```
class Program{
	int var, integerId = 4;

	static void _main(){
		var = -23 - 22;
		print("N
		Sanity.");
		for(integerId in 0 to 100 steps 10){
			;
		}
	}

	// Comment.
	bool boolean(){
		real var_2 = 0.14;

		if(var < 8 && 32 - var > var_2){
			var = var | (integerId - 2);
			var = var & 0b10010010;
		}else if(Ayyye :D){
			while(true){
				var = 3;
				break;
			}
			return true;
		}
		return false;
	}

	real _func(int i, int j){
		return i % j * 3.52;
	}
}
```

Symbol Table

Lexeme	Token
0	var
1	integerId
2	_main
3	boolean
4	var_2
5	_func
6	i
7	j

Output

Lexeme	Token	Attribute
class	.,.	-
Program		-
{		-
int		-
var (Add to symbol table)		0
,		-
integerId (Add to symbol table)		1
=		-
4		4
;		-
static		-
void		-
_main (Add to symbol table)		2
(		-
)		-
{		-
var		0
=		-
-		-
23		23
-		-
22		22
;		-
print		-
(		-
“N San” + “lty.”		“NSanlty.”
)		-
;		-
for		-
(		-
integerId		1
in		-
0		0
to		-
100		100
steps		-
10		10
)		-
{		-
;		-
}		-
}		-
bool		-

boolean (Add to symbol table)		3
(		-
)		-
{		-
real		-
var_2 (Add to symbol table)		4
=		-
0.14		0.14
;		-
if		-
(		-
var		0
<		-
8		8
&&		-
32		32
-		-
var		0
>		-
var_2		4
)		-
{		-
var		0
=		-
var		0
|		-
(		-
integerId		1
-		-
2		2
)		-
;		-
var		0
=		-
var		0
&		-
0b10010010		146
;		-
}		-
else		-
if		-
(		-
)		-
{		-
while		-
(		-

true		-
)		-
{		-
var		0
=		-
3		3
;		-
break		-
;		-
}		-
return		-
true		-
;		-
}		-
return		-
false		-
;		-
}		-
real		-
_func (Add to symbol table)		5
(		-
int		-
I (Add to symbol table)		6
,		-
int		-
j (Add to symbol table)		7
)		-
{		-
return		-
i		6
%		-
j		7
*		-
3.52		3.52
;		-
}		-
}		-