

Εργαστηριακή Άσκηση Flex/Bison
Ιούνιος 2023

	Όνοματεπώνυμο	ΑΜ	Έτος	e-mail
1	ΛΕΦΑΚΗΣ ΑΛΕΞΑΝΔΡΟΣ	1064880	6	up1064880@upnet.gr
	ΑΥΓΕΡΗΣ ΝΙΚΟΛΑΟΣ	1067508	5	up1067508@upnet.gr
	ΣΠΥΡΟΣ ΓΑΡΟΥΦΑΛΗΣ	1067460	5	up1067460@upnet.gr
	ΙΩΑΝΝΗΣ ΖΑΧΟΣ	1053363	5	up1053363@upnet.gr

ΑΣΚΗΣΗ 1

A)BNF

<XML> ::= <Layout> | <RelativeLayout> | <comment>

<comment> ::= "<!--" <String>"-->"

<CloseTag> ::= >

<CloseTag2> ::= />

<Layout> ::= <<LinearLayout >

<Necessary Attribute>

[android:id = " <String> "]

[android:orientation= " <String> "]

<CloseTag>

<Content>

< </LinearLayout> >

<RelativeLayout> ::= <<RelativeLayout>>

<Necessary Attribute>

[android:id = " <String> "]

<CloseTag>

<Content> "

< </RelativeLayout>>

<Attributes> ::= <Attribute> | <Attributes> <Attribute>

<Necessary Attribute> ::= android:layout_height = " (<String> | <Integer>)"
android:layout_width=" (<String> | <Integer>)"

<radio button> ::= < <RadioButton>
 <Necessary Attribute>

<Content> ::= <Element> | <Content> <Element>

<Element> ::= <Layout> | <RelativeLayout>

| < <TextView> <Necessary Attribute> <android:text=" <String> " >

 [android:id = " <String> "] [android:textColor= " <String> "] <CloseTag2>

| < <ImageView > <Necessary Attribute> <android:src=" " <String> " > [android:id=" <String> "]

 [android:textColor=" <String> "] <CloseTag2>

| < <Button> <Necessary Attribute> <android:text=" <String> ">

 [android:id= " <String> "] [android:padding=" <Integer>"] <CloseTag2>

| <radio button>

| < <RadioGroup > <Necessary Attribute> <radio button>

 [android:id=" <String> "]

 [android:checkedButton=" <String>"] </RadioGroup>

| < <ProgressBar> <Necessary Attribute> [android:max=" <Integer>"]

 [android:id=" <String> "]

