

**ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ FLEX – BISON**

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2022-2023

ΣΥΝΤΑΚΤΕΣ :

|  |  |  |  |
| --- | --- | --- | --- |
| Ονοματεπώνυμο | ΑΜ | ΕΤΟΣ | EMAIL |
| ΜΙΡΑ ΙΣΛΑΜΑΪ | 1070736 | 5ο | miraslamaj@gmail.com |
| ΔΗΜΗΤΡΑ ΜΑΥΡΙΔΟΥ | 1070770 | 6ο | up1070770@upnet.gr |
| ΧΡΥΣΟΥΛΑ ΚΑΤΣΑΝΤΑ | 1067503 | 5ο | up1067503@upnet.gr |

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

Αρχές Γλωσσών Προγραμματισμού & Μεταφραστών

ΔΙΔΑΣΚΟΝΤΕΣ: Ιωάννης Γαροφαλάκης, Σπυρίδων Σιούτας, Παναγιώτης Χατζηδούκας

# BNF ΓΡΑΜΜΑΤΙΚΗ

---Bnf Declaration for android interface xml compiler ---

#### ELEMENTS

ΣΤΟΙΧΕΙΑ ΜΕ ΠΕΡΙΕΧΟΜΕΝΟ

<RootElement> ::= <LinearElement>

| <RelativeElement>

<LinearElement> ::= <LinearStartTag> <elements> <LinearEndTag>

<LinearStartTag> ::= “<” LinearLayout <mandContent> <linear\_optional> ”>”

<linear\_optional> ::= <id\_attribute><orientation\_attribute>

| <id\_attribute>

<LinearEndTag> ::= “<”/LinearLayout”>”

<elements> ::= <RootElement> <elements>

| <RadioGroup> <elements>

| <textview> <elements>

| <imageview> <elements>

| <buttonElement> <elements>

| <progressbar> <elements>

| <RootElement>

|<RadioGroup>

| <textview>

| <imageview>

| <buttonElement>

| <progressbar>

<RelativeElement> ::= <RelativeStartTag> <relative\_elements> <RelativeEndTag>

<RelativeStartTag> ::= “<”RelativeLayout <mandContent><root\_optional>”>”

<root\_optional> ::= <id\_attribute> | e

<RelativeEndTag> ::= “<”/RelativeLayout”>”

<relative\_elements> ::= <elements> | e

<RadioGroup> ::= <RadioGroupStart><radio\_element><RadioGroupEnd>

<RadioGroupStart> ::= “<”RadioGroup <radiogroup\_mand> <radiogroup\_opt> “>”

<radiogroup\_mand> ::= <mandcontent> <numofButtons\_attribute>

<radiogroup\_opt> ::= <id\_attribute><checkedbutton\_attribute>

| <id\_attribute>

| <checkedbutton\_attribute>

| e

<RadioGroupEnd> ::= “<”/RadioGroup”>”

<radio\_element> ::= <RadioButton> <radio\_element>

| <RadioButton>

ΣΤΟΙΧΕΙΑ ΔΙΧΩΣ ΠΕΡΙΕΧΟΜΕΝΟ

<buttonElement>::=“<”Button <button\_mandatory\_cont> <button\_optional\_cont> /“>”

<button\_mandatory\_cont> ::= <mandcontent> <text\_attribute>

<button\_optional\_cont>::= <id\_attribute> <padding\_attribute>

|<id\_attribute> | <padding\_attribute> | e

<textview> ::= “<”TextView <button\_mandatory\_cont><textview\_opt> /”>”

<textview\_opt> ::= <id\_attribute> <textColor\_attribute>

|<id\_attribute>

| <textColor\_attribute> |e

<imageview> ::= “<”ImageView <imageview\_mand><button\_optional\_cont>/”>”

<imageview\_mand> ::= <mandcontent> <src\_attribute>

<progressbar> ::= “<”ProgressBar <mandcontent> <progressbar\_opt>/”>”

<progressbar\_opt> ::= <id\_attribute> <max\_attribute> <progress\_attribute>

| <id\_attribute> <max\_attribute>

| <id\_attribute> <progress\_attribute>

| <id\_attribute> | <progress\_attribute>

| <max\_attribute> | e

<RadioButton>::= “<”RadioButton <button\_mandatory\_cont> <root\_optional> /”>”

#### ATTRIBUTES

<mandContent> ::= <androidTag> <WIDTH>= <value> <androidTag> <HEIGHT>= <value>

<text\_attribute> ::= <androidTag> text= <string>

<id\_attribute> ::= <androidTag> id= <string>

<numofButtons\_attribute> ::= <androidTag> numberOfButtons= <arithmetic\_value>

//Σχολιο : <arithmetic\_value> χρησιμοποιείται για να ορίσουμε τιμές int οι οποίες θα βρίσκονται σε “ ”

<padding\_attribute> ::= <androidTag> padding= <arithmetic\_value>

<textColor\_attribute> ::= <androidTag> textColor= <string>

<src\_attribute> ::= <androidTag> src= <string>

<max\_attribute> ::= <androidTag> max= <arithmetic\_value>

<progress\_attribute> ::= <androidTag> progress = <arithmetic\_value>

<checkedbutton\_attribute> ::= <androidTag> checkedButton= <string>

<arithmetic\_value> ::= “ “ “<number>” ” “

<orientation\_attribute>::= <androidTag> orientation=“ ” ” vertical” “ “

| <androidTag> orientation=“ " ” horizontal “ " “

<androidTag> ::= android:

<string> ::= “ “ “ <many\_chars>“ “ “

<value> ::= <number>

| <WRAP\_CONTENT>

|<MATCH\_PARENT>

<character> ::= <letter> | <digit> | <special\_character>

<many\_chars>::= <character> <many\_chars> | <character>

<letter> ::= a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

<number>::= <digit\_nonzero> <digit> <number>

| <digit\_nonzero> <digit>

| <digit\_nonzero>

<digit>::=<digit\_nonzero> | 0 //αυτό γιατι ένας θετικός ακέραιος δεν γινεται να ξεκινάει απο 0

<digit\_nonzero> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<special\_character> ::= ! | @ | # | $ | % | ^ | & | \* | ( | ) | \ | - | \_ | = | + | [ | \ | ] | { | } | “|” | ; | : | ' | " | , | . | < | > | ? | / | ~ | `

### ΠΑΡΑΔΟΧΕΣ

* Για δική μας διευκόλυνση, θεωρήσαμε πως τα χαρακτηριστικά των στοιχείων θα μπαίνουν μόνο με μια συγκεκριμένη σειρά. Συγκεκριμένα , θα πρέπει να γράφονται πάντα τα υποχρεωτικά χαρακτηριστικά πρώτα και μετά τα προαιρετικά του κάθε στοιχείου. Τα χαρακτηριστικά “android:layout\_width” και “android:layout\_height” (που είναι και καθολικά υποχρεωτικά χαρακτηριστικά) θα πρέπει να γραφτούν πρώτα για να θεωρηθεί σωστή η σύνταξη του xml αρχείου. Στην συνέχεια, θα πρέπει να γραφτεί το ειδικό υποχρεωτικό χαρακτηριστικό του στοιχείου , εάν έχει , για παραδειγμα στο TextView μετα τα χαρακτηριστικα “layout\_width” και “layout\_height” θα πρέπει να εμφανίζεται το “android:text” και τέλος μπαίνουν τα προαιρετικά χαρακτηριστικά. Γενικότερα , ακολουθήθηκε η σειρά με την οποία τα χαρακτηριστικά αυτα εμφανίζονται στην εκφώνηση της άσκησης .
* Χρησιμοποιήθηκαν τα διπλά εισαγωγικά **“ ”**, με την λογική που παρατίθεται στο παρακάτω λινκ:

<https://eng.libretexts.org/Bookshelves/Computer_Science/Programming_and_Computation_Fundamentals/Foundations_of_Computation_(Critchlow_and_Eck)/04%3A_Grammars/4.02%3A_Application_-_BNF>

Ειδικότερα, έγινε η χρήση τους μόνο για τα “<” και “>” , διότι αυτά τα δύο σύμβολα έχουν συγκεκριμένη σημασία στην BNF γραμματική και θα θέλαμε να αποφύγουμε την σύγχυση με την ύπαρξή τους στην γλώσσα XML. Το ίδιο έγινε και με τα σύμβολα **“** και **|**, που χρησιμοποιούνται και στην xml και στην bnf.

