

Alunos: Samuel Eduardo | Luiz Ricardo  
Nome da Linguagem: CC# (Competitive C#)  
Linguagem desenvolvimento: C#

Introdução: A Competitive C# é uma linguagem para estudo de maratonas de programação. Ela irá conter funções nativas de alguns temas como Teoria dos Grafos, Teoria dos Números, etc. Por exemplo, ela contém um tipo de variável chamada grafo, e nativamente contém implementada várias funções como: menor caminho entre dois vértices, árvore geradora mínima, etc.

Extensão dos códigos fontes: .cc#

### Tabela de Tokens:

Token	Lexema	Regex	Descrição
var	a,ab,a4f,a_b	^([A-Z a-z _] ([A-Z0-9a-z])*)\$	nomenclatura das variáveis
se	if	^if\$	palavra reservada if
senao	else	^else\$	palavra reservada else
abre_ch	{	^{	início de bloco de instruções
fecha_ch	}	^}	fim de bloco de instruções
tipo	(int   double   text   grafo, grafop)	^(int)\$ ^(double)\$ ^(text)\$ ^(grafo)\$ ^(grafop)\$	tipo de variável inteira
abre_p	(	^\(\$	abre parênteses
fecha_p	)	^\)\$	fecha parênteses
repet	loop	^loop\$	palavra reservada loop
op_ar	+, -, *, /	^(+)\$ ^(^-)\$ ^(*)\$ ^(/)\$	operações aritméticas
op_rel	>, <, <=, >=, ==, !=	^(>)\$ ^(<)\$ ^(<=)\$ ^(>=)\$ ^(!=)\$ ^(==)\$	operações relacionais

atr	=	^=\$	operador igual
contem	in	^in\$	palavra reservada in
num_inteiro	1,2,3,4434,4332	^([0-9])*\$	números inteiros no código
num_real	1.3, 1.0, 323.43	^([0-9])+\.([0-9])*\$	números reais no código
strings	"alo", "w", "dsds12"	^".*\$	cadeias de caracteres no código
ponto_virgula	;	^;\$	delimitador de final de instrução
abre_com	\	^\\\$	abre bloco de comentários
fech_com	/	^/\$	fecha bloco de comentários
fun	fun	^fun\$	início de uso de função
,	,	^,\$	vírgula

O projeto foi feito com a IDE Visual Studio 2017.

Para execução direta pela IDE, basta abrir o projeto (arquivo Compilador.sln, dentro da pasta Compilador) pelo Visual Studio.

Nos anexos tem o instalador do programa (CompiladorSetup). Após instalado (costuma demorar um pouco pra instalar), vá no seu disco local, Arquivos de Programa, entre na pasta Default Company Name, CompiladorCC#, execute o arquivo Compilador.exe

Como a gente não usou bibliotecas prontas pra fazer a análise léxica, a gente padronizou que tudo na linguagem deve ser separado por espaços.

Eis um exemplo de um código em CC#

```
grafo g ;  
int i ;  
loop ( i = 0 ; i < n ; i = i + 1 )  
{  
    g = fun gera_grafo_aleatorio ( 10 , 20 ) ;  
  
    int resp ;  
  
    resp = fun arvore_geradora_minima ( g ) ;  
  
    fun print ( resp ) ;  
}
```

Basicamente, na tela do programa, na parte superior, temos o botão para limpamos o código, um botão para abrir um arquivo fonte em CC# (extensão .cc#), um botão “Salvar como...”, para salvar o código que foi feito no programa, e um botão para visualização da tabela de Tokens.

Na parte inferior direita, temos um botão que executa a análise léxica, um botão que executa a análise sintática(que só é habilitado depois de que foi feita a análise léxica) e um botão para limpar a caixa de debug.

## Tela do programa

