

### Переменные

Все переменные имеют целочисленный тип. Считается, что все функции возвращают целочисленный тип. Нет неявного приведения `int -> boolean`.

Процедура инициализации переменных не предусмотрена. ID (имя переменной) - последовательность из букв английского алфавита, цифр и нижнего подчёркивания, не начинающаяся с цифры и не совпадающая с ключевыми словами.

### Конструкции

Цикл

```
while (LEXP) { тело цикла }
```

Ввод вывод

```
lout « ID;  
lin » ID;
```

Условные выражения

```
if (LEXP) {тело1} [else {тело2}]
```

Вызов функции

```
FUN_ID (аргументы (EXP) через запятую);
```

Объявление функции

```
define FUN_ID() {1}
```

Возвращаемое значение

```
return EXP;
```

Присвоение

```
ID := EXP;
```

### Выражения

Выражения бывают двух типов: арифметические (EXP) и логические (LEXP).

$EXP = ID \mid LITERAL \mid FUNCTION-CALL \mid (EXP) \mid EXP \text{ op } EXP$ ,

где op из  $\{+, -, *, /, \%\}$

$LEXP = EXP \text{ sr } EXP \mid LEXP \text{ op } LEXP \mid (LEXP)$ ,

где sr из  $\{<, >, <=, >=, ==, !=\}$ , а op из  $\{||, \&\&\}$

Выражение, дополненное точкой с запятой, является корректной самостоятельной инструкцией. Например, можно писать так:

```
while (a < 2) { 3 + 3; }
```