

ΑΡΧΕΣ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

FLEX-BISON ΣΥΝΤΑΚΤΙΚΟΣ ΑΝΑΛΥΤΗΣ

JSON

ΜΕΛΗ ΟΜΑΔΑΣ ΥΛΟΠΟΙΗΣΗΣ ΤΟΥ ΠΡΟΤΖΕΚΤ:

ΠΑΠΑΧΡΙΣΤΟΦΙΛΟΥ ΣΑΡΑΝΤΗΣ 3^ο έτος (1072600)

ΝΙΚΟΛΑΟΥ ΣΤΑΥΡΟΣ 3^ο έτος (1069660)

ΑΜΙΤΣΗΣ ΛΕΥΤΕΡΗΣ 3^ο έτος (1072464)

ΔΙΑΔΙΚΑΣΙΑ ΥΛΟΠΟΙΗΣΗΣ ΤΟΥ ΠΡΟΤΖΕΚΤ:

Για την υλοποίηση του πρότζεκτ εργαστήκαμε αναθέτοντας στον καθένα διαφορετικό μέρος της εργασίας, κάναμε συνάντηση κάθε εβδομάδα προκειμένου να συζητήσουμε την πρόοδο που έκανε ο καθένας, πως θα γίνει η συνένωση των κομματιών στα οποία δούλεψε ο καθένας και με ποια θα συνεχίσει.

ΠΕΡΙΕΧΟΜΕΝΑ:

- Συντακτικός ορισμός της γραμματικής της γλώσσας σε BNF. (σελ.2)
- Λεκτικός αναλυτής Lexer-Flex. (σελ.5)
- Συντακτικός αναλυτής Parser-Bison. (σελ.7)
- Εκτέλεση του myParser.exe (σελ.12)

BNF

<last-result-and-active> ::= <last> <active>

<last> ::= <gameId> <drawId> <drawTime> <status> <drawBreak> <visualDraw>
<pricePoints> <winningNumbers> <prizeCategories> <wagerStatistics>

<active> ::= <gameId> <drawId> <drawTime> <status> <drawBreak> <visualDraw>
<pricePoints> <prizeCategories> <wagerStatistics>

<gameId> ::= <INTEGER>

<drawId> ::= <INTEGER>

<drawTime> ::= <INTEGER>

<status> ::= <STRING>

<drawBreak> ::= <INTEGER>

<visualDraw> ::= <INTEGER>

<pricePoints> ::= <amount>

<winningNumbers> ::= <list> <bonus>

<prizeCategories> ::= <id> <dividend> <winners> <distributed> <jackpot> <fixed>
<categoryType> <gameType>

if <id=1> then <minimumDistributed> | <id> <dividend> <winners>
<distributed> <jackpot> <fixed> <categoryType> <gameType>

if <id=1> then <minimumDistributed>, <prizeCategories>

<wagerStatistics> ::= <columns> <wagers> <addOn>

<amount> ::= <FLOAT>

<list> ::= <INTEGER> | <INTEGER> , <list>

<bonus> ::= <INTEGER>
 <id> ::= <1|2|3|4|5|6|7|8>
 <dividend> ::= <FLOAT>
 <winners> ::= <INTEGER>
 <distributed> ::= <FLOAT>
 <jackpot> ::= <FLOAT>
 <fixed> ::= <FLOAT>
 <categoryType> ::= 0|1
 <gameType> ::= <STRING>
 < minimumDistributed > ::= <FLOAT>
 <columns> ::= <INTEGER>
 <wagers> ::= <INTEGER>
 <addOn> ::= []
 <content> ::= <gameId> <drawId> <drawTime> <status> <drawBreak>
 <visualDraw> <pricePoints> <prizeCategories> <wagerStatistics>|
 <gameId> <drawId> <drawTime> <status> <drawBreak> <visualDraw>
 <pricePoints> <prizeCategories> <wagerStatistics>,<content>
 <totalPages> ::= <INTEGER>
 <totalElements> ::= <INTEGER>
 <last> ::= <BOOLEAN>
 <numberOfElements> ::= <INTEGER>
 <sort> ::= <direction> <property> <ignoreCase> <nullHandling> <descending>
 <ascending>
 <direction> ::= <STRING>