# FLEX ΚΩΔΙΚΑΣ

### DECLARATIONS

Αρχικά ορίζονται οι βιβλιοθήκες που θα χρησιμοποιήσουμε , όπως και οι συναρτήσεις και μεταβλητές. Συγκεκριμένα, η συνάρτηση yyerror() ,ορίζεται στον τμήμα των auxiliary functions, δηλώνεται για την διαχείριση των ορθογραφικών σφαλμάτων. Η μεταβλητή errors αρχικοποιείτε και χρησιμοποιείται ως μετρητής σφαλμάτων.

Δηλώνεται και μία μεταβλητή τύπου int ενσωματωμένη στο Flex (%option yylineno), όπου κάθε φορά που θα συναντά το Flex τον χαρακτήρα νέας γραμμής ('\n'), η μεταβλητή θα αυξάνεται κατά 1 αυτόματα.

Σημαντική , είναι και η μεταβλητή “int invalid\_comments”, που θα χρησιμοποιηθεί για να ορίσουμε σε συγκεκριμένες σημεια του xml κώδικα πως δεν επιτρέπεται η ύπαρξη σχολίων.

Δηλώθηκαν επίσης δύο exclusive start condition, για την περιγραφή των τιμών μέσα σε διπλά εισαγωγικά (%x VALUE) και για την περιγραφή των σχολίων (%x COMMENTS) , όπως αναφέρεται στο Flex manual από την σελ. 21.

Τέλος, δίνονται κάποιες ονομασίες για κάποιες κανονικές εκφράσεις που θα χρειαστούν, με τις ποιο σημαντικές τα {ΙΝΤ} και {STRING}. Το {INT}, περιγράφει όλους τους ακέραιους αριθμούς μεγαλύτερο ή ίσο του μηδενός. Το {STRING} περιγράφει αλφαριθμητικά που ξεκινούν από τουλάχιστον ένα γράμμα και μπορούν να έχουν ανάμεσα ή στο τέλος κάτω παύλα “\_” , παύλα “-”κενό “ ” και αριθμούς. Κάποια παραδείγματα : “match\_parent”, “TV2”, “button-12 red” ενώ το “12something” θεωρείται λάθος.

%%

### TRANSLATION RULES

Στο τμήμα των translation rules ορίστηκαν αρχικά όλες οι λέξεις-κλειδιά , καθώς ο flex λεκτικός αναλυτής αναζητεί από πάνω προς τα κάτω εάν ένας κανόνας υφίσταται στο αρχείο που εξετάζουμε. Συγκεκριμένα οι λέξεις-κλειδιά που ορίστηκαν είναι όλα τα ονόματα των στοιχείων (LinearLayout, RelativeLayout κλπ.) και τα χαρακτηριστικά των στοιχείων (layout\_width, layout\_height, textColor κλπ.) ξεχωριστά από την λέξη-κλειδί “android:” που επίσης ορίζουμε.

Μετά τις λέξεις-κλειδιά , περιγράφονται οι τιμές που παίρνουν τα χαρακτηριστικά με τον ύπο-αναλυτή <VALUE>. Μέσα σε αυτόν τον αναλυτή ,που εφόσον είναι exclusive δεν επηρεάζεται από τους κανόνες “έξω” από αυτόν, μόλις βρεθεί ένας ακέραιος σύμφωνα με τον ορισμό του {INT} μέσα σε εισαγωγικά επιστρέφεται ένα συγκεκριμένο token για τους αριθμούς, μόλις βρεθεί ένα αλφαριθμητικό που συμπίπτει με τον ορισμό του {STRING} επιστρέφετε το αντίστοιχο token. Ορίστηκαν και εδώ κάποιες συγκεκριμένες λέξεις κλειδιά (όπως το “match\_parent”) και σε οποιαδήποτε άλλη περίπτωση έχουμε να κάνουμε με σφάλματα που τα διαχειρίζουμε μέσω της yyerror , εμφανίζοντας ένα μήνυμα λάθους.

Με έναν ίδιο τρόπο ορίζονται και τα σχόλια με τον υπο-αναλυτή <COMMENTS>. Σαν κανόνες σε αυτόν τον υπο-αναλυτή έχουμε πως αγνοούμε ότι υπάρχει ενδιάμεσα στα “<!--” και “-->” , με μόνη εξαίρεση το “--” που μόλις διαβαστεί από το flex εμφανίζεται μήνυμα ορθογραφικού λάθους. Επιπλέον αν βρεθεί η τιμή invalid\_comment να είναι ίση με 1 , σημαίνει πως βρισκόμαστε σε περιοχή κώδικα που δεν επιτρέπονται τα σχόλια και έτσι εκτυπώνεται ένα κατάλληλο μήνυμα μέσω της yyerror().

Και στους δύο υπο-αναλυτές άμα τελειώσει το αρχείο χωρίς να τους κλείσουμε, τα <comment> με “-->” και τα <VALUE> με διπλό εισαγωγικό **“** , εμφανίζεται σφάλμα.

Πολύ σημαντικό ήταν να ορίσουμε κανόνες για ετικέτες αρχής , ετικέτες τέλους και για το “=”, όπου αυτό βλέπουμε στην συνέχεια. Συγκεκριμένα στις ετικέτες αρχής θέτουμε το invalid\_comments να γίνει ίσο με 1 , διότι μέσα στην περιγραφή των στοιχείων δεν επιτρέπονται τα σχόλια, και στις ετικέτες τέλους θέτουμε το invalid\_comments ίσο με 0 , όπως αρχικά για να δείξουμε πως επιτρέπονται πλέον τα σχόλια.

Με κάθε κανόνα , επιστρέφεται ένα συγκεκριμένο token (λεκτική μονάδα) εκτός από τα κενά και τα σχόλια.

%%

### AUXILIARY FUNCTIONS

Εδώ ορίσαμε την συνάρτηση yyerror() , όπου αναφέραμε και παραπάνω, εκτυπώνει το σφάλμα που προέκυψε και σε ποια γραμμή. Άμα εμφανιστούν 10 σφάλματα συνολικά τερματίζεται η εκτέλεση του προγράμματος. (Μεταφερθηκε στην πορεία στο .y αρχειο).

# BISON ΚΩΔΙΚΑΣ

Στο πρώτο κομμάτι του bison, ορίσαμε τις απαραίτητες αρχικοποιήσεις των βιβλιοθηκών της c, καθώς και κάποιες βοηθητικές μεταβλητές για την εκτέλεση της συντακτικής ανάλυσης. Σε αυτό το κομμάτι καλέσαμε επίσης τις συναρτήσεις ,που ορίζονται στο flex, με το keyword extern. Με αυτό τον τρόπο καταφέρνουμε να μεγαλώσουμε την εμβέλεια ορατότητας της c με σκοπό να έχουμε πρόσβαση στα εξής του flex :

**yyin**, **yylineno**(που έχει περιγραφεί παραπάνω), **yyparse**() για να την επικυροποίηση των δεδομένων εισόδου, **yylineno , yyval** (τιμές της εισόδου) **, yylex.**

### DEFINITIONS

Μετά τις δηλώσεις σε c προχωρήσαμε στην δήλωση των tokens. Σε αυτό το σημείο δηλώνονται όλα τα tokens που έχουν δηλωθεί ότι επιστρέφονται από το flex και μέσω του bison δημιουργείται το αρχείο bisonFile.tab.h που περιλαμβάνει κάποιες μοναδικές τιμες (κωδικούς) για όλα τα tokens. Το σημείο αυτό είναι πολύ σημαντικό καθώς εδώ γίνεται η σύνδεση των εισόδων που αναγνωρίζει το flex με τους κανόνες που υπακούν στην γλώσσα που ορίζουμε.