 [android:progress = " <Integer>"] <CloseTag2>

<String> ::= ([a-zA-Z0-9| " "])+

<Integer> ::= [1-9] [0-9]*

B)ΠΕΡΙΓΡΑΦΗ ΚΩΔΙΚΑ

flex

1. `whitespace [\t\r\n]*`: Αντιστοιχεί σε οποιονδήποτε συνδυασμό κενών χαρακτήρων, όπως κενά, `tab`, carriage return και `newline`. Αυτή η έκφραση χρησιμοποιείται για να αγνοήσει τα κενά μεταξύ των λέξεων ή των γραμμών.
2. `string [a-zA-Z._0-9]+[\t\r\n]*[a-zA-Z._0-9]+`: Αντιστοιχεί σε μια αλφαριθμητική ακολουθία που περιέχει γράμματα, ψηφία, τελείες και κάτω παύλες. Η ακολουθία μπορεί να περιέχει κενά στην αρχή ή το τέλος της.
3. `num [1-9][0-9]*`: Αντιστοιχεί σε έναν μη μηδενικό αριθμό. Πρέπει να ξεκινά με έναν αριθμό από 1 έως 9 και μπορεί να ακολουθούν οποιαδήποτε αριθμητική ακολουθία.
4. `character [a-zA-Z]`: Αντιστοιχεί σε ένα μόνο γράμμα από το `a` έως το `z` (πεζό ή κεφαλαίο).
5. `word {character}+`: Αντιστοιχεί σε μια ακολουθία γραμμάτων που αποτελούν ένα λεκτικό. Το λεκτικό πρέπει να αποτελείται από ένα ή περισσότερα γράμματα.
6. `alphanumeric ({word}*{num}*)*`: Αντιστοιχεί σε μια ακολουθία που περιέχει λέξεις και αριθμούς, με τις λέξεις να μπορούν να ακολουθούνται από αριθμούς. Αυτό επιτρέπει τη συνδυασμένη χρήση λέξεων και αριθμών.
7. `text ({alphanumeric}|\)*`: Αντιστοιχεί σε μια ακολουθία χαρακτήρων που περιέχει λέξεις, αριθμούς, τελείες, κάτω παύλες και κενά. Οποιοσδήποτε χαρακτήρας ενδιάμεσα σε αλφαριθμητικά περιβάλλεται με εισαγωγικά. Αυτό επιτρέπει την περιγραφή κειμένου που περιέχει λέξεις και αριθμούς, αλλά και κενά μεταξύ των λέξεων.
8. `comments "<!-- \"{text}\" -->"`: Αντιστοιχεί σε ένα σχόλιο που ξεκινά με `<!--`, ακολουθείται από κείμενο (`text`) και τελειώνει με `-->`. Αυτή η έκφραση χρησιμοποιείται για να αναγνωρίσει και να αγνοήσει τα σχόλια στον.

bison

Η παραγωγή `android_layout: linear_layout | relative_layout`; καθορίζει ότι ένα `android_layout` μπορεί να είναι είτε ένα `linear_layout` είτε ένα `relative_layout`. Ανάλογα με την επιλογή, θα ακολουθήσει η αντίστοιχη παραγωγή για το επιλεγμένο `layout`.

Ας εξηγήσουμε ακόμα περισσότερο την παραγωγή `linear_layout`:

Η παραγωγή `linear_layout: LINEAR_LAYOUT linear android_elements END_LINEAR_LAYOUT TK_EOF`; καθορίζει τη δομή ενός (`LinearLayout`) στη γλώσσα που περιγράφετε. Η παραγωγή αποτελείται από τα εξής στοιχεία:

1. `LINEAR_LAYOUT`: Αναγνωρίζει την αρχή

2. `linear`: Αναφέρεται σε έναν κανόνα με το όνομα `linear` που περιγράφει τα χαρακτηριστικά.
3. `android_elements`: Αναφέρεται στα στοιχεία
 - `text_view`
 - | `image_view`
 - | `button`
 - | `radio_group`
 - | `progress_bar`
 - | `linear_layout`
 - | `relative_layout`
4. `END_LINEAR_LAYOUT`: Αναγνωρίζει το τέλος.
5. `TK_EOF`: Αναγνωρίζει το τέλος του αρχείου.

1. Η παραγωγή `linear: ll params CLOSE_TAG | params ll CLOSE_TAG | params CLOSE_TAG` αναφέρεται στη δημιουργία ενός γραμμικού διάταξης (`LinearLayout`). Αυτός ο κανόνας συνδυάζει τον κανόνα `ll` με το `params` και το `CLOSE_TAG`, ή το `params` με τον κανόνα `ll` και το `CLOSE_TAG`. Ο κανόνας `ll` χρησιμοποιείται για να προσθέσει επιπλέον στοιχεία, ενώ ο κανόνας `params` ορίζει τις παραμέτρους, όπως το πλάτος και το ύψος.
2. Η παραγωγή `params: ANDROID_LAYOUT_WIDTH TK_EQUAL TK_E TK_STRING TK_E ANDROID_LAYOUT_HEIGHT TK_EQUAL TK_E TK_STRING TK_E` είναι υποχρεωτική για κάθε στοιχείο.
3. Η παραγωγή `ll: ll ll | ll` αναφέρεται στη σειρά που πρέπει να εμφανίζονται οι παράμετροι και τα στοιχεία. Ο κανόνας `ll` μπορεί να είναι αναδρομικός, προσθέτοντας συνεχώς νέες παραμέτρους ή στοιχεία στο γραμμικό διάγραμμα. Ο κανόνας `ll` περιγράφει τα στοιχεία που πρέπει να εμφανίζονται τα οποία δεν είναι υποχρεωτικά, όπως η προσανατολισμός (`ANDROID_ORIENTATION`) ή το αναγνωριστικό (`ANDROID_ID`) με την αντίστοιχη τιμή.