<property>::= <STRING>

<ignoreCase>::= <BOOLEAN>

<nullHandling>::= <STRING>

<descending>::= <BOOLEAN>

<ascending>::= <BOOLEAN>

<first>::= <BOOLEAN>

<size>::= <INT>

<number>::= <INT>

<INTEGER> ::= <DIGIT> | <NONZERO> <DIGITS>

<DIGITS> ::= <DIGIT> | <DIGIT> <DIGITS>

<DIGIT> ::= 0 | <NONZERO>

<NONZERO> ::= 1|2|3|4|5|6|7|8|9

<WORD> ::= <LETTER> | <WORD> <LETTER>

<LETTER> ::= A|B|C|...|Z|a|b|...|z

<STRING> ::= <LETTER> <STRING> | <DIGIT> <STRING> | <LETTER> | <DIGIT>

<BOOLEAN>::= <"True"|"False">

ΛΕΚΤΙΚΟΣ ΑΝΑΛΥΤΗΣ LEXER-FLEX

```
1 %{
2 #include "y.tab.h"
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <string.h>
6 int line_num=1;
7 %}
```

Σε αυτό το κομμάτι του lexer περιλαμβάνονται οι απαραίτητες βιβλιοθήκες προκειμένου να είναι λειτουργικός καθώς και το y.tab.h από το οποίο θα παρθούν τα tokens.

```
11
12 %option noyywrap
13
14
15 integer      [0-9]+
16 float        {integer}\.{integer}
17 boolean       ("true"|"false")
18 string        \"[^\"]*\"
19 cstring       {string}.{string}
20 ws            [ \t\n\r]
21 idnum         [1-8]
22 category      [0-1]
23
24
```

Ακριβώς από κάτω υπάρχει Το option noyywrap χρησιμοποιείται προκειμένου ο lexer να μην καλέσει το option noyywrap όταν βρεθεί σε end-of-file. Δηλαδή ο lexer σταματάει να διαβάζει όταν τελειώσει το αρχείο το οποίο διαβάζει και δεν ψάχνει επόμενο αρχείο για να διαβάσει. Ύστερα είναι οι ονομασίες των μεταβλητών που θα

χρησιμοποιηθούν καθώς και τα στοιχεία από τα οποία θα αποτελούνται.

Τέλος είναι τα tokens που θα χρησιμοποιηθούν:

```
25 %%
26 "+" {printf("%s", yytext);return plus;}
27 "-" {printf("%s", yytext);return minus;}
28 ":" {printf("%s", yytext);return start;}
29 "," {printf("%s", yytext);return next;}
30 "." {printf("%s", yytext);return stop;}
31 "[" {printf("%s", yytext);return StartBracers;}
32 "]" {printf("%s", yytext);return StopBracers;}
33 "{" {printf("%s", yytext);return StartBrackets;}
34 "}" {printf("%s", yytext);return StopBrackets;}
35 \n      {++line_num;}
36 {ws}    {;}
37 {boolean} {printf("%s", yytext);return BOOLEAN;}
38 {float}   {printf("%s", yytext); yylval.t_float = atof(yytext); return FLOAT;}
39 {integer} {printf("%s", yytext); yylval.t_int = atoi(yytext); return INT;}
40 \"last\" {printf("%s", yytext);return Last; }
41
42 \"gameId\" {printf("%s", yytext);return gameId; }
43 \"drawId\" {printf("%s", yytext);return drawID;}
44 \"drawTime\" {printf("%s", yytext);return DrawTime;}
45 \"status\" {printf("%s", yytext);return Status;}
46 \"drawBreak\" {printf("%s", yytext);return DrawBreak;}
47 \"visualDraw\" {printf("%s", yytext);return VisualDraw;}
48 \"pricePoints\" {printf("%s", yytext);return PricePoints;}
49 \"winningNumbers\" {printf("%s", yytext);return WinningNumbers;}
50 \"prizeCategories\" {printf("%s", yytext);return PrizeCategories;}
51
```

```

51 \wagerStatistics\ {printf("%s", yytext);return WagerStatistics;}
52 \list\ {printf("%s", yytext);return List;}
53 \bonus\ {printf("%s", yytext);return Bonus;}
54 \id\ {printf("%s", yytext);return ID;}
55 \divident\ {printf("%s", yytext);return Divident;}
56 \winners\ {printf("%s", yytext);return Winners;}
57 \distributed\ {printf("%s", yytext);return Distributed;}
58 \jackpot\ {printf("%s", yytext);return Jackpot;}
59 \fixed\ {printf("%s", yytext);return Fixed;}
60 \categoryType\ {printf("%s", yytext);return CategoryType;}
61 \gameType\ {printf("%s\n", yytext);return GameType;}
62 \minimumDistributed\ {printf("%s", yytext);return MinimumDistributed;}
63 \amount\ {printf("%s", yytext);return Amount;}
64 \columns\ {printf("%s", yytext);return Columns; }
65 \wagers\ {printf("%s", yytext);return Wager; }
66 \addOn\ {printf("%s", yytext);return AddOn; }
67 \content\ {printf("%s", yytext);return Content;}
68 \totalPages\ {printf("%s",yytext);return TotalPages;}
69 \totalElements\ {printf("%s",yytext);return TotalElements;}
70 \numberOfElements\ {printf("%s",yytext);return NumberOfElements;}
71 \sort\ {printf("%s",yytext);return Sort;}
72 \direction\ {printf("%s",yytext);return Direction;}
73 \property\ {printf("%s",yytext);return Property;}
74 \ignoreCase\ {printf("%s",yytext);return IgnoreCase;}
75 \nullHandling\ {printf("%s",yytext);return NullHandling;}
76 \descending\ {printf("%s",yytext);return Descending;}
77 \ascending\ {printf("%s",yytext);return Ascending;}
78 \first\ {printf("%s",yytext);return First;}
79 \size\ {printf("%s",yytext);return Size;}
80 \number\ {printf("%s",yytext);return Number;}
81 {string} {printf("%s", yytext); yylval.t_str = strdup(yytext); return STRING; }
82