### RULES

Το κομμάτι των κανόνων στο bison είναι και αυτό που ορίζει το συντακτικό που πρέπει να ακολουθεί οποιοδήποτε αρχείο εισόδου. Ανάλογα με την αλληλουχία χαρακτήρων που διαβάζει το flex, επιστρέφει τα αντίστοιχα tokens στο bison, όπου και μέσω των εκφράσεων ορίζονται οι πιθανοί αποδεκτοί συνδιασμοί που αναγνωρίζει ο compiler που σχεδιάσαμε.

Για την σύνθεση των διαφορετικών κανόνων πρώτα σχεδιάσαμε τους κανόνες με το BNF και μετά ανάλογα με την σύνταξη του bison κάναμε τις απαραίτητες αλλαγές. Στις περιπτώσεις που θέλαμε να έχουμε το κενό αλφαριθμητικό ως είσοδο, χρησιμοποιήσαμε το %empty.

Στην δήλωση του κανόνα που αφορά το “max\_value”, για το ερώτημα 2.e. ορίσαμε την τιμή της μεταβλητής android\_max\_value με την τιμή που διαβάζει ο parser σε εκείνη την περίπτωση ώστε να μπορέσουμε να “αποθηκεύσουμε” την τιμή του max. Στην συνέχεια ορίζουμε τον κανόνα για το progress\_attribute με με την χρήση της μεταβλητής που διατηρεί το value του max και σε κώδικα c πραγματοποιούμε τις κατάλληλες συγκρίσεις για να ικανοποιηθούν οι περιορισμοί του ερωτήματος. Έτσι με την βοήθεια αυτών των μεταβλητών καταφέρνουμε να ελέγχουμε τις τιμές που παίρνει το progress\_value ανάλογα με τις τιμές του max.

Μετά την εμφάνιση των τελευταίων %% συμβόλων προχωρήσαμε στην δήλωση των συναρτήσεων.

### SUBROUTINES

Σε αυτή την περιοχή του bison μεταφέραμε την συνάρτηση yyerror που στην αρχή είχαμε ορίσει στο flex. Η συνάρτηση σε αυτή την περίπτωση εμφανίζει κατάλληλα μυνήματα για να ενημερώσει τον χρήστη σχετικά με τα λάθη που αναγνώρισε το bison σε συνδιασμό με το flex.

Επίσης σε αυτό το σημείο έχουμε όσες συναρτήσεις προέκυψαν από τα ερωτήματα 2 και 3.

Πιο συγκεκριμένα, η συνάρτηση idUnique χρησιμοποιείται για την αποθήκευση των id των στοιχείων σε array και τον έλεγχο μοναδικότητάς τους. Αρχικά, υπάρχει ένας βρόχος που ελέγχει τις ήδη υπάρχουσες τιμές id και αν κάποια από αυτές είναι ίδια με την τρέχουσα τιμή, εμφανίζει ανάλογο μήνυμα λάθους. Στη συνέχεια, υπάρχει μία if, η οποία χρησιμεύει για την αύξηση του μεγέθους του array, κάθε φορά που δεν υπάρχει χώρος για την αποθήκευση κάποιας τιμής. Τέλος, το μοναδικό id προστείθεται στο array.

Η συνάρτηση save\_rb\_id() αποθηκεύει τα id των RadioButton που βρίσκονται εντός ενός Radiogroup σε ένα προσωρινό πίνακα. Γίνεται αρχικά έλεγχος αν βρισκόμαστε στο περιεχόμενο ενός RadioGroup, αν δεν βρισκόμαστε εκεί , δεν μας ενδιαφέρει η αποθήκευση του id και δεν εκτελείται τίποτα. Όταν βρεθούμε μέσα σε ένα RadioGroup , αποθηκευούμε την τιμή του id στον πίνακα. Ο πίνακας έχει τόσες θέσεις όσα είναι και τα RadioButton ενός RadioGroup. Το πλήθος των Radiogroup μετριέται με τον μετρητη rbs. Το πρώτο id ενός RadioButton αποθηκεύεται στην πρωτη θέση του πίνακα , το δεύτερο στην δεύτερη και ούτω καθεξής.

Η συνάρτηση cmp\_check() συγκρίνει το όρισμα του χαρακτηριστικού checkedButton με αυτά των id των εμφωλευμένων RadioButton. Αν δεν εχει οριστεί το χαρακτηριστικό checkedButton , το όρισμα είναι κενό , και δεν εκτελείται κάποια σύγκριση. Σε άλλη περίπτωση όμως, ελέγχουμε, με ένα for loop, κάθε τιμή id που αποθηκεύτηκε από την save\_rb\_id(), αν αρχικά είναι NULL. Αν δεν είναι NULL αυξάνουμε τον μετρητή y (που αρχικά ήταν 0) και εξετάζεται αν η τιμή του id είναι ίδια με αυτή του checkedButton. Στην περίπτωση που είναι ίδια, αυξάνουμε τον μετρητη x (που αρχικά ήταν 0). Αφού τερματίσει ο for βρόγχος μας, κοιτάζουμε τις τιμές των x και y. Με x=0 σημαίνει πως δεν είχαμε κανένα ίδιο όρισμα με κάποιο id και παράλληλα αν y>0 σημαίνει πως είχαν οριστεί τιμές id στα RadioButtons, οπότε αν x==0 και y>0 εμφανίζεται ένα σφάλμα για το γεγονός ότι δεν χρησιμοποιήθηκε μια τιμή id ενός RadioButton για το όρισμα του checkedButton.

Τέλος, εδώ ορίζουμε την main συνάρτηση όπου ανοίγουμε το αρχείο εισόδου που δίνει ο χρήστης και το διατρέχουμε με το flex. Μέσω της yyparse κάνουμε διάβασμα των αντίστοιχων tokens και αντιστοιχία στους κανόνες. Αν δεν υπάρχει κάποιο error από την εκτέλεση τότε εμφανίζουμε το κατάλληλο μήνυμα επιτυχίας.

# ΚΑΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ

### MAKE FILE

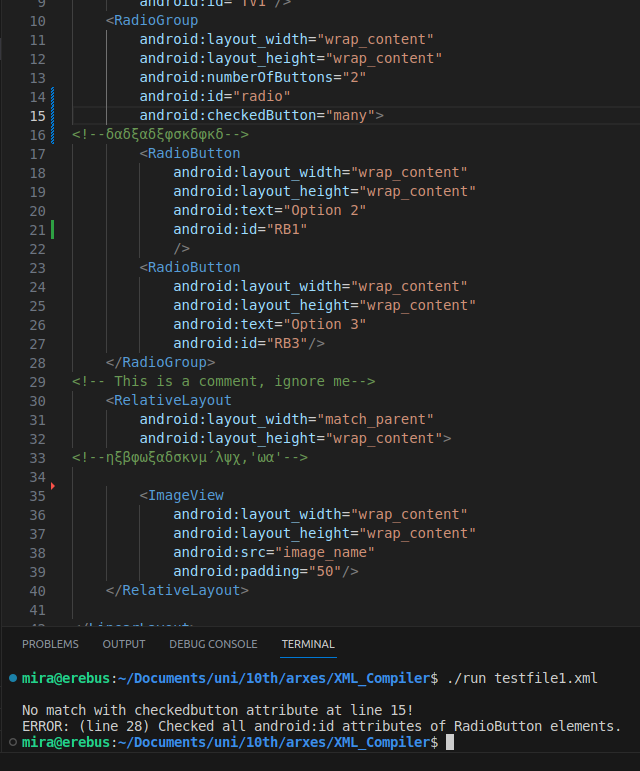
Η εκτέλεση του flex και bison έγινε σε ένα περιβάλλον linux με την προεγκατάσταση των flex, bison μέσω terminal. Για την εκτέλεση του κάθε αντίστοιχου αρχείου πραγματοποιούσαμε επαναλαμβανόμενες κινήσεις, οπότε μέσω του αρχείου Makefile καταγράψαμε τις γραμμές που επανεκτελούσαμε στο terminal και με την εντολή make στο σωστό directory εκτελούσαμε ταυτόχρονα bison,flex και gcc compiler. Επομένως, καταλήγαμε με το αρχείο **run ,**που είναι εκτελέσιμο, μέσω του οποίου μπορούμε να ελέγξουμε το αρχείοο εισόδου που ειπθυμούμε.