Αντίστοιχα γίνονται και τα υπόλοιπα αλλά δεν προλάβαμε λόγο χρόνου

ΚΩΔΙΚΑΣ

FLEX FILE

```
%{
#include<stdio.h>
#include<stdlib.h>
```

```

#include"parser.tab.h"

void printToken(int tokenID);

int correct=0;


%}

%option noyywrap

%option yylineno


whitespace [ \t\r\n]*


string [a-zA-Z._0-9]+[ \t\r\n]*[a-zA-Z._0-9]+
num [1-9][0-9]*
character [a-zA-Z]
word {character}+
alphanumeric ({word}*{num})*


text ({alphanumeric}|"\"")*
comments "<!-- \"{text}\" -->"


%%

{comments} printf("%s\n",yytext);

\" {ECHO; return TK_E;}

"=" {ECHO; return TK_EQUAL;}

"<LinearLayout" {ECHO; return LINEAR_LAYOUT; }

"</LinearLayout>" {ECHO; return END_LINEAR_LAYOUT; }

"<RelativeLayout" {ECHO; return RELATIVE_LAYOUT; }

"</RelativeLayout>" {ECHO; return END_RELATIVE_LAYOUT; }

"<TextView" {ECHO; return TEXT_VIEW; }

">" {ECHO; return CLOSE_TAG; }

"/>" {ECHO; return CLOSE_TAG2; }

"<ImageView" {ECHO; return IMAGE_VIEW; }

"<Button" {ECHO; return BUTTON; }

```

```

"<RadioGroup"      {ECHO; return RADIO_GROUP; }
"</RadioGroup>"    {ECHO; return END_RADIO_GROUP; }
"<RadioButton"     {ECHO; return RADIO_BUTTON; }
"<ProgressBar"    {ECHO; return PROGRESS_BAR; }
"android:id"       {ECHO; return ANDROID_ID; }
"android:orientation" {ECHO; return ANDROID_ORIENTATION; }
"android:text"      {ECHO; return ANDROID_TEXT; }
"android:textColor" {ECHO; return ANDROID_TEXT_COLOR; }
"android:src"       {ECHO; return ANDROID_SRC; }
"android:padding"   {ECHO; return ANDROID_PADDING; }
"android:max"       {ECHO; return ANDROID_MAX; }
"android:progress"  {ECHO; return ANDROID_PROGRESS; }
"android:checkedButton" {ECHO; return ANDROID_CHECKED_BUTTON; }
"android:layout_width" {ECHO; return ANDROID_LAYOUT_WIDTH; }
"android:layout_height" {ECHO; return ANDROID_LAYOUT_HEIGHT; }
"radio_button_count" {ECHO; return RADIO_BUTTON_COUNT;}

```

```

{string} {ECHO; return TK_STRING;}

```

```

{num} {ECHO; return TK_NUM;}

```

```

{whitespace} { /* Return nothing */ ECHO;}

```

```

. {printf("Unrecognized character at line %d!\n",yylineno);}

```

```

%%

```

```

void printToken(int tokenID){
printf("Found token '%s' (%d) at line %d\n", yytext, tokenID,
yylineno);
}

