

Въведение в курса.

Програми, езици от високо ниво, бройни системи, Backus-Naur, променливи, вход и изход, условен оператор, тернарен оператор, switch, цикли.

Изготвил: гл.ас. д-р. Нора Ангелова

Бройни системи

- **Число (0x10)**

$$2\ 3\ 4 = 2 * 10^2 + 3 * 10^1 + 4$$

Какво става, ако имаме не 10, а 16 цифри

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

- **Число (0x16)**

$$2\ 3\ 4 = 2 * 16^2 + 3 * 16^1 + 4$$

Ами ако имаме само две цифри?

- **Число (Binary)**

$$1\ 0\ 1 = 1 * 2^2 + 0 * 2^1 + 1$$

Backus-Naur

$\langle \text{digit} \rangle ::= 0 | 1 | \dots | 9$

$\langle \text{int} \rangle ::= \langle \text{digit} \rangle \mid \langle \text{int} \rangle \langle \text{digit} \rangle$

Променливи

- Типове

`int` – цели числа

`double` – реални числа

`char` – СИМВОЛ

`bool` – булева стойност (`true(1)/false(0)`)

- Стойност

```
int a = 5;
```

```
int b = 10;
```



Променливи

- Адрес

- Присвояване на стойност

- Оператор за присвояване

```
int c = a + b;
```

- Оператор за вход и изход

```
cin >> <променлива> {>> <променлива>}опц;
```

```
cout << <променлива> {<< <променлива>}опц;
```

Аритметични оператори

- + (събиране)
- - (изваждане)
- * (умножение)
- / (целочислено деление)
- % (остатък от целочислено деление)

Логически оператори

- Оператор за логическо умножение (конюнкция)

A	B	A && B
false	false	false
false	true	false
true	false	false
true	true	true

Логически оператори

- Оператор за логическо събиране (дизюнкция)

A	B	A B
false	false	false
false	true	true
true	false	true
true	true	true

- Оператор за логическо отрицание

A	! A
false	true
true	false

Оператори за сравнение

- == - сравнение за равно
- != - сравнение за различно
- > - сравнение за по-голямо
- >= - сравнение за по-голямо или равно
- < - сравнение за по-малко
- <= - сравнение за по-мало или равно

Условен оператор

- `if (<условие>) <оператор>`
 - `if` – запазена дума
 - `<условие>` - булев израз
 - `<оператор>` - произволен оператор

Пример:

```
if (a < 3) {  
    cout << "a e < 3" << endl;  
}
```

Оператор if/else

- `if (<условие>) <оператор1>`
 `else <оператор2>`
 - `if` – запазена дума
 - `<условие>` - булев израз
 - `<оператор1>` и `<оператор2>` - произволни оператори

Пример:

```
if (a < 3) {  
    cout << "a e < 3" << endl;  
} else {  
    cout << "a e >= 3" << endl;  
}
```

Вложени условни оператори

- `if (<условие>) <оператор1>`
 `else <оператор2>`
 - `<оператор1>` и `<оператор2>` са произволни оператори за управление на изчислителния процес, в това число могат да бъдат условни оператори

Пример:

```
if (a > 4) {  
    b = 5;  
} else if (a < 4) {  
    b = -5;  
} else b = 0;
```

Тернарен оператор

- (<условие>) ? <оператор1> : <оператор2>

Пример:

```
int a = 5;
```

```
int b = 3;
```

```
int larger = (a > b) ? a : b;
```

Оператор switch

- `switch(<израз>) {`
 `case <израз1> : <редица от оператори 1>`
 `case <израз2> : <редица от оператори 2>`
 `...`
 `[default: <редица от оператори n>]опц`
 `}`
- `break` – прекратява изпълнението на най-вътрешния, съдържащ го оператор `switch` или оператор за цикъл.

Масиви

- $T \text{ <променлива>[size] = \{<редица от константни изрази>\}_{опц}$
 - T - име или дефиниция на произволен тип, различен от псевдоним, `void` или функционален
 - <променлива> - идентификатор
 - $size$ – константен израз от интегрален или изброен тип с положителна стойност
 - $\text{<редица от константни изрази> ::=$
 $\text{<константен израз>} \mid$
 $\text{<константен израз>, <редица от константни изрази>}$

Константните изрази са от тип T или от тип съвместим с него.

Пример:

```
int a[100];
```

Масиви

- `int a[100]`
 - Индекс – $i \in [0, 99]$
 - Достъп до елементите на масив
 - Въвеждане и извеждане на елементи на масив

```
cin >> a[i]; cout << a[i];
```


Оператори за цикъл

- for

```
for(<инициализация>;<условие>;<корекция>)  
    <оператор>
```

- while

```
while(<условие>)<оператор>
```

- do/while

```
do
```

```
<оператор>
```

```
While(<условие>)
```



Благодаря за вниманието