Технології розробки алгоритмів розв'язання інженерних задач

Лекція №4

Викладач: Дрєєв Олександр Миколайович

4. Динамічні масиви, списки.

4.1. Тип масиву, списку. Поділ коду програми на керування даними та функціональність.

4.2. Оцінювання складності та середнього часу виконання основних операцій.

4.3. Списки. Основні операції над списком.

4.4. Оцінювання часу виконання основних операцій.

4.5. Сортування масиву, вектору та списку. Алгоритми сортування та їх складність.

Правила оформлення програми

Масив, список

Масив — ділянка пам’яті, яка містить послідовно записані однорідні дані.

```
class Sdata{
  char date;
  char month;
  int    year;};
Sdata* dates[10]; dates[0].date=24;
dates[0].month=9; dates[0].year=2012;
```

№	...	57	58	59	60	61	62	63	...
Зн.	?	?	24	9	7	220	9	?	?

Правила оформлення програми

Масив, список

Список — ділянка пам’яті, яка містить записані однорідні дані в довільному розташуванні.

```
#include <stdio.h>
class Slist
{
  int data;
  int month;
  int year;
  Slist* nextData;
};
```

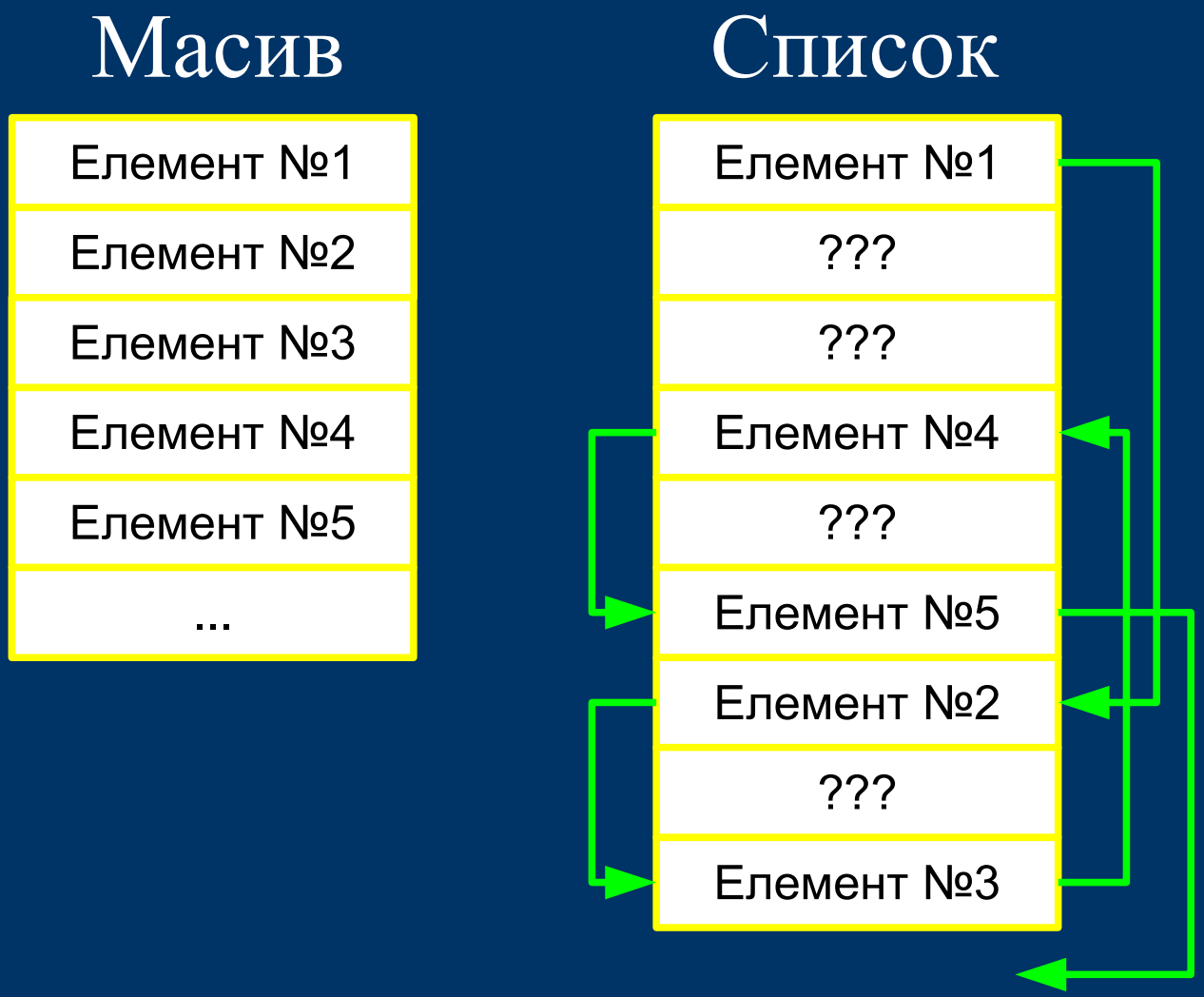
Правила оформлення програми

Масив, 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 оформлення програми

Масив, список



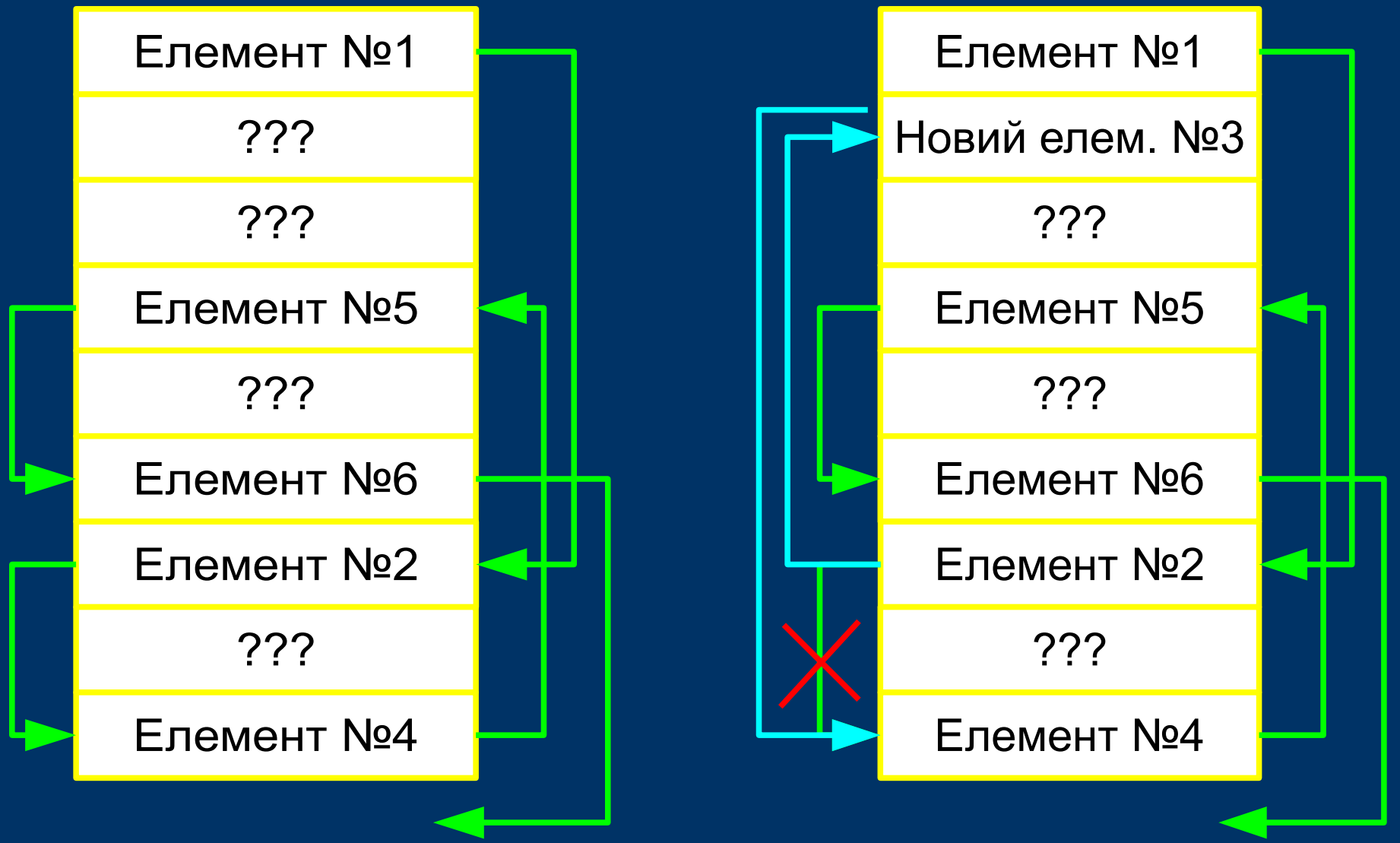
Правила оформлення програми

Масив, список

МАСИВ	СПИСОК
Суцільний блок пам’яті	Кожен елемент має свій блок пам’яті
Швидкий доступ до кожного елементу	Для доступу потрібно пройти весь ланцюг
Елемент не додається, потрібно робити запас	Елементи легко вставляються та вилуч.
Дуже просте користування	Більш складне користування
Швидко працює	Гнучкий в використанні