### TESTING ΚΑΙ ΣΤΙΓΜΙΟΤΥΠΑ ΟΘΟΝΗΣ ΑΠΟ ΤΗΝ ΕΚΤΕΛΕΣΗ

Παρακάτω, παραθέτουμε κάποια στιγμιότυπα οθόνης, όπου δοκιμάζουμε τον αναλυτή μας με διαφορετικά test files.

Περίπτωση: To checked button δεν έχει ιδια τιμή με κανένα radiobutton id

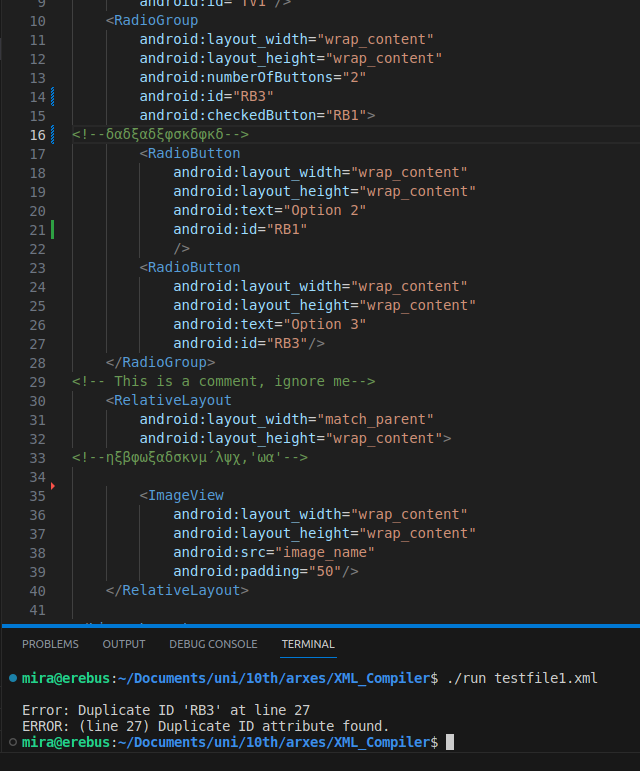
Αποτέλεσμα: Εμφάνιση σφάλματος



Περίπτωση: Τα radio group και radio button έχουν ίδιο χαρακτηριστικό android:id.

Αποτέλεσμα: Μήνυμα σφάλματος γιατί το συγκεκριμένο χαρακτηριστικό πρέπει να είναι μοναδικό.

Επιπρόσθετες πληροφορίες: Εδώ φαίνεται και πως το checkedButton έχει την ιδια τιμή με το id του πρώτου εμφωλευμένου radioButton και πως αυτο δουλευει σωστά καθώς δεν έχουμε μηνύματα σφάλματος για αυτό.



Περίπτωση: Το χαρακτηριστικό android:padding έχει αλφαριθμητικό.

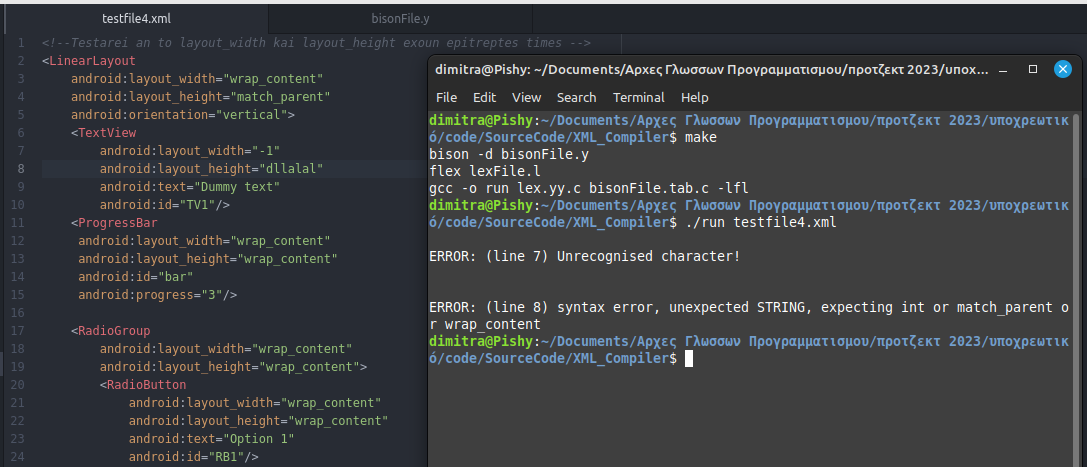
Αποτέλεσμα: Εμφάνιση σφάλματος γιατί παίρνει μόνο θετικό ακέραιο.

A screen shot of a computer program

Description automatically generated

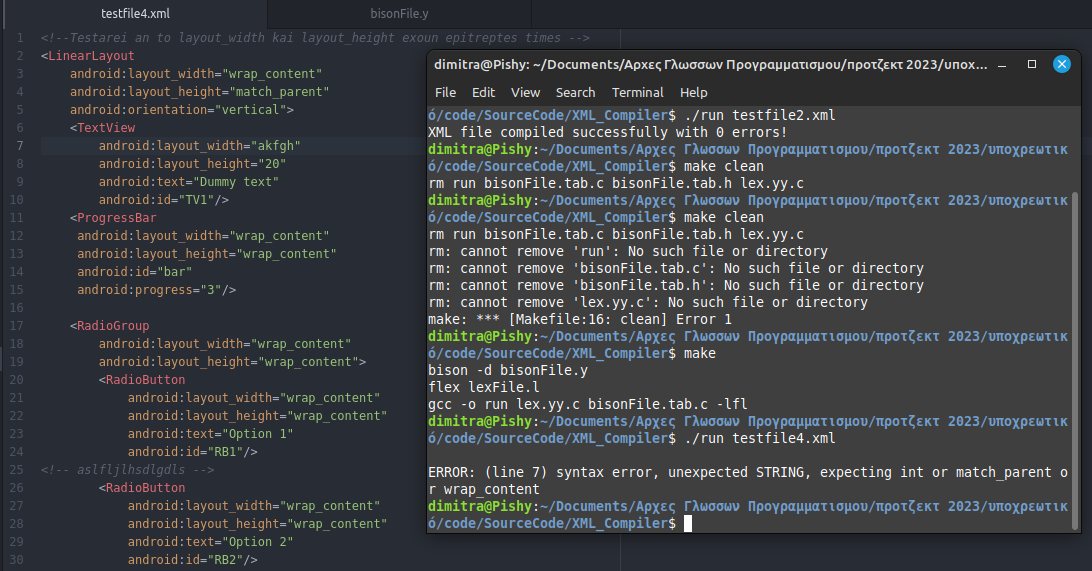
Περίπτωση: Τα χαρακτηριστικά android:layout\_width και android:layout\_height έχουν μη επιτρεπτές τιμές