```

BISON FILE

```

%{

```

```

#include <stdio.h>

```

```

#include <stdlib.h>

```

```

#include <string.h>

```

```

#include <stdbool.h>

```

```
#define MAX_IDS 100
```

```
char* id_values[MAX_IDS];
```

```
int id_count = 0;
```

```
extern int yylineno;
```

```
extern int yylex();
```

```
extern FILE* yyin;
```

```
void yyerror(const char* s);
```

```
extern char* yytext;
```

```
int maxAttributeValue = 0;
```

```
int expected_count=0;
```

```
int radio_button_count = 0;
```

```
%}
```

```
%union {
```

```
    char* string;
```

```
    int integer;
```

```
    double decimal;
```

```
}
```

```
%define parse.error verbose
```

```
%token LINEAR_LAYOUT "<LinearLayout"
```

```
%token END_LINEAR_LAYOUT "</LinearLayout>"
```

```
%token RELATIVE_LAYOUT "<RelativeLayout"
```

```
%token END_RELATIVE_LAYOUT "</RelativeLayout>"
```

```
%token TEXT_VIEW "<TextView"
```

```
%token CLOSE_TAG ">"
```

```
%token CLOSE_TAG2 ">"
```

```
%token IMAGE_VIEW "<ImageView"
```

```
%token BUTTON "<Button"
```

%token RADIO_GROUP "<RadioGroup"
%token END_RADIO_GROUP "</RadioGroup>"
%token RADIO_BUTTON "<RadioButton"
%token PROGRESS_BAR "<ProgressBar"
%token ANDROID_ID "android:id"
%token ANDROID_ORIENTATION "android:orientation"
%token ANDROID_TEXT "android:text"
%token ANDROID_TEXT_COLOR "android:textColor"
%token ANDROID_SRC "android:src"
%token ANDROID_PADDING "android:padding"
%token ANDROID_MAX "android:max"
%token ANDROID_PROGRESS "android:progress"
%token ANDROID_CHECKED_BUTTON "android:checkedButton"
%token ANDROID_LAYOUT_WIDTH "android:layout_width"
%token ANDROID_LAYOUT_HEIGHT "android:layout_height"
%token TK_EQUAL "="
%token TK_E
%token RADIO_BUTTON_COUNT "radio_button_count"
%token <integer> TK_NUM "num"

%token <string> TK_STRING "String"

%{

%}

%%

android_layout: linear_layout
 | relative_layout
 ;

linear_layout: LINEAR_LAYOUT


```
linear
    android_elements
END_LINEAR_LAYOUT
TK_EOF
;
```

```
linear: ll params CLOSE_TAG
    | params ll CLOSE_TAG
    | params CLOSE_TAG
;
```

```
params: ANDROID_LAYOUT_WIDTH TK_EQUAL TK_E TK_STRING TK_E ANDROID_LAYOUT_HEIGHT
TK_EQUAL TK_E TK_STRING TK_E

    | ANDROID_LAYOUT_HEIGHT al ANDROID_LAYOUT_WIDTH al
;
```

```
ll: ll l1 | l1
;
```

```
l1: ANDROID_ORIENTATION al
    | ANDROID_ID TK_EQUAL TK_E TK_STRING TK_E

;
```

```
relative_layout: RELATIVE_LAYOUT
    relative
    android_elements
END_RELATIVE_LAYOUT
;
```

```
relative: rl params CLOSE_TAG
    | params CLOSE_TAG
    | params rl CLOSE_TAG
```

;

rl: ANDROID_ID al

;

android_elements: android_element
| android_elements android_element
;

android_element: text_view
| image_view
| button
| radio_group
| progress_bar
| linear_layout
| relative_layout
;

text_view: TEXT_VIEW
tv
CLOSE_TAG2
;

```
tv: params text ANDROID_TEXT al
    | text ANDROID_TEXT al params text
    | ANDROID_TEXT al params text
    | params ANDROID_TEXT al
;
```

```
text: text textv
    | textv
;
```

```
textv: ANDROID_ID TK_EQUAL TK_E TK_STRING TK_E
    | ANDROID_TEXT_COLOR al
;
```

```
image_view: IMAGE_VIEW
    iv3
    CLOSE_TAG2
;
```

```
iv3: params iv2 ANDROID_SRC al
    | ANDROID_SRC al iv2 params
    | ANDROID_SRC al params iv2
    | params ANDROID_SRC al iv2
    | params ANDROID_SRC
    | ANDROID_SRC params
;
```

```
iv2: iv2 iv
    | iv
;
```

