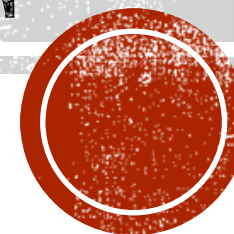


ЯЗЫКИ ПРОГРАММИРОВАНИЯ И МЕТОДЫ ТРАНСЛЯЦИИ



Лекция 8

6 апреля 2018 г.

ТАБЛИЦА СИМВОЛОВ

```
int a = 0, b;  
  
int fact(int n){  
    // ...  
}  
  
int main(){  
    int x, a;  
    return 0;  
}
```

SYMBOL TABLE

code	name	kind	type	len	default	scope
0	a	var	int	None	0	-1
1	b	var	int	None	None	-1
2	fact	func	int	1	None	-1
3	n	None	None	None	None	-1
4	n	var	int	None	None	2
5	main	func	int	0	None	-1
6	x	None	None	None	None	-1
7	x	var	int	None	None	5
8	a	var	int	None	None	5



ФУНКЦИИ ТАБЛИЦЫ СИМВОЛОВ

Создание временной ячейки памяти

```
int Alloc(int scope)
```

Объявление переменной/массива/функции

```
bool setIdType(int code, int scope, int type, int default = 0)
```

```
bool setArrType(int code, int scope, int type, int len)
```

```
bool setFuncType(int code, int type, int len)
```

Поиск и проверка переменной/массива/функции

```
bool checkId(int scope, int &code)
```

```
bool checkArr(int scope, int &code)
```

```
bool checkFunc(int scope, int &code, int len)
```



ФУНКЦИИ ТАБЛИЦЫ СИМВОЛОВ

```
bool setIdType(int code, int scope, int type, int default = 0)
```

Вызывается для объявления переменной или параметра функции. Параметры:

- `code` – числовой код идентификатора
- `scope` – контекст, в котором происходит объявление: `-1` для глобального или код функции для локального
- `type` – тип переменной
- `default` – значение по умолчанию

Находит в таблице символов имя переменной (`name`) с указанным `code`. После этого проверяет, если в таблице символов уже есть **объявленный** идентификатор с совпадающими `scope` и `name`, печатает соответствующее сообщение об ошибке и возвращает `False`. В противном случае объявляет переменную (возможно создавая новую строку) с указанными `name`, `scope`, `type` и `default`.



ФУНКЦИИ ТАБЛИЦЫ СИМВОЛОВ

```
bool setArrayType(int code, int scope, int type, int len)
```

Вызывается для объявления массива. Поведение – как и у setIdType.

Параметры:

- code – числовой код идентификатора
- scope – контекст, в котором происходит объявление
- type – тип массива
- len – размер массива

```
bool setFuncType(int code, int type, int len)
```

Вызывается для объявления функции. Поведение – как и у setIdType.

Параметры:

- code – числовой код идентификатора
- type – тип возвращаемого значения
- len – количество параметров



ПРИМЕР РАБОТЫ ОБЪЯВЛЕНИЯ ФУНКЦИЙ

```
int a = 0, b;  
  
int fact(int n){  
    // ...  
}  
  
int main(){  
    int x, a;  
    return 0;  
}
```

SYMBOL TABLE

code	name	kind	type	len	default	scope
0	a	var	int	None	0	-1
1	b	var	int	None	None	-1
2	fact	func	int	1	None	-1
3	n	None	None	None	None	-1
4	n	var	int	None	None	2
5	main	func	int	0	None	-1
6	x	None	None	None	None	-1
7	x	var	int	None	None	5
8	a	var	int	None	None	5



ОБЛАСТЬ ВИДИМОСТИ ПЕРЕМЕННЫХ

```
    0      1
int a = 0, b;

    2
int main(){
    3      1
    int x, b;
    3      0      1
    x = (2 + a) * b;
    return 0;
}
```

SYMBOL TABLE

code	name	kind	type	len	default	scope
0	a	var	int	None	0	-1
1	b	var	int	None	None	-1
2	main	func	int	0	None	-1
3	x	None	None	None	None	-1
4	x	var	int	None	None	2
5	b	var	int	None	None	2
6	[tmp1]	var	int	None	None	2
7	[tmp2]	var	int	None	None	2



ФУНКЦИИ ТАБЛИЦЫ СИМВОЛОВ

```
bool checkId(int scope, int &code)
```

Вызывается для проверки, является ли данный идентификатор объявленной переменной. Параметры:

- `scope` – код контекста
- `code` – код идентификатора (переданный по ссылке)

Функция работает по следующему алгоритму:

1. берет из таблицы символов `name` – имя идентификатора с кодом `code`
2. ищет в контексте `scope` идентификатор с именем `name`
1. если не находит и `scope > -1`, то повторяет поиск для `scope = -1`
3. если найденный идентификатор не объявлен, то возвращает ложь
4. если объявлен, но это не переменная, то возвращает ложь
5. иначе (это переменная и она объявлена) записывает ее `code` в параметр-ссылку (так как функция ищет переменную по имени в локальном и глобальном контекстах, то найденный `code` может не совпадать с переданным изначально) и возвращает истину



УПРАЖНЕНИЯ

Упражнение 1

Напишите таблицу символов и список атомов

```
int main() {
    int A, B, i, result = 1;
    in A; in B;
    for(i = A; i <= B; ++i) {
        result = result * i;
    }
    out "\nResult = ";
    out result;
}
```



УПРАЖНЕНИЯ

Упражнение 2

```
int buf[10];
int print(int a, int b, int c){
    out a;
    out b;
    out c;
}
int main(){
    int i, a;
    while(1){
        in buf[i];
    }
    print((i+2)-(a * print(1,2,print(4,5,6))), 1, 'c');
}
```