Αποτέλεσμα: Εμφάνιση σφάλματος



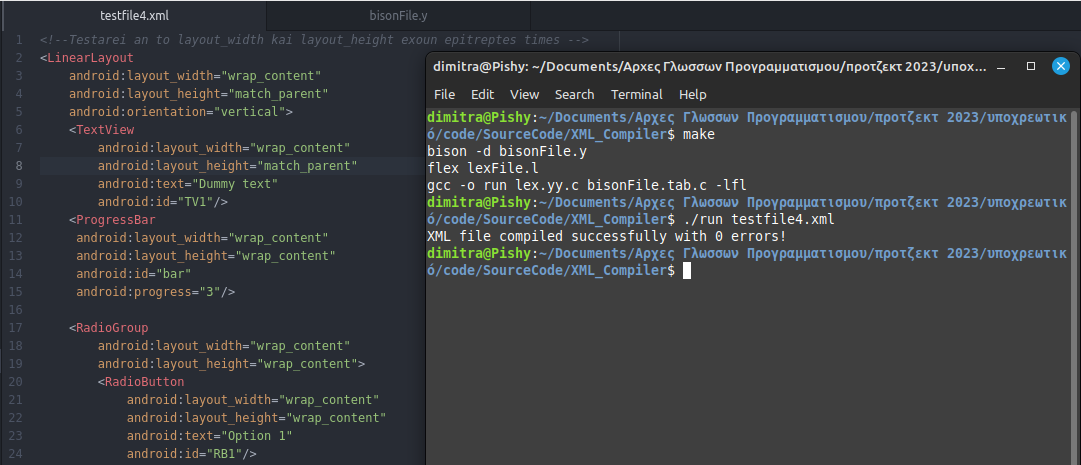
Περίπτωση: Το χαρακτηριστικό android:layout\_width έχει μη επιτρεπτή τιμή

Αποτέλεσμα: Εμφάνιση σφάλματος



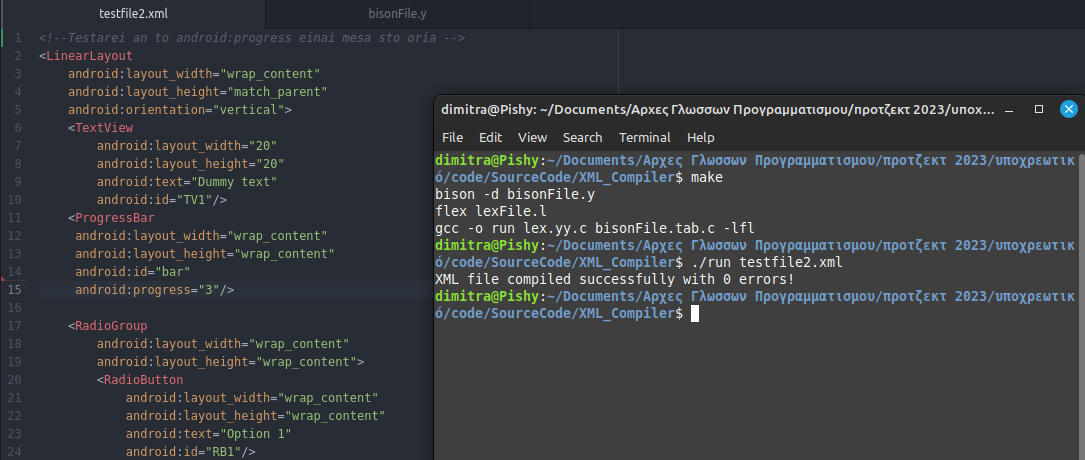
Περίπτωση: Τα χαρακτηριστικά android:layout\_width και android:layout\_height έχουν επιτρεπτές τιμές.

Αποτέλεσμα: Δεν υπάρχει κανένα σφάλμα.



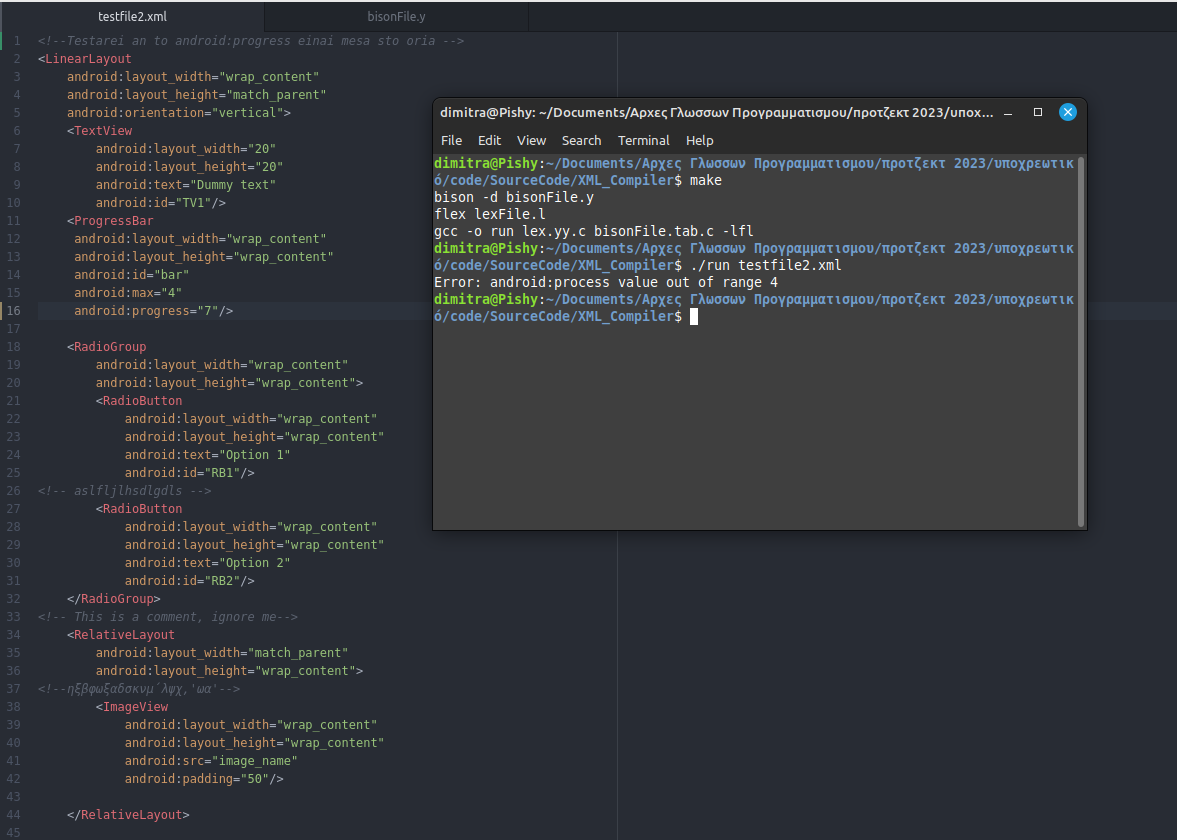
Περίπτωση: Το χαρακτηριστικό android:max δεν έχει οριστεί

Αποτέλεσμα: Δεν υπάρχει κανένα σφάλμα.