iv: ANDROID_ID aI

| ANDROID_PADDING aI

;

button: BUTTON ANDROID_LAYOUT_WIDTH aI

ANDROID_LAYOUT_HEIGHT aI b3 CLOSE_TAG2

| BUTTON ANDROID_LAYOUT_WIDTH aI

ANDROID_LAYOUT_HEIGHT aI CLOSE_TAG2

;

b3: b2 ANDROID_TEXT aI b2

| ANDROID_TEXT aI b2

;

b2: b2 b

| b

;

b: ANDROID_ID aI

| ANDROID_PADDING TK_EQUAL TK_E TK_STRING TK_E

;

radio_group: RADIO_GROUP

rg3

END_RADIO_GROUP

;

```
rg3: params RADIO_BUTTON_COUNT TK_EQUAL TK_E TK_NUM TK_E rg1 radio_buttons
| rg1 params RADIO_BUTTON_COUNT TK_EQUAL TK_E TK_NUM TK_E rg radio_buttons
| params RADIO_BUTTON_COUNT TK_EQUAL TK_E TK_NUM TK_E radio_buttons
;
```

```
rg1: rg1 rg
| rg
;
```

```
rg: ANDROID_ID al
| ANDROID_CHECKED_BUTTON al

| CLOSE_TAG
;
```

```
radio_buttons: radio_button
| radio_buttons radio_button

;
```

```
radio_button: RADIO_BUTTON rb3 { radio_button_count++; } CLOSE_TAG2
;
```

```
rb3: params rb1 ANDROID_TEXT al
| rb1 params ANDROID_TEXT al
| ANDROID_TEXT al params rb1
| rb1 ANDROID_TEXT al params
| params ANDROID_TEXT al
| ANDROID_TEXT al params

;
```

rb1: rb1 rb

| rb

;

rb:ANDROID_ID TK_EQUAL TK_E TK_STRING TK_E

;

progress_bar: PROGRESS_BAR progress2 CLOSE_TAG2

;

progress2: params progress

| progress params

| params

;

progress: progress p1

| p1

;

p1: ANDROID_ID al

| ANDROID_MAX TK_EQUAL TK_E TK_NUM TK_E

| ANDROID_PROGRESS TK_EQUAL TK_E TK_NUM TK_E

;

al : TK_EQUAL TK_E al1 TK_E

a1 : TK_STRING | TK_NUM

%%

```
int main(int argc, char* argv[]) {
    int token;

    if (argc > 1) {
        yyin = fopen(argv[1], "r");
        if (yyin == NULL) {
            perror("Error opening file");
            return -1;
        }
    }

    if (yyparse() == 0) {
        printf("\n\nThe compile was successful!\n");
        fclose(yyin);
        return 0;
    } else {
        fclose(yyin);
        return -1;
    }
}

void yyerror(const char* s) {
    fprintf(stderr, "error: %s on line %d\n", s, yylineno);
    fprintf(stderr, "Parsing failed\n");
}
```

ΑΣΚΗΣΗ 2

a)

Προσθέτουμε τις απαραίτητες κεφαλίδες στην κορυφή του αρχείου Bison:

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX_IDS 100

char* id_values[MAX_IDS];

int id_count = 0;
```

Τροποποιήουμε τον κανόνα παραγωγής για κάθε χαρακτηριστικό ANDROID_ID:

```
ANDROID_ID TK_EQUAL TK_E TK_STRING TK_E {
    // Έλεγχος για μοναδικότητα της τιμής android:id
    char* id_value = $4;
    for (int i = 0; i < id_count; i++) {
        if (strcmp(id_value, id_values[i]) == 0) {
            yyerror("Duplicate android:id value");
            return 1;
        }
    }
    // Αποθήκευση της τιμής android:id στον πίνακα id_values
    id_values[id_count++] = id_value;
}
```

b)

