Типове и променливи

доц. д-р Нора Ангелова

Памет

- Програмен стек
- Динамична памет (*Heap*)

Памет

- Редица от елементи със стойност 0 и 1, наречени битове.

- Клетка от паметта (дума) – групиране на няколко бита.

Памет

Измерва се в:

- bit
- байт (B) – 8bit
- килобайт (KB) – 2^{10}B ~1000B
- мегабайт (MB) – 2^{20}B
- гигабайт (GB) – 2^{30}B
- терабайт (TB) – 2^{40}B

Памет

- Да си заделим памет

място → адрес

име → идентификатор

големина → тип

Идентификатор

- Последователност от букви, цифри и _.
- Не може да започва с цифра.
- Не може да се използват ключовите думи в езика.

Пример:

```
index
nextMonthIndex
firstCharacter
```

- Езикът различава главните от малките латински букви.

Пример:

```
index и Index // различни идентификатори
```

- **Показват значението на съответната променлива/функция.**

Идентификатор

- Конвенции за именуване:

```
nextIndex;  // camel case

NextIndex;  // pascal case

next_index; // snake case
```

Типове

- Скаларни

int – чели числа;
double – реални числа;
float – реални числа;
char – символ;
bool – булева стойност (true/false);
enum – изброен;
void – празен тип
auto – *ще го използваме по ООП и СДП*
Указател и псевдоним;

- Съставни – масив, низ, вектор.

Типове

- http://www.cplusplus.com/doc/tutorial/variables/
- https://en.cppreference.com/w/cpp/language/types
- Един бит се използва за определяне на знака

** Стойностите могат да бъдат различни*

Да проверим размера на даден тип

```
#include <iostream>
using namespace std;

int main() {
  cout << "Size of char: " << sizeof(char) << endl;

  cout << "Size of int: " << sizeof(int) << endl;

  cout << "Size of float: " << sizeof(float) << endl;

  cout << "Size of double: " << sizeof(double) << endl;

  return 0;
}
```

Размер на променлива или тип

- Излизане извън границата на даден тип

Преобразуване на типове

```
double doubleVar = 2.3;
```

- Неявно преобразуване

```
int intVar = doubleVar; // 2
```

- Явно преобразуване
  - (тип)<израз> или тип (<израз>)

```
(int)(1.52 + 56.2) // 57
(double)(123 + 18) // 141.0
```

  - static_cast<тип>(израз)

```
static_cast<int>(1.52 + 56.2) // 57
static_cast<double>(123 + 18) // 141.0
```

Символен тип

- Състои се от крайно и наредено множество от символи

ASCII

```
(int)'F';    // 70
(char)65;   // A
```

- Символите могат да бъдат сравнявани

```
'F' < 'Z'      // true
```

Декларация на променлива

- <тип> <име_на_променлива>{, <име_на_променлива>}$_{опц}$;

<тип> ::= тип данни от изучените до момента;
<име_на_променлива> ::= идентификатор;

Пример:

```
int index;
int nextMonthIndex;
```

или

```
int index, nextMonthIndex;
```

Инициализация на променлива

- <тип> <име_на_променлива> = <стойност>
  {, <име_на_променлива> = <стойност>}$_{опц}$;

```
int nextIndex = 1;
```

- <тип> <име_на_променлива>(<стойност>)
  {, <име_на_променлива>(<стойност>)}$_{опц}$;

```
int nextIndex(1);
```

- Инициализация след декларация

```
int nextIndex;
//…
nextIndex = 3;
```

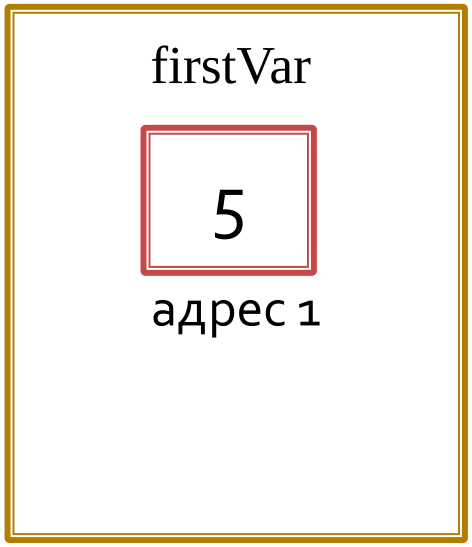
Дефиниция на променлива

Декларация + инициализация

Променливи

- Адрес
- Присвояване на стойност

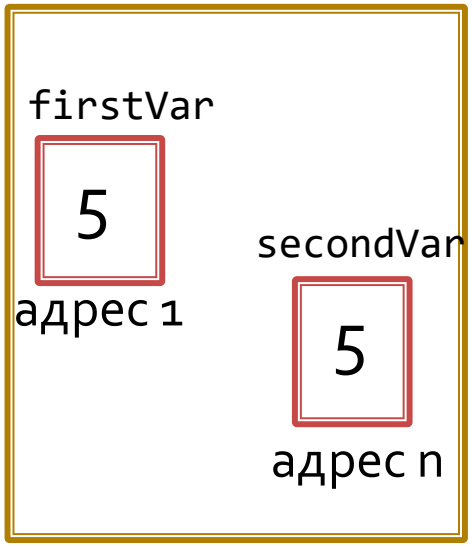
```
int firstVar = 5;
```



Променливи

- Адрес
- Присвояване на стойност

```
int firstVar = 5;
int secondVar = firstVar;
```



Константи

- const <тип> <име_на_променлива> = <стойност>;

```
// Декларация на константа
const float PI = 3.14;

PI = 3.1415 // Грешка
```

** Имената на константите често се изписват с главни латински букви.*

Област на променливи и константи

- Започва от нейната декларация/дефиниция.
- Продължава до края на блока (оператора).

```
    …
    double myVar;
    …
```

} – края на блока

Следва продължение ...