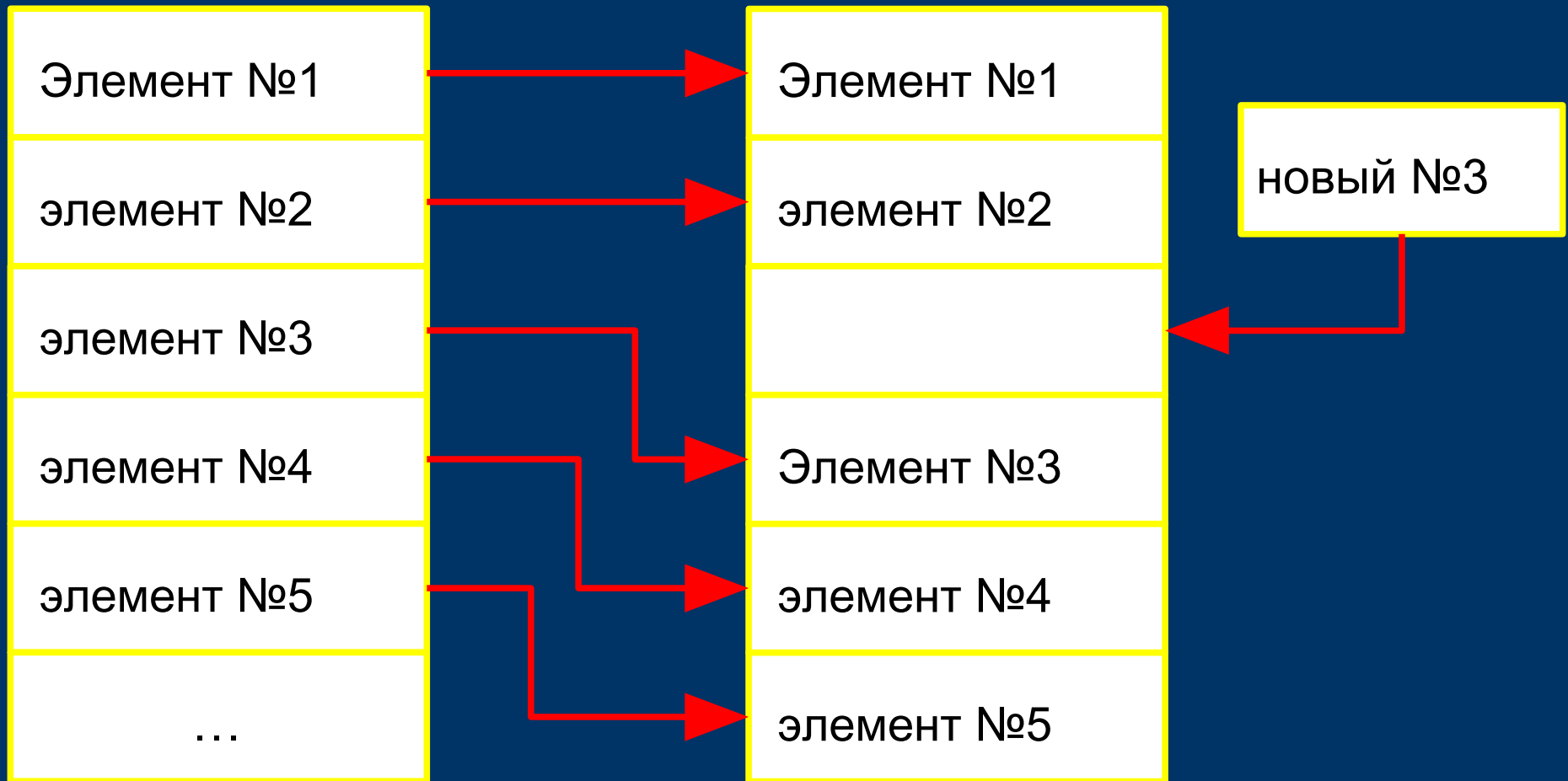
Правила оформления программы

Список вставка элемента



Правила оформления программы

Массив, вставка элемента



Правила оформления программы

Массив, список - использование std

```
#include <iostream>
```

```
#include <vector>
```

```
using namespace std;
```

```
int main () {
```

```
    vector < double > Chisla;
```

```
    int i = 0;
```

```
    for (l = 0; i < 10; i ++)
```

```
    {Chisla.push_back (1.0 + i) }
```

```
    for (l = 0; i < chisla.size (); i ++) cout << chisla [i] << ""; cout << endl;
```

```
    vector < double > :: iterator element = chisla.begin () + 4, chisla.erase (element)
```

```
    for (l = 0; i < chisla.size (); i ++) cout << chisla [i] << ""; cout << endl; chisla.clear ();
```

```
    return 0; }
```

Правила оформления программы

Массив, список - использование std

```
#include <iostream>
```

```
#include <list>
```

```
using namespace std;
```

```
int main () {
```

```
    list < double > Chisla;
```

```
    unsigned int i = 0;
```

```
    for (i = 0; i < 10; i++) {chisla.push_back (i) }
```

```
    list <double> :: iterator L;
```

```
    for (L = chisla.begin (); L != Chisla.end (); L++)
```

```
        cout << * L << " "; cout <<
```

```
endl;
```

```
    L = chisla.begin (); L++; L++; L++; L++; chisla.erase (L);
```

```
    for (L = chisla.begin (); L != chisla.end (); L++)
```

```
        cout << * L << " "; cout <<
```

```
endl; return 0; }
```