```
params: ANDROID_LAYOUT_WIDTH TK_EQUAL TK_E TK_STRING TK_E
ANDROID_LAYOUT_HEIGHT TK_EQUAL TK_E TK_STRING TK_E {
    if (strcmp($4, "wrap_content") != 0 && strcmp($4, "match_parent") != 0 && atoi($4)
    <= 0) {
        fprintf(stderr, "Invalid value for android:layout_width on line %d\n", yylineno);
        YYERROR;
    }

    if (strcmp($8, "wrap_content") != 0 && strcmp($8, "match_parent") != 0 &&
    atoi($8) <= 0) {
```



```

        fprintf(stderr, "Invalid value for android:layout_height on line %d\n", yylineno);
        YYERROR;

    }
} ;

```

c)

```

b: ANDROID_ID al
| ANDROID_PADDING TK_EQUAL TK_E TK_DIGIT TK_E {
    // Check if the value is a positive integer
    int intValue = $4; // Store the integer value directly
    if (intValue <= 0) {
        fprintf(stderr, "error: Invalid value '%d' for attribute 'android:padding' on line %d\n",
            intValue, yylineno);
    }
}
;

```

e)

Προσθέτουμε μια καθολική μεταβλητή για να αποθηκεύσουμε τη μέγιστη τιμή που έχει οριστεί στο χαρακτηριστικό android:max:

```
int maxAttributeValue = 0;
```

b: ANDROID_ID a|

```
| ANDROID_MAX TK_EQUAL TK_E TK_DIGIT TK_E{  
    maxAttributeValue = $4;  
}
```

```
| ANDROID_PROGRESS TK_EQUAL TK_E TK_DIGIT TK_E{  
    int progressValue = $4;  
    if (progressValue < 0 || progressValue > maxAttributeValue) {  
        fprintf(stderr, "error: Invalid value '%s' for attribute 'android:progress' on  
line %d\n",  
            $4, yylineno);  
    }  
}  
;  
;
```

ΑΣΚΗΣΗ 3

Ορίζουμε ένα νέο token στην ενότητα lexer του κώδικα:

```
%token RADIO_BUTTON_COUNT "radio_button_count"
```

Προσθέτουμε το νέο διακριτικό στη γραμματική ενότητα του κώδικα, ως υποχρεωτικό χαρακτηριστικό του στοιχείου <RadioGroup>:

```
rg3: params RADIO_BUTTON_COUNT TK_EQUAL TK_E TK_NUM TK_E { int expected_count  
= $5; } rg1 radio_buttons  
| rg1 params rg1 RADIO_BUTTON_COUNT TK_EQUAL TK_E TK_DIGIT TK_E radio_buttons  
;
```

Προσθέτουμε μια μεταβλητή για να παρακολουθούμε τον αριθμό των στοιχείων `<RadioButton/>` που συναντήθηκαν:

```
int radio_button_count = 0;
```

Τροποποιούμε τον κανόνα `radio_button` για να αυξήσουμε τη μεταβλητή `radio_button_count`:

```
radio_button: RADIO_BUTTON rb3
                { radio_button_count++; }
                CLOSE_TAG2 ;
```

Τροποποιήσουμε τον κανόνα `radio_buttons` για να ελέγξουμε τον αριθμό των στοιχείων `<RadioButton/>`:

```
radio_buttons: radio_button
                | radio_buttons radio_button
                {
                    if (radio_button_count != expected_count) {
                        printf("Error: The count of radio buttons (%d) differs from the expected count (%d).\n",
                               radio_button_count, expected_count);
                    }
                }
            }
```

ΑΣΚΗΣΗ 1 ΑΠΟΤΕΛΕΣΜΑΤΑ

```
flex EXA.l
```

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

```
gcc parser.tab.c lex.yy.c -o parser
```

```
./parser.exe input.txt
```

```

[04] ~/tmp
$ ./parser.exe input.txt
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical">

  <TextView
    android:layout_width="20"
    android:layout_height="20"
    android:id="TV1"
    android:text="Dummy text"/>

  <RadioGroup
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <RadioButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="RB1"
      android:text="Option 1"/>

