



МАСИВИ

1. Структурни типове данни

Структура от данни - организирана информация, която може да бъде описана, създадена и обработена с помощта на програма.

Скаларни типове данни:

Целочислен – int

Реален – double

Логически – bool

Символен – char

Съставни типове данни:

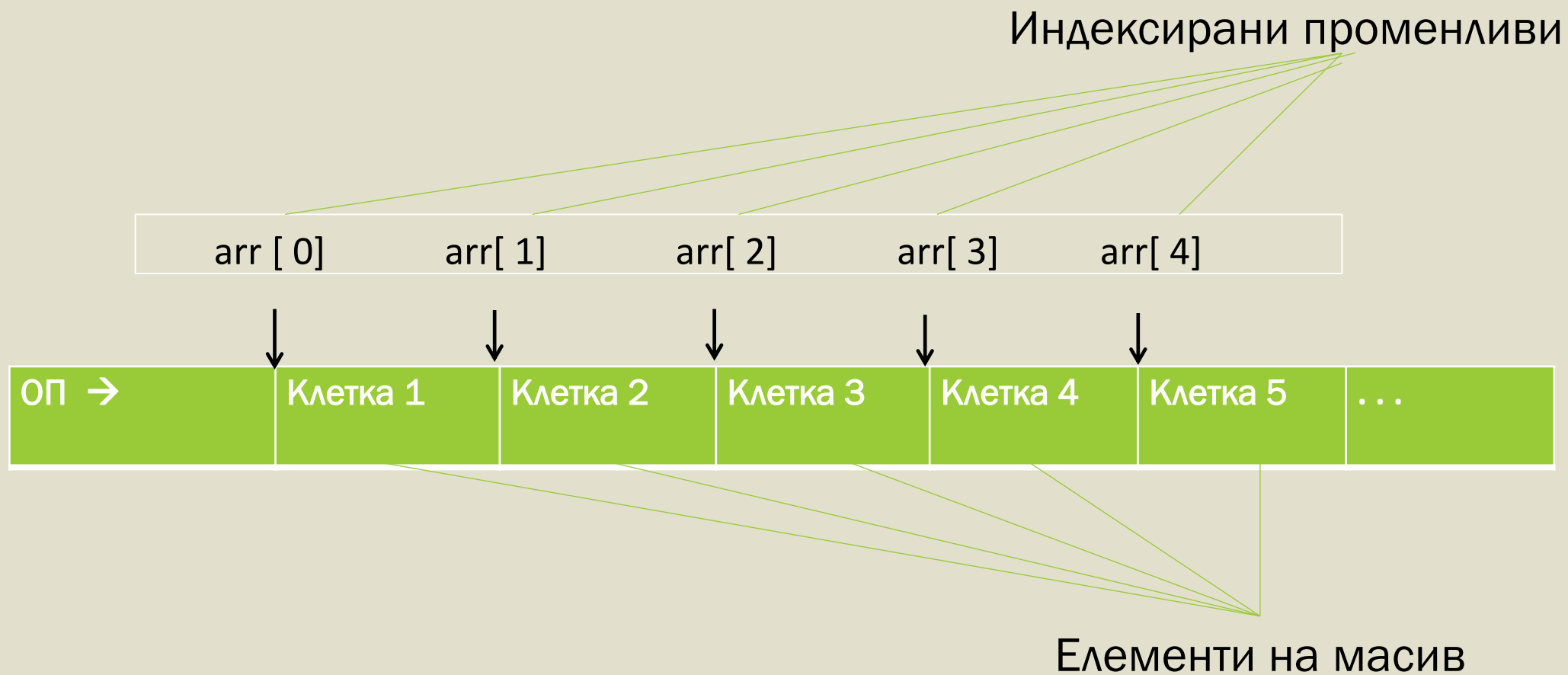
Масив

Символен низ

Структура

2. Структурен тип данни – масив

крайна редица от еднотипни елементи с пряк достъп до всеки елемент

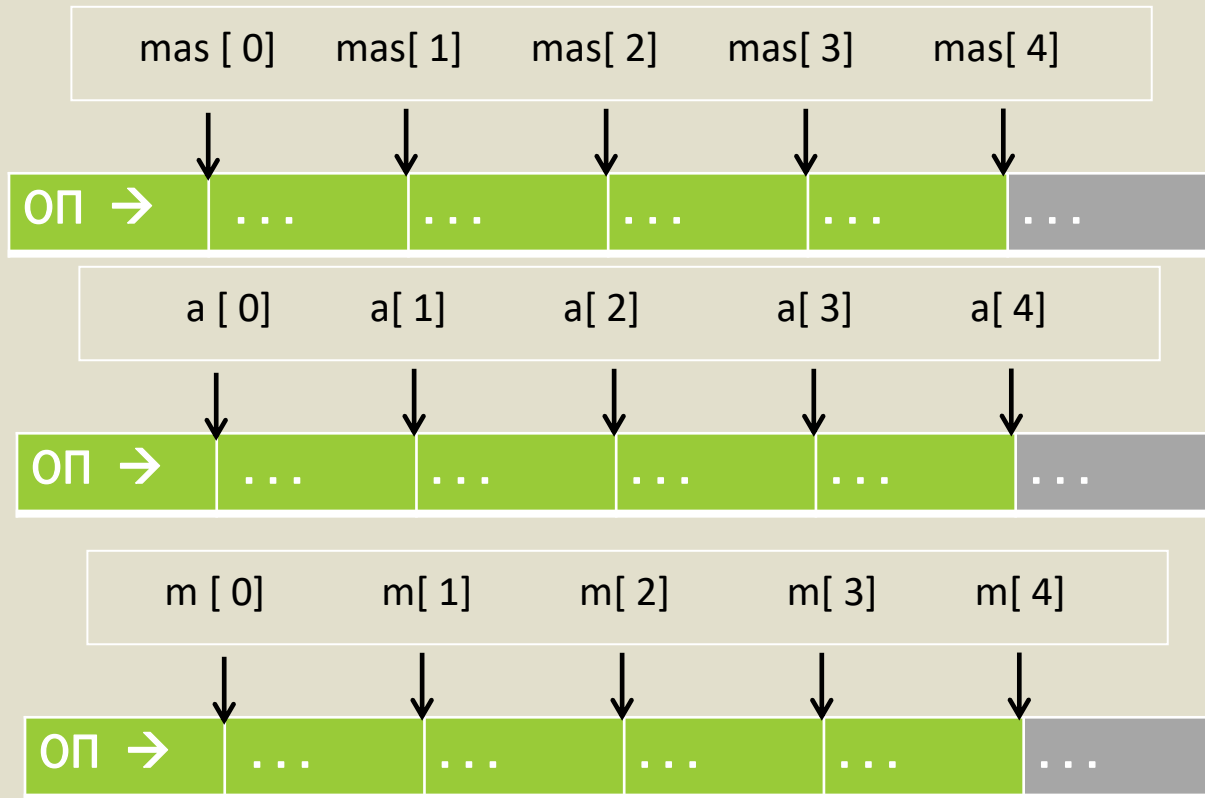


3. Деклариране на масив

Синтаксис: <тип> <име_на_масив> [<брой_елементи>];

- <тип> Стандартен или структурен
- <име_на_масив> Идентификатор
- <брой_елементи> Константа, задаваща максималния брой елементи

3. Деклариране на масив - примери



int mas[5];

В ОП се заделя място за 5 последователни клетки по 4 байта.

double a[5];

В ОП се заделя място за 5 последователни клетки по 8 байта

char m[5];

В ОП се заделя място за 5 последователни клетки по 1 байт.