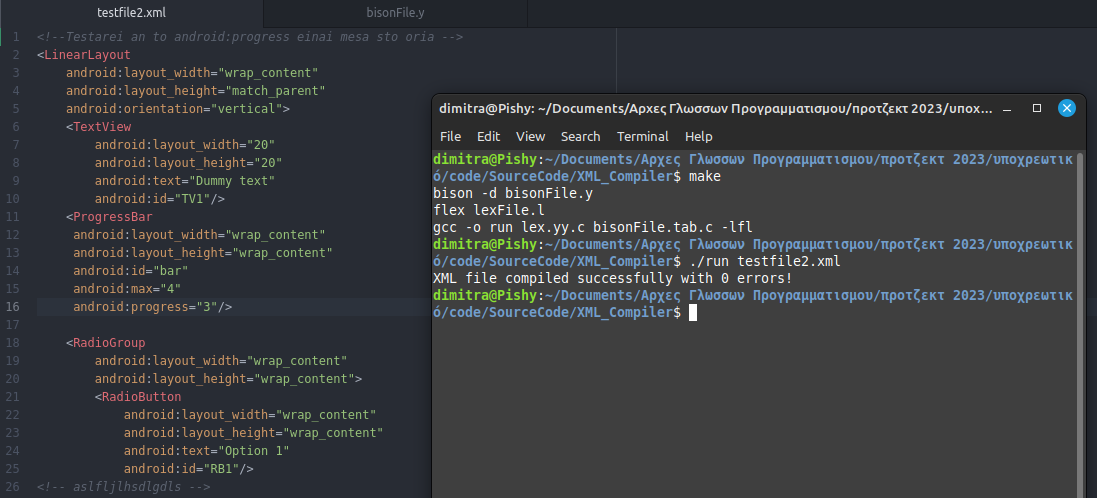
Περίπτωση: Το χαρακτηριστικό android:max είναι 4, και το χαρακτηριστικό android:progress είναι 7.

Αποτέλεσμα: Εμφάνιση σφάλματος, γιατί το android:progress πρέπει να είναι μεταξύ του 0 και της τιμής του android:max.



Περίπτωση: Το χαρακτηριστικό android:max είναι 4 και το χαρακτηριστικό android:progress είναι 3.

Αποτέλεσμα: Δεν υπάρχει κανένα σφάλμα.



Περίπτωση: Tο υποχρεωτικό χαρακτηριστικό numberOfButtons (που μπαίνει αμέσως μετά το android:layout\_height) για το radiogroup δεν υπάρχει.

Αποτέλεσμα: Εμφάνιση σφάλματος



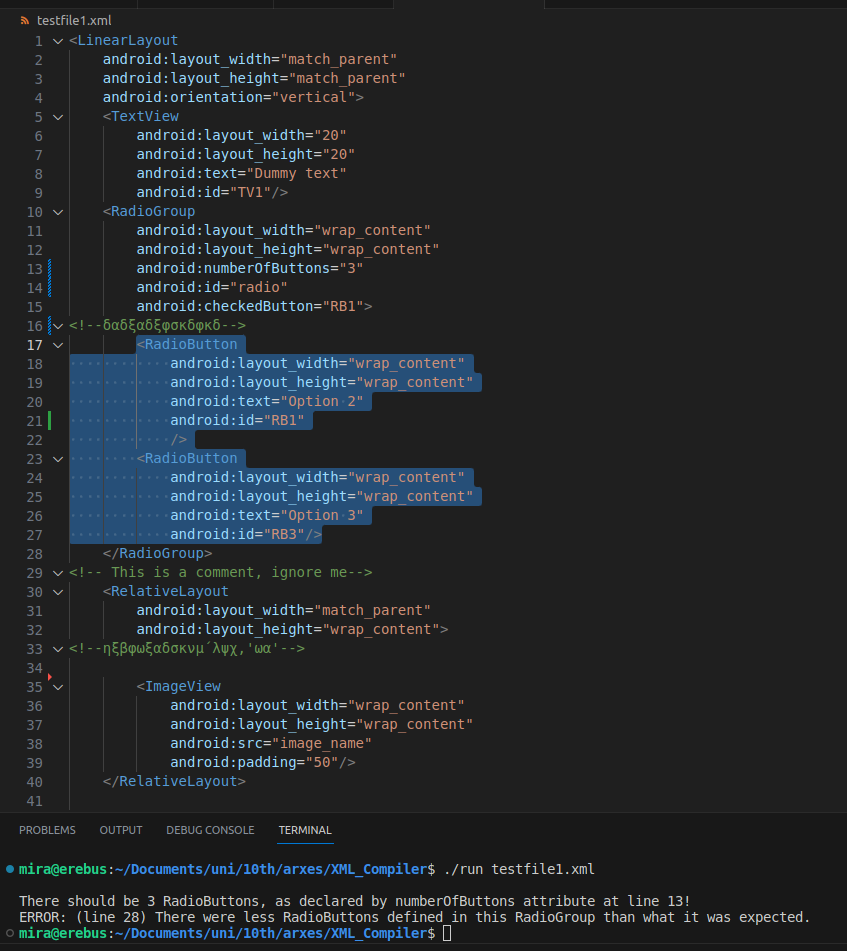
Περίπτωση: Tο υποχρεωτικό χαρακτηριστικό numberOfButtons είναι ίσο με αλφαριθμητικό , αλλά έπρεπε να είναι θετικού ακεραίου τύπου.

Αποτέλεσμα: Εμφάνιση σφάλματος



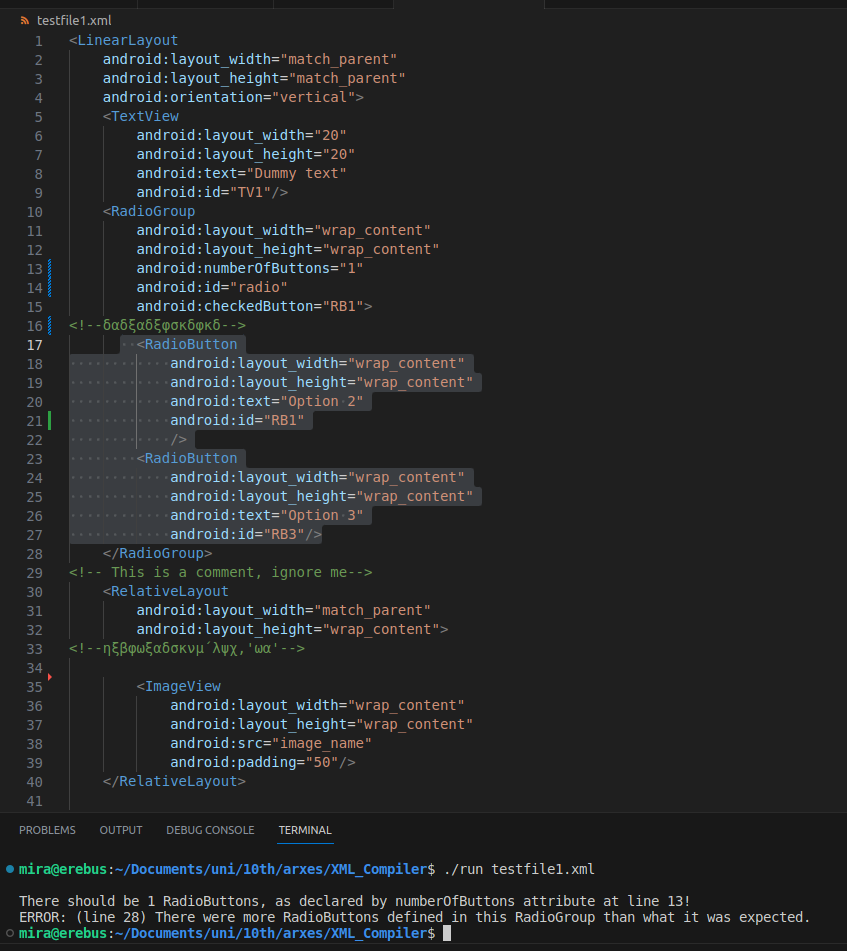
Περίπτωση: Το υποχρεωτικό χαρακτηριστικό numberOfButtons έχει την τιμή 3, ενώ υπάρχουν μόνο 2 radiobutton(λιγότερα)