    <RadioButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="RB2"
      android:text="Option 2"/>

  </RadioGroup>

  <!-- this is a comment -->

  <RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:src="image_name"
      android:padding="50" />

  </RelativeLayout>

</LinearLayout>

The compile was successful!
user@DESKTOP-JVDGAFH ~/tmp
$ _

```

Από την τρίτη γραμμή από το τέλος βγάλαμε το σύμβολο >

The compile was successful!

user@DESKTOP-JVDGAFH ~/tmp

\$./parser.exe input.txt

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical">

  <TextView
    android:layout_width="20"
    android:layout_height="20"
    android:id="TV1"
    android:text="Dummy text"/>

  <RadioGroup
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <RadioButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="RB1"
      android:text="Option 1"/>

    <RadioButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="RB2"
      android:text="Option 2"/>

  </RadioGroup>

  <!-- this is a comment -->

  <RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:src="image_name"
      android:padding="50" Unrecognized character at line 41!
```

error: syntax error, unexpected </RelativeLayout> on line 43

```
</RelativeLayout>
```

user@DESKTOP-JVDGAFH ~/tmp

\$ _

Αφαιρούμε το android:layout_width="20" από το TextView

```
~/tmp

<ImageView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:src="image_name"
  android:padding="50"/>

</RelativeLayout>

</LinearLayout>

The compile was successful!

user@DESKTOP-JVDGAFH ~/tmp
$ ./parser.exe input.txt
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical">

  <TextView
    android:layout_height="20"
error: syntax error, unexpected android:id, expecting android:layout_width on line 9
Parsing failed
    android:id
user@DESKTOP-JVDGAFH ~/tmp
$
```

ΑΣΚΗΣΗ 2 ΑΠΟΤΕΛΕΣΜΑΤΑ

ΓΕΝΙΚΑ ΣΤΗΝ ΑΣΚΗΣΗ 2 ΔΕΝ ΕΙΧΑΜΕ ERRORS ΑΛΛΛΑ ΔΕΝ ΕΤΡΕΧΑΝ. ΕΜΦΑΝΙΖΑΝ ΑΥΤΟ ΤΟ ΜΗΝΥΜΑ

```
~/tmp
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical">

  <TextView
    android:layout_width="20"
    android:layout_height="20"
    android:id="TV1"
    android:text="Dummy text"/>

  <RadioGroup
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    radio_button_count="2">

    <RadioButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="RB1"
      android:text="Option 1"/>

    <RadioButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
Segmentation fault (core dumped)

user@DESKTOP-JVDGAFH ~/tmp
$ _
```

ΑΣΚΗΣΗ 3 ΑΠΟΤΕΛΕΣΜΑΤΑ

ΕΔΩ ΕΧΟΥΜΕ ΠΡΟΣΘΕΣΗ ΤΟ ΥΠΟΧΡΕΩΤΙΚΟ ΣΤΟΙΧΕΙΟ `radio_button_count` ΑΛΛΑ ΟΠΩΣ ΚΑΙ ΣΤΗΝ ΔΥΟ ΔΕΝ ΓΙΝΕΤΑΙ ΕΛΕΓΧΟΣ ΓΙΑ ΤΟΝ ΑΡΙΘΜΟ ΤΩΝ ΚΟΥΜΠΙΩΝ

```
~/tmp
$ ./parser.exe input.txt
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical">

  <TextView
    android:layout_width="20"
    android:layout_height="20"
    android:id="@idTV1"
    android:text="Dummy text"/>

  <RadioGroup
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    radio_button_count="2">

    <RadioButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@idRB1"
      android:text="Option 1"/>

    <RadioButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@idRB2"
      android:text="Option 2"/>

  </RadioGroup>

  <!-- This is a comment -->

  <RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:src="image_name"
      android:padding="50"/>

  </RelativeLayout>
</LinearLayout>

The compile was successful!
user@DESKTOP-JVDGAFH ~/tmp
$ _
```