```

ΣΥΝΤΑΚΤΙΚΟΣ ΑΝΑΛΥΤΗΣ PARSER-BISON

```
1 %{
2 #include <stdio.h>
3 #include <math.h>
4 #include <stdlib.h>
5
6 void yyerror(char *);
7 extern FILE *yyin;
8 extern FILE *yyout;
9 extern int line_num;
10 extern int yylex();
11 extern int yyval;
12 extern char* yytext;
13 extern char* ActiveCheck;
14 extern int yyparse();
15 void CategoryRule(int a);
16 void yyerror(char *s);
17 void idRULE(int a);
18 void GameIDRule(int g);
19 void Act(char *d);
20 void JsonItemsNum(int number);
21 void ListRule(int list);
22 void ListCountRule(int count);
23 int idnum;
24 int category;
25 int jsonitems=0;
26 int countlist=0;
27 int listnum;
28 %}
```

Ομοίως με τον lexer το πρώτο κομμάτι του parser περιέχει τις βιβλιοθήκες καθώς και τις δηλώσεις μεταβλητών και συναρτήσεων που χρησιμοποιούνται παρακάτω .

```
31 %union
32 {
33     char *t_str;
34     int t_int;
35     float t_float;
36 }
37
38
39
40
41
42
43 %token plus
44 %token minus
45 %token start
46 %token next
47 %token stop
48 %token StartBracers
49 %token StopBracers
50 %token StartBrackets
51 %token StopBrackets
52 %token BOOLEAN
53 %token<t_str> STRING
54 %token<t_int> INT
55 %token<t_float> FLOAT
56 %token gameId
57 %token drawID
58 %token DrawTime
59 %token Status
60 %token DrawBreak
61 %token VisualDraw
62 %token PricePoints
63 %token WinningNumbers
64 %token PrizeCategories
65 %token WagerStatistics
```

Σε αυτό το κομμάτι περιέχει το union για τις μεταβλητές που θα χρησιμοποιηθούν για την ανάθεση τιμών στις παραμέτρους των

Συναρτήσεων καθώς και τα tokens που θα χρησιμοποιηθούν.

```
-----
66 %token List
67 %token Bonus
68 %token ID
69 %token Divident
70 %token Winners
71 %token Distributed
72 %token Jackpot
73 %token Fixed
74 %token CategoryType
75 %token GameType
76 %token MinimumDistributed
77 %token Last
78 %token Columns
79 %token Wager
80 %token AddOn
81 %token Amount
82 %token Content
83 %token TotalPages
84 %token TotalElements
85 %token NumberOfElements
86 %token Sort
87 %token Direction
88 %token Property
89 %token IgnoreCase
90 %token NullHandling
91 %token Descending
92 %token Ascending
93 %token First
94 %token Size
95 %token Number
96
97 %%
```