Броят елементи може да бъде зададен и с константа, която е предварително декларирана.

const int n=3

Long my_array[n]

4. Инициализация на масив

■ Първи начин:

```
int arr[5]
```

```
arr[0] = 5;
```

```
arr[1] = 19;
```

```
arr[2] = -22;
```

```
arr[3] = 43;
```

```
arr[4] = -6;
```

■ Втори начин:

```
int arr[5] = {5,19,-22,43,-6};
```

■ Трети начин:

```
int arr[ ] = {10,99,77};
```

■ Четвърти начин:

```
double arr[5];
```

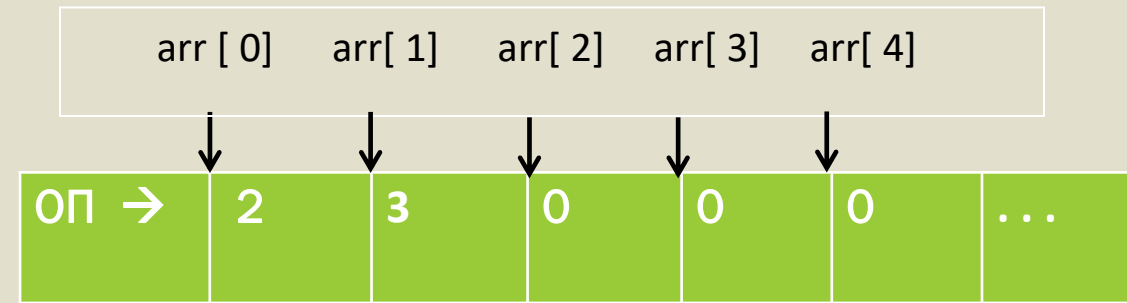
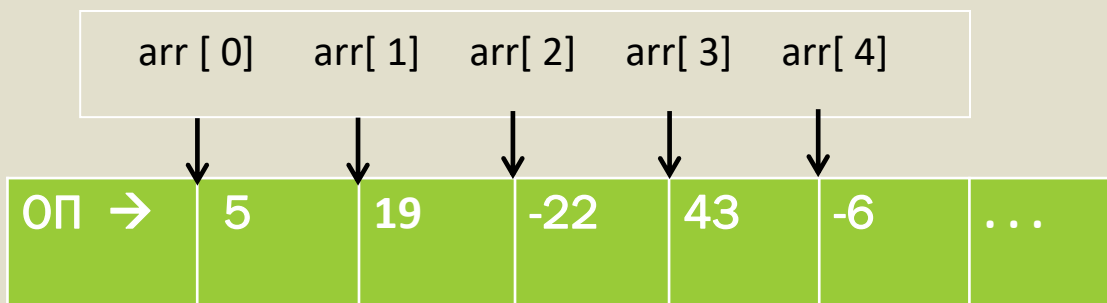
```
for(int i=0;i<5;i++)
```

```
{
```

```
    cout<<"arr["<<i<<"]=">>arr[i];
```

```
}
```

```
int arr[5] = {2,3};
```



5. Въвеждане стойностите на елементите на масив

```
int arr[5];  
for(int i=0;i<5;i++)  
{  
cout<<"arr["<<i<<"] =";cin>>arr[i];  
}
```

```
int arr[100] , n;  
cout<<"n=";cin>>n;  
for(int i=0;i<n;i++)  
{  
cout<<"arr["<<i<<"] =";cin>>arr[i];  
}
```

6. Извеждане стойностите на елементите на масив

```
int arr[5];
```

```
for(int i=0;i<5;i++)
```

```
cout<<"arr["<<i<<"]="<<arr[i]"\n";
```


7. Основни алгоритми за работа с масиви.

■ Задача №1

Да се състави програма, която въвежда от клавиатурата стойностите на 10 елемента на масив, които са цели числа. Да се изведе сборът им.

