

# Указатели

доц. д-р Нора Ангелова

---

# Оператор &

- Синтаксис

&<променлива>

<променлива> ::= дефинирана променлива.

- Семантика

Намира адреса на променливата.

# Указатели

$\langle T \rangle^* \langle \text{име\_променлива} \rangle [= \langle \text{стойност} \rangle]_{\text{опц}}$

- Семантика – дефинира указател към променлива от тип T.

$\langle T \rangle ::=$  име или дефиниция на тип.

$\langle \text{име\_променлива} \rangle ::=$  идентификатор.

$\langle \text{стойност} \rangle ::=$  шестнадесетично число, представляващо адрес на данна от тип T или `nullptr` (NULL).

Пример:

```
double *ptr = nullptr;
```

```
double myFirstDouble = 5;
```

```
ptr = &myFirstDouble;
```

```
myFirstDouble
```

```
5.3
```

```
адресY
```

```
ptr
```

```
адресY
```

```
адресX
```

# Оператор \*

- Синтаксис

\*<указател>

- Семантика

Намира стойността, която е записана на адреса, който е записан в указателят (стойността на елемента сочен от указателя).

# Указатели

```
int firstVar = 5;  
int secondVar = 3;  
int *ptrFirstVar = &firstVar;  
int *ptrSecondVar = &secondVar;
```



# Указатели

```
int firstVar = 5;  
int *ptrFirstVar = &firstVar;
```

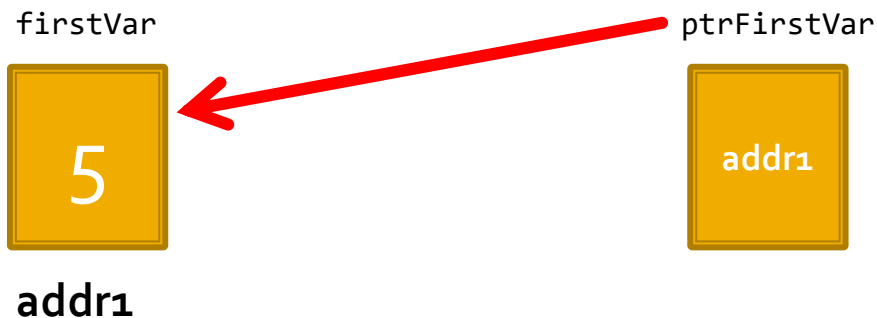
ТИП

```
firstVar == 5
```

(int)

```
&firstVar == addr1
```

(int \*)



ТИП

```
ptrFirstVar == addr1
```

(int \*)

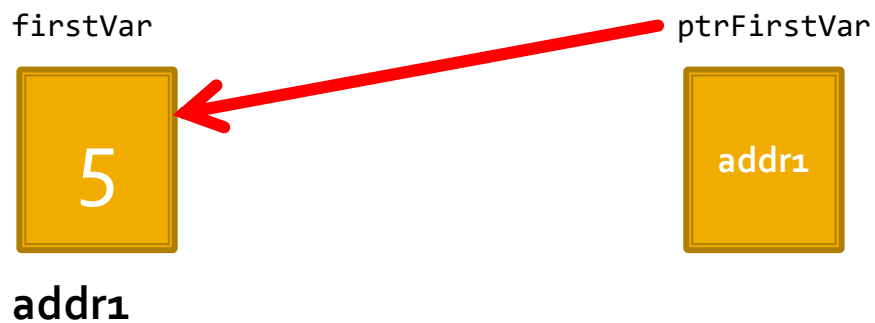
```
*ptrFirstVar == 5
```

(int)

# Указатели

```
int firstVar = 5;  
int *ptrFirstVar = &firstVar;
```

```
*ptrFirstVar = 5; // Присвоява нова стойност на firstVar
```



# Указатели

- Забележка

Дефиницията:  $T^* a, b;$

е еквивалентна на:

$T^* a;$

$T b;$

Дефиницията:  $T^* a, b;$

е еквивалентна на:

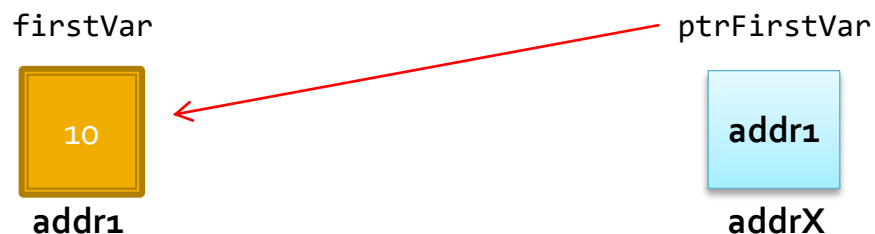
$T^* a;$

$T^* b;$

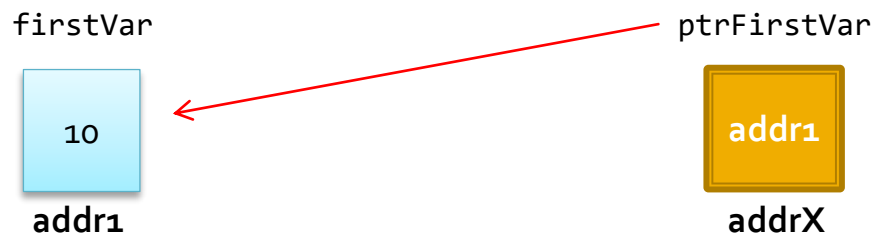


# Константи указатели и константни ст-сти

- Константен указател



- Указател към константна стойност – не може да се променя стойността през указателя



- Константен указател към константна стойност



```
int const * ptr;
```

ptr is a **pointer** to **const int**

```
const int * ptr;
```

ptr is a **pointer** to **int constant** (i.e. const int)

```
int * const ptr;
```

ptr is a **const pointer** to **int**

```
const int * const ptr;
```

ptr is a **constant pointer** to **const int**

# Аритметични и логически операции

- +, -, ++, --, ==, !=, >, >=, <, <=

*\* Не е възможно въвеждане на данни от тип указател чрез оператора >>.*

# Адресна аритметика

```
int *intPtr;  
double *doublePtr;
```

- `intPtr = intPtr + 1; // премества указателя`

```
intPtr = intPtr + 1*4,
```

4 е броят на байтовете, необходими за записване на данна от тип `int`.

- `doublePtr = doublePtr + 1; // премества указателя`

```
doublePtr = doublePtr + 1*8,
```

8 е броят на байтовете, необходими за записване на данна от тип `double`

Следва продължение...