

Указатели и псевдонимы

Тип указател

- MS: всички възможни lvalue от даден тип и специалната стойност NULL
- Интегрален нечислов тип
- Параметризиран тип: ако T е някой тип, T* е тип “указател към T”
- Физическо представяне: цяло число с размера на машинната дума, указващо адреса на указваната lvalue в паметта

Операции с указатели

- рефериране (&<lvalue>)
- дерефериране (*<указател>) - **унарен!**
- указателна аритметика (+, -, +=, -=, ++, --)
- сравнение (==, !=, <, >, <=, >=)
- извеждане (<<)
- **няма въвеждане (>>) !**

Дефиниране на указателни променливи

- `<тип> *<идентификатор> [= <израз>]`
`{, *<идентификатор> [= <израз>] };`

Примери:

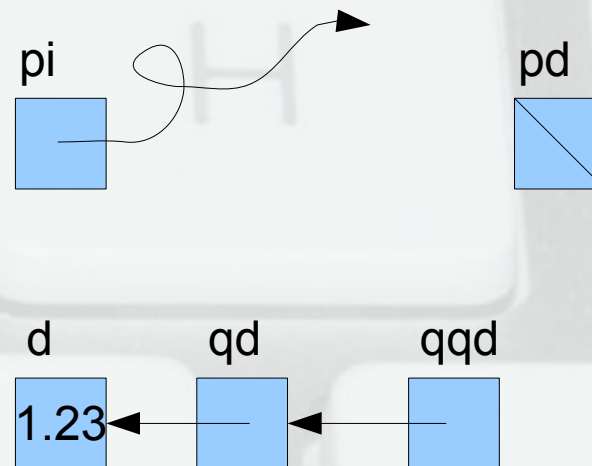
```
int *pi;
```

```
double *pd = NULL;
```

```
double d = 1.23;
```

```
double *qd = &d;
```

```
double **qqd = &qd;
```

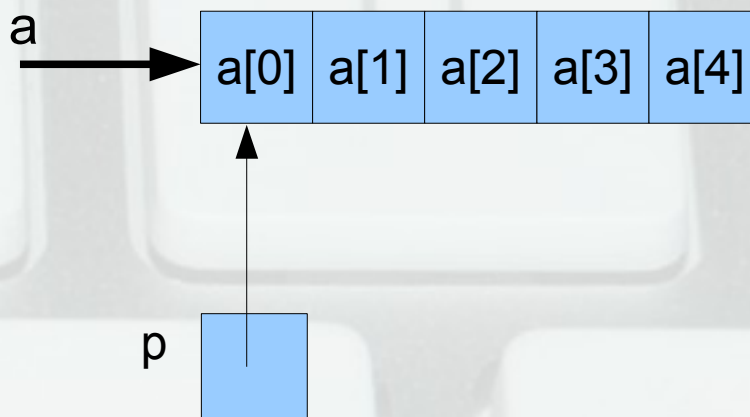


Рефериране и дерефериране

- `int x = 5, *p = &x, *q = p, y = *p + 2;`
- `*p++; p = &y; *q = 1; *p = *q;`
- **Внимание:**
 - `<lvalue>` е `rvalue!` (~~`&x = p`~~)
 - `*<rvalue>` е `lvalue!` (`*p = x`)
 - `&>(*p) ↔ p`
 - `*(&x) ↔ x`

Указатели и масиви

- Името на масив е **константен указател** към първия му елемент
- `int a[5];`
- `int* p = a;`
- `*p = 15; cout << a[0];`
- `*a = 20; a = p;`



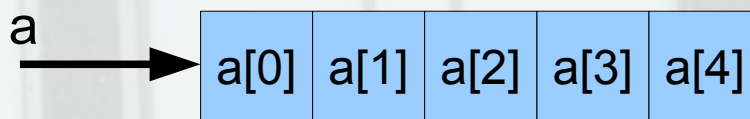
Указателна аритметика

- **sizeof**(<тип> | <израз>) — връща размера в байтове, заеман в паметта от <израз> или променлива от <тип>
- <указател> [**+**|**-**] <цяло число>
<цяло число> **+** <указател>
- $T^* p;$
 $p + i \leftrightarrow (T^*)((int)p + i * sizeof(T))$

Указатели и масиви

- Името на масив е **константен указател** към първия му елемент

- $a[i] \leftrightarrow *(a+i)$



```
int a[5], x;
```

```
cout << *a; *(a+1) = 7; *(a+4)--;
```

```
a++; a--; a = &x;
```

- Странно, но вярно: $a[i] \leftrightarrow *(a+i) \leftrightarrow *(i+a) \leftrightarrow i[a]$