Αποτέλεσμα: Εμφάνιση σφάλματος



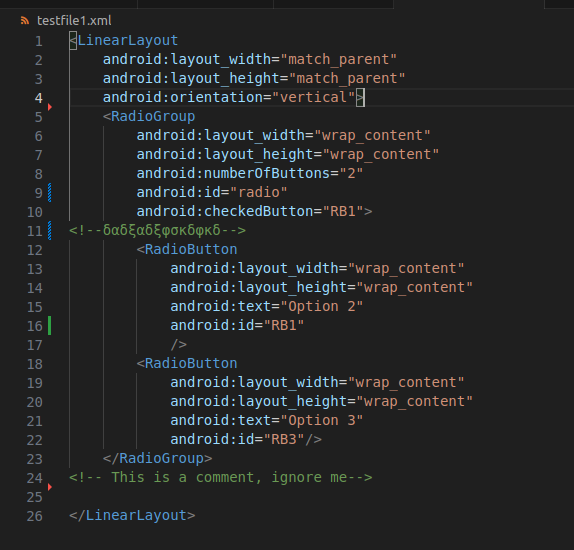
Περίπτωση: Το υποχρεωτικό χαρακτηριστικό numberOfButtons έχει την τιμή 1, ενώ υπάρχουν μόνο 2 radiobutton(περισσότερα)

Αποτέλεσμα: Εμφάνιση σφάλματος



Περίπτωση: Το υποχρεωτικό χαρακτηριστικό numberOfButtons έχει την τιμή 2, όσα και τα radiobutton

Αποτέλεσμα: Δεν υπάρχει λάθος.



A screen shot of a computer error

Description automatically generated

## Ο ΚΩΔΙΚΑΣ--lexFile.l

%{

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "bisonFile.tab.h" // including tokens from bison

void yyerror(const char\* err);

int invalid\_comment =0;

%}

/\* Μεταβλητή τύπου int ενσωματωμένη στο Flex. Κάθε φορά που θα συναντά το Flex τον χαρακτήρα νέας γραμμής ('\n'), η μεταβλητή θα αυξάνεται ΑΥΤΟΜΑΤΑ κατά 1 \*/

%option yylineno

%option noyywrap

%x VALUE

%x COMMENT

LETTER [a-zA-Z]

SPECIAL\_CHAR [\_\- ]

DIGIT [0-9]

INT [1-9]{DIGIT}\*|0

STRING {LETTER}+{SPECIAL\_CHAR}\*{LETTER}\*{DIGIT}\*{LETTER}\*

WHITESPACE [ \t]

%%

"LinearLayout" { return LAYOUT\_1; }

"RelativeLayout" { return LAYOUT\_2; }

"RadioGroup" { return RGROUP; }

"TextView" { return TEXTVIEW; }

"ImageView" { return IMAGEVIEW; }

"Button" { return BUTTON; }

"RadioButton" { return RBUTTON; }

"ProgressBar" { return PROGRESSBAR; }

"android:" { return ANDROIDTAG; }

"id" { return ID;}

"orientation" { return ORIENTATION; }

"text" { return TEXT; }

"textColor" { return TEXTCOLOR; }

"src" { return SOURCE; }

"padding" { return PADDING; }

"checkedButton" { return CHECK\_B; }

"max" { return MAX; }

"progress" { return PROGRESS; }

"layout\_width" { return WIDTH; }

"layout\_height" { return HEIGHT; }

"numberOfButtons" { yylval.integer=atoi(yytext); return NUM\_BUTTONS;}

\" {BEGIN(VALUE);}

<VALUE>\" {BEGIN(INITIAL);}

<VALUE>"vertical" { return VERTICAL; }

<VALUE>"horizontal" { return HORIZONTAL; }

<VALUE>"wrap\_content" { return WRAP\_CONTENT; }

<VALUE>"match\_parent" { return MATCH\_PARENT; }

<VALUE>{INT} { yylval.integer=atoi(yytext); return INT;}

<VALUE>{STRING} { yylval.string=strdup(yytext); return STRING;}

<VALUE>{INT}{STRING} {yyerror("String value can't start from integer!\n");}

<VALUE>. {yyerror("Unrecognised character!\n");}

<VALUE><<EOF>> {yyerror("You've reached the end of file, without closing a string value!\n");}

"<!--" {BEGIN(COMMENT);}

<COMMENT>"-->" {BEGIN(INITIAL);} //comments cannot be tokens we just ignore them , no return statements

<COMMENT>. { if(invalid\_comment==1){ yyerror("You can't comment here!\n");} }

<COMMENT>"--" {yyerror("Illegal arguement '--' used in comment!\n"); BEGIN(COMMENT);}

<COMMENT><<EOF>> {yyerror("You've reached the end of file, without closing the comment!\n");}

"<" { invalid\_comment=1; return START\_TAG; }

"/>" { invalid\_comment=0; return SMALL\_CLOSETAG; }

">" { invalid\_comment=0; return ENDTAG; }

"</" { invalid\_comment=1; return CLOSETAG; }

"=" { return ASSIGN; }

{WHITESPACE} { /\* Κενά (space) μέσα στο αρχείο - απλά τα αγνοούμε \*/ }

\n|\r\n { /\*το ιδιο με το whitespace και /r/n ειναι αλλαγή γραμμής για windows\*/}

. { yyerror("Unrecognised character!\n"); }

<<EOF>> { return T\_EOF;}

%%

//η μεταβλητη yytext ειναι ενσωματομενη στη flex και μας δείχνει την αναγνωρισμένη λέξη

//αντιστοιχα το yylineno μας δειχνει σε ποια γραμμή του κώδικα βρισκόμαστε

## Ο ΚΩΔΙΚΑΣ--bisonFile.y

%{

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "bisonFile.tab.h"

int android\_progress\_value; // To store the android:progress value

int android\_max\_value; // To store the android:max value

int errors=0;

extern FILE \*yyin; //Το yyin είναι ειδική μεταβλητή του Flex.

extern int yylineno;

extern int yyparse();

extern YYSTYPE yylval; // Token value

int yylex();

void yyerror(const char\* s);

//FUNCTIONS AND VALUES FOR COMPARING CHECKEDBUTTON WITH RADIOBUTTON IDS

void cmp\_check();

void save\_rb\_id(char\* id);

int rbs=1; //number of radio buttons (radiogroup must always contain at leat one , so we initialize it to 1)

int rb\_state=0; //state=1 for when we are defining a radiogroup else state=0

char \*\*radio\_button\_id;

char\* checked\_value;

int line[2];

int number\_value=0;

void check\_number();

//NEW CODE, BE CAREFUL

//function and variables for unique id

void idUnique(char\* id);

char\*\* unique\_ids = NULL; // Array pointer for storing unique IDs

int unique\_id\_count = 0; // Current number of unique IDs

int unique\_ids\_size = 0; // Current size of the array

%}

%define parse.error verbose

%union{

int integer;

char\* string;

}

%token <integer> INT "int"

%token <string> STRING

%token MATCH\_PARENT "match\_parent"

%token WRAP\_CONTENT "wrap\_content"

%token ASSIGN "="

%token START\_TAG "<"

%token SMALL\_CLOSETAG "/>"

%token ENDTAG ">"

%token CLOSETAG "</"