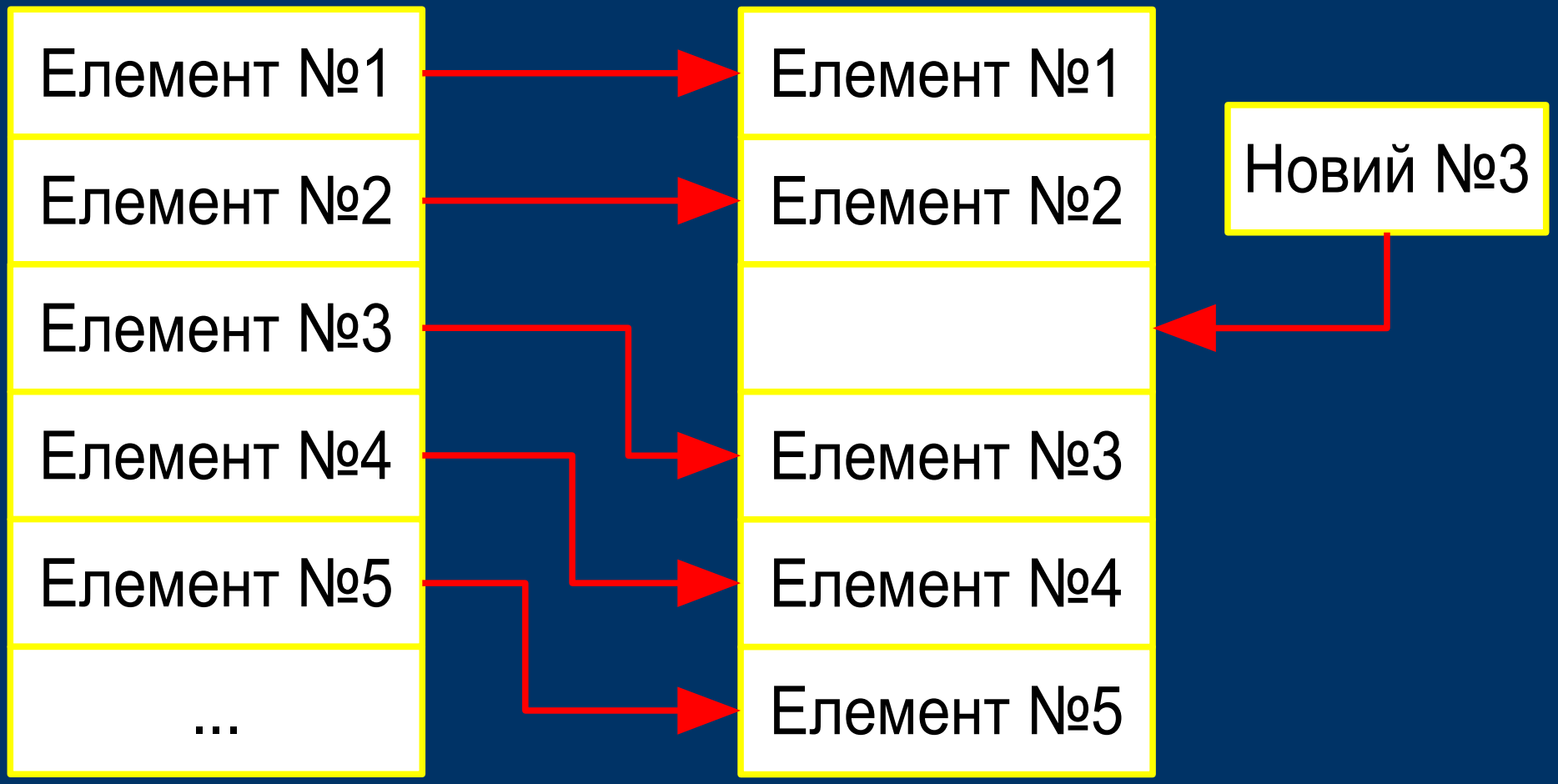
Правила оформлення програми

Список, вставка елементу



Правила оформлення програми

Масив, вставка елементу



Правила оформлення програми

Масив, список — використання std

```
#include <iostream>
#include <vector>
using namespace std;
int main()
{    vector<double> chisla;
     int i = 0;
     for( i=0; i<10; i++)
     { chisla.push_back(1.0+i); }
     for(i=0;i<chisla.size();i++) cout << chisla[i] << " ";
     cout << endl;
     vector<double>::iterator element = chisla.begin()+4;
     chisla.erase(element);
     for(i=0;i<chisla.size();i++) cout << chisla[i] << " ";
     cout << endl;
     chisla.clear();
     return 0;
}
```

Правила оформлення програми

Масив, список — використання 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;
}
```