Решение – задача №1

```
#include<iostream>
using namespace std;
int main()
{
int arr [ 10] , i;
for (i=0;i<10;i++)
{
cout<<"arr["<<i<<"] =";cin>>arr[ i];
}
int sum=0;
for(i=0;i<10;i++)sum=sum+arr[ i];
cout<<"sum="<<sum<<endl;
return 0;
}
```

Примерен вход:

1,2,3,4,5,6,7,8,9,10

Примерен изход:

Sum=55

7. Основни алгоритми за работа с масиви.

■ Задача №2

Да се състави програма, която въвежда в масив среднодневните температури за период от един месец. Да се преброят дните, в които средната температура е равна на предварително зададено число.

Решение – задача №2

```
#include<iostream>
using namespace std;
int main()
{
double arr[ 31] ,k;
int i,n;
cout<<„Въведи брой дни в месеца:";cin>>n;
for(i=0;i<n;i++) { cout<<“arr[”<<i<<“]=”; cin>>arr[ i];}
cout<<„Предварително зададена температура:";cin>>k;
int br=0;
for (i=0;i<n;i++)
if(arr[ i] ==k) br++;
cout<<"br="<<br<<endl;
return 0;
}
```

Примерен вход:

n=28; k=3

12,24,15,18,19,

32,7,9,13,15,18

19,17,16,13,10,

8,5,3,0,2,6,3,2,

5,7

Примерен изход:

br = 2

7. Основни алгоритми за работа с масиви.

■ Задача №3

Да се състави програма, която въвежда в масив средния годишен успех на всеки ученик от един клас. Да се изведе най-високият успех.

Решение – задача №3

```
#include<iostream>
using namespace std;
int main()
{
double arr[ 30];
int i,n;
cout<<„Въведи брой ученици:“;cin>>n;
for(i=0;i<n;i++)
{
cout<<“arr[”<<i<<“]=”; cin>>arr[i ];
}

double max=arr[ 0];
for (i=1;i<n;i++)
if (arr[ i]>max) max=arr[ i];
cout<<max;
return 0; }
```

Примерен вход:

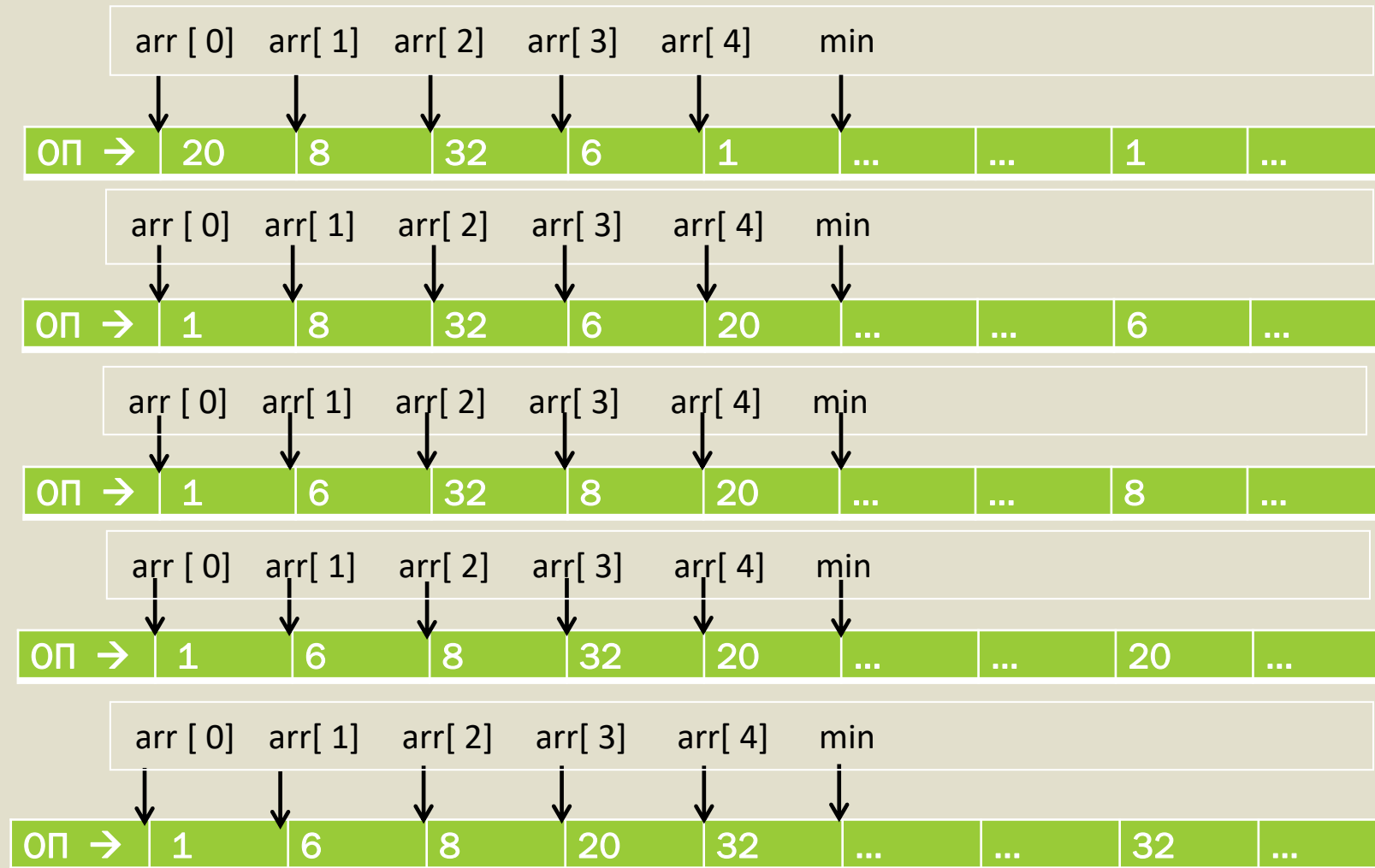
n=20

3.5,4.68,5.63,5.15,6.00,
5.35,5.63,4.32,3.56,4.5,
5.5,4.2,4.7,4.9,5.15,5.3
5.5,4.5,3.4,2.3,3.8

Примерен изход:

6.00

8. Сортиране на масиви – метод на пряка селекция



8. Сортиране на масиви – по метода на пряка селекция

■ Задача № 4

Даден е масив от цели числа

$arr [n] = \{20,8,32,6,1\}$. Да се сортира масивът във възходящ ред и да се изведе на екрана.

Решение - задача №4

```
#include<iostream>
using namespace std;
int main()
{
const int n=5;
int arr[ n] ={20,8,32,6,1}, min, k,
swap;
for(int i=0;i<n-1;i++)
{
k=i;
min=arr[ i];
for(int j=i+1;j<n;j++)
if(arr[ j]<min)
{
min=arr[j];
k=j;
}
swap= arr[ k];
arr[ k]=arr[ i];
arr[ i]=swap;
}
for(int k=0;k<n;k++)
cout<<"arr["<<k<<"]="<<arr[ k]<<endl;
return 0;
}
```

8. Сортиране на масиви - по метода на мехурчето

■ Задача № 5

Даден е масив от цели числа

$arr [n] = (182, 128, 32, 67, 158)$. Да се сортира

масивът във възходящ ред и да се изведе на екрана

Решение - задача №5

```
#include<iostream>
using namespace std;
int main()
{
const int n=5;
int arr[ n] ={182,128,32,67,158},
k, swap;
for(int i=0;i<n-1;i++)
for(int j=0;j<n-i-1;j++)
if(arr[ j]>arr[ j+1] )
{
swap=arr[ j];
arr [ j]=arr [ j+1];
arr [ j+1]=swap;
}
for(int k=0;k<n;k++)
cout<<"arr["<<k<<"]="<<arr[ k]<<endl;
return 0;
}
```