Ύστερα είναι το κομμάτι του parser όπου περιέχει την γραμματική που απαιτείται για να διαβαστούν τα αρχεία:

```
1 1000
98 jsonFile: StartBrackets ChooseR StopBrackets;
99
100 ChooseR: LastAndActiveR| ContentetcR ;
101
102 ContentetcR: Content start ContentArrayR next TotalPages start INT next TotalElements start INT next
103              Last start BOOLEAN next NumberOfElements start INT next Sort start SortArrayR next
104              First start BOOLEAN next Size start INT next Number start INT ;
105
106 ContentArrayR: StartBracers LastR ContentArrayR
107                |next LastR ContentArrayR
108                |StopBracers;
109
110
111 LastAndActiveR: Last start LastR next STRING start ActiveR ;
112
113 SortArrayR: StartBracers SortR SortArrayR
114              |next SortR SortArrayR
115              |StopBracers;
116
117 SortR: StartBrackets Direction start STRING next Property start STRING next IgnoreCase
118        start BOOLEAN next NullHandling start STRING next Descending start BOOLEAN
119        next Ascending start BOOLEAN StopBrackets;
```



```

121
122 LastR:      StartBrackets gameID start INT{$<t_int>$=$<t_int>4;idnum=yyval.t_int;GameIDRule(idnum);}
123             next drawID start INT next DrawTime start INT next
124             Status start STRING next DrawBreak start INT next VisualDraw start INT
125             next PricePoints start PPR next WinningNumbers start WNR next
126             PrizeCategories start JSONARRAYR {JsonItemsNum(jsonitems);jsonitems=0;}
127             next WagerStatistics start WSR StopBrackets ;
128
129
130 ActiveR: StartBrackets gameID start INT{$<t_int>$=$<t_int>4;idnum=yyval.t_int;GameIDRule(idnum);}
131             next drawID start INT next DrawTime start INT next
132             Status start STRING next DrawBreak start INT next VisualDraw start INT
133             next PricePoints start PPR next PrizeCategories start JSONARRAYR {JsonItemsNum(jsonitems);jsonitems=0;} next
134             WagerStatistics start WSR StopBrackets;
135
136 PPR:      StartBrackets Amount start FLOAT StopBrackets;
137
138 WNR:      StartBrackets List start ListR{ListCountRule(countlist);countlist=0;} next Bonus start bonusR StopBrackets;
139
140 ListR:    StartBracers INT{$<t_int>$=$2;listnum=yyval.t_int;ListRule(listnum);countlist++;} ListR
141             | next INT{$<t_int>$=$2;listnum=yyval.t_int;ListRule(listnum);countlist++;} ListR
142             | StopBracers;

144 bonusR: StartBracers INT StopBracers;
145
146
147 WSR:      StartBrackets Columns start INT next Wager start INT next AddOn start JARRAYR
148             StopBrackets
149             ;
150
151 JARRAYR: StartBracers TYPER JARRAYR
152             |next TYPER JARRAYR
153             |StopBracers;
154
155
156 TYPER: INT|FLOAT|STRING| ;
157
158 JSONARRAYR: StartBracers JSONITEMR{jsonitems++;} JSONARRAYR
159             |next JSONITEMR{jsonitems++;} JSONARRAYR
160             |StopBracers;
161
162
163 JSONITEMR: StartBrackets ID start INT {$<t_int>$=$4;idnum=yylval.t_int;} next Divident start FLOAT
164             next Winners start INT next Distributed start FLOAT next Jackpot start FLOAT next Fixed start FLOAT next CategoryType start
165             INT {$<t_int>$=$29;category=yylval.t_int; CategoryRule(category);} next GameType start STRING
166             MinimumDisR;
167
168 MinimumDisR: next MinimumDistributed {idRULE(idnum);} start FLOAT StopBrackets
169             | StopBrackets
170             ;
171
172

```

Τέλος θα γίνει αναλυτικός σχολιασμός των συναρτήσεων που χρησιμοποιήθηκαν για να παραμετροποιήσουμε των parser.

```

173
174 void ListCountRule(int count)
175 {
176     if(count!=5)
177     {
178         printf("\nList needs exactly 5 integers! \n");
179         exit(EXIT_FAILURE);
180     }
181 }

```

Η συνάρτηση αυτή φροντίζει ώστε μέσα στην λίστα του token List υπάρχουν ακριβώς 5 αντικείμενα και παρακάτω βλέπουμε πως χρησιμοποιήθηκε η

συγκεκριμένη συνάρτηση. Ουσιαστικά κάθε φορά που διάβασε INT ο parser πρόσθετε μονάδα στην μεταβλητή countlist η οποία μπαίνει σαν παράμετρος μέσα στην συνάρτηση και ύστερα ξαναγίνεται 0 έτσι ώστε στην επόμενη λίστα να μην ξεκινήσει από το 5.

```

138 WNR:   StartBrackets List start ListR{ListCountRule(countlist);countlist=0;} next Bonus start bonusR StopBrackets;
139
140 ListR:   StartBracers INT{<t_int>=$2;listnum=yyval.t_int;ListRule(listnum);countlist++;} ListR
141         | next INT{<t_int>=$2;listnum=yyval.t_int;ListRule(listnum);countlist++;} ListR
142         | StopBracers;

```

Η παρακάτω συνάρτηση είναι μία πολύ απλή συνάρτηση που ελέγχει αν τα νούμερα της λίστας είναι μεταξύ του 1 και του 45.

```

183 void ListRule(int list)
184 {
185     if(list<1 || list>45)
186     {
187         printf("\nYou need integers between 1 and 45 in this section! \n");
188         exit(EXIT_FAILURE);
189     }
190 }
191

```

```

191
192 void JsonItemsNum(int number)
193 {
194     if(number!=8)
195     {
196         printf("\nYou need exactly 8 json items! \n");
197         exit(EXIT_FAILURE);
198     }

```

Η συνάρτηση αυτή είναι ακριβώς ίδια με την ListCount μόνο που αυτή την φορά μετράει τα εμφωλευμένα JsonItems στο array.

Η παρακάτω συνάρτηση είναι αυτή που ζητείται στο τρίτο ερώτημα η οποία φροντίζει να ελεγχθεί το gameId και αν δεν είναι ένα από αυτά που δίνονται στην εκφώνηση να τερματίσει το parsing.

```
201 void GameIDRule(int g)
202 {
203     if((g==1100) || (g==1110) || (g==2100) || (g==2101) || (g==5103) || (g==5104) || (g==5106) ) ;
204     else
205     {
206         printf("\nWrong GameID: %d \n",g);
207         exit(EXIT_FAILURE);
208     }
209 }
```

```
211 void CategoryRule(int a)
212 {
213     if(a>1 || a<0)
214     {
215         printf("\nWrong Category Type\n");
216         exit(EXIT_FAILURE);}
217     }
218
219
220
```

Άλλη μια απλή function που ελέγχει αν το category είναι είτε 1 είτε 0.

```
221 void idRULE(int a)
222 {
223     if(a>8 || a<1)
224     {
225         printf("\nWrong id number \n");
226         exit(EXIT_FAILURE);
227     }
228
229     else if (a!=1) {
230
231         printf("\nID is not 1\n");
232         exit(EXIT_FAILURE);
233     }
234
235 }
236 }
237
```

Αυτή η εντολή έχει δύο χρήσεις η μία είναι να ελέγχει αν το id του jsonItem είναι μεταξύ του 1 και του 8. Η δεύτερη είναι να ελέγχει το minimumdistributed να υπάρχει μόνο στην περίπτωση που το id=1 όπως φαίνεται παρακάτω.

```
163 JSONITEMR: StartBrackets ID start INT {$<t_int>$=$4;idnum=yylval.t_int;} next Divident start FLOAT
164 next Winners start INT next Distributed start FLOAT next Jackpot start FLOAT next Fixed start FLOAT next CategoryType start
165 INT {$<t_int>$=$29;category=yylval.t_int; CategoryRule(category);} next GameType start STRING
166 MinimumDisR;
167
168 MinimumDisR: next MinimumDistributed {idRULE(idnum);} start FLOAT StopBrackets
169 | StopBrackets
170 ;
```

ΕΚΤΕΛΕΣΗ ΤΟΥ myParser.exe

Η εκτέλεση του προγράμματος μέσω του λειτουργικού του Ubuntu γίνεται με μια σειρά πολύ απλών εντολών :

```
flex lexer.l
```

```
bison -y -d parser.y
```

```
gcc -c y.tab.c lex.yy.c
```

```
gcc y.tab.o lex.yy.o -o myParser.exe
```

```
./myParser.exe file
```