%token LAYOUT\_1 LAYOUT\_2

%token RGROUP

%token TEXTVIEW IMAGEVIEW

%token BUTTON

%token RBUTTON

%token PROGRESSBAR

%token ANDROIDTAG

%token WIDTH HEIGHT

%token ID

%token ORIENTATION

%token VERTICAL

%token HORIZONTAL

%token TEXT

%token TEXTCOLOR

%token SOURCE

%token PADDING

%token CHECK\_B

%token MAX

%token PROGRESS

%token <integer> NUM\_BUTTONS

%token T\_EOF 0

%type<string> id\_attribute checkedbutton\_attribute

%%

RootElement : LinearElement

| RelativeElement

LinearElement : LinearStartTag elements LinearEndTag

LinearStartTag : START\_TAG LAYOUT\_1 mandContent linear\_optional ENDTAG

linear\_optional : id\_attribute orientation\_attribute

| id\_attribute

| orientation\_attribute

| %empty

LinearEndTag : CLOSETAG LAYOUT\_1 ENDTAG

elements : RootElement elements

| RadioGroup elements

| textview elements

| imageview elements

| buttonElement elements

| progressbar elements

| RootElement

| RadioGroup

| textview

| imageview

| buttonElement

| progressbar

RelativeElement : RelativeStartTag relative\_elements RelativeEndTag

RelativeStartTag : START\_TAG LAYOUT\_2 mandContent root\_optional ENDTAG

root\_optional : id\_attribute

| %empty

RelativeEndTag : CLOSETAG LAYOUT\_2 ENDTAG

relative\_elements : elements

| %empty

RadioGroup : RadioGroupStart radio\_element RadioGroupEnd

RadioGroupStart : START\_TAG RGROUP radiogroup\_mand radiogroup\_opt ENDTAG {rb\_state=1;}

radiogroup\_mand : mandContent numofButtons\_attribute

radiogroup\_opt : id\_attribute checkedbutton\_attribute

| id\_attribute

| checkedbutton\_attribute

| %empty

RadioGroupEnd : CLOSETAG RGROUP ENDTAG { cmp\_check(); check\_number(); rb\_state=0; rbs=1;}

/\*initialize values as before\*/

radio\_element : RadioButton radio\_element

| RadioButton

buttonElement : START\_TAG BUTTON button\_mandatory\_cont button\_optional\_cont SMALL\_CLOSETAG

button\_mandatory\_cont : mandContent text\_attribute

button\_optional\_cont : id\_attribute padding\_attribute

| id\_attribute

| padding\_attribute

| %empty

textview : START\_TAG TEXTVIEW button\_mandatory\_cont textview\_opt SMALL\_CLOSETAG

textview\_opt : id\_attribute textColor\_attribute

| id\_attribute

| textColor\_attribute

| %empty

imageview : START\_TAG IMAGEVIEW imageview\_mand button\_optional\_cont SMALL\_CLOSETAG

imageview\_mand : mandContent src\_attribute

progressbar : START\_TAG PROGRESSBAR mandContent progressbar\_opt SMALL\_CLOSETAG

progressbar\_opt : id\_attribute max\_attribute progress\_attribute

| id\_attribute max\_attribute

| id\_attribute progress\_attribute

| id\_attribute

| max\_attribute

| progress\_attribute

| %empty

RadioButton : START\_TAG RBUTTON button\_mandatory\_cont root\_optional SMALL\_CLOSETAG {rbs++; }

/\*increment rbs counter after declaration of every radiobutton\*/

mandContent : ANDROIDTAG WIDTH ASSIGN value ANDROIDTAG HEIGHT ASSIGN value

text\_attribute : ANDROIDTAG TEXT ASSIGN STRING

id\_attribute : ANDROIDTAG ID ASSIGN STRING {save\_rb\_id($4); idUnique($4);}

padding\_attribute : ANDROIDTAG PADDING ASSIGN INT

textColor\_attribute : ANDROIDTAG TEXTCOLOR ASSIGN STRING

src\_attribute : ANDROIDTAG SOURCE ASSIGN STRING

max\_attribute : ANDROIDTAG MAX ASSIGN max\_value

max\_value: INT

{

android\_max\_value = $1;

}

progress\_attribute : ANDROIDTAG PROGRESS ASSIGN progress\_value

{

if (android\_progress\_value < 0 || android\_progress\_value > android\_max\_value && android\_max\_value!=0) {

fprintf(stderr, "Error: android:progress value out of range %d\n",android\_max\_value );

exit(1);

}

};

progress\_value: INT

{

android\_progress\_value = $1;

}

checkedbutton\_attribute : ANDROIDTAG CHECK\_B ASSIGN STRING {checked\_value=$4; line[0]=yylineno;}

orientation\_attribute : ANDROIDTAG ORIENTATION ASSIGN VERTICAL

| ANDROIDTAG ORIENTATION ASSIGN HORIZONTAL

numofButtons\_attribute : ANDROIDTAG NUM\_BUTTONS ASSIGN INT { line[1]=yylineno; number\_value=$4;}

value : WRAP\_CONTENT

| MATCH\_PARENT

| INT

%%

/\*void yyerror(const char \*s) {

fprintf(stderr, "Error: %s\n", s);

}\*/

void yyerror(const char\* err){

errors++;

printf("\nERROR: (line %d) %s\n",yylineno,err);

if(errors==10){

printf("\nMaximum number of errors found.\n");

exit(EXIT\_FAILURE);

}

}

/\*Function to check if numberOfButtons attribute is equal to the actual number of RadioButtons inside a RadioGroup\*/

void check\_number(){

int x=rbs-1;

if(x<number\_value){

printf("\nThere should be %d RadioButtons, as declared by numberOfButtons attribute at line %d!",number\_value,line[1]);

yyerror("There were less RadioButtons defined in this RadioGroup than what it was expected.");

}

else if(x>number\_value){

printf("\nThere should be %d RadioButtons, as declared by numberOfButtons attribute at line %d!",number\_value,line[1]);

yyerror("There were more RadioButtons defined in this RadioGroup than what it was expected.");

}

}

/\*Compare if the value of checkedButton is equal to one of the radiobuttons ids\*/

void cmp\_check(){

int x=0, y=0;

if(checked\_value!=NULL){

for(int i=0;i<rbs-1;i++){

if(radio\_button\_id[i]!=NULL){

if(strcmp(checked\_value,radio\_button\_id[i])==0){

x++;

//if x>0 one id matches the checkedButton attribute

}

y++;

//if y>0 at least one id attribute was not null

}

}

if(x==0 && y!=0){

//if there were no matches , but there was a not null id attribute

printf("\nNo match with checkedbutton attribute at line %d!",line[0]);

yyerror("Checked all android:id attributes of RadioButton elements.");

}

}

}

/\*save the string ids of the radiobuttons declared\*/

void save\_rb\_id(char\* id){

if(rb\_state==0){

/\*do nothing\*/

}

else if(rb\_state==1){

radio\_button\_id[rbs-1]=id;

}

else{

printf("some exception occured\n");

}

radio\_button\_id=(char\*\*)realloc(radio\_button\_id,(rbs+1)\*sizeof(char\*));

}

//NEW CODE, BE CAREFUL

void idUnique(char\* id){

// Check for duplicate ID

for (int i = 0; i < unique\_id\_count; i++) {

if (strcmp(id, unique\_ids[i]) == 0) {

fprintf(stderr, "\nError: Duplicate ID '%s' at line %d", id, yylineno);

yyerror("Duplicate ID attribute found.");

break;

}

}

// Resize the array if necessary

if (unique\_id\_count >= unique\_ids\_size) {

unique\_ids\_size += 10; // Increase the size by 10

unique\_ids = (char\*\*)realloc(unique\_ids, unique\_ids\_size \* sizeof(char\*));

}

// Add the unique ID to the array

unique\_ids[unique\_id\_count++] = strdup(id);

}

//END OF NEW CODE

int main(int argc, char\* argv[]) {

radio\_button\_id=(char\*\*)calloc(1,sizeof(char\*));

yyin = fopen(argv[1], "r");

if (yyin == NULL) {

printf("%s: File not found\n", argv[1]);

return 1;

}

yyparse();

if(errors==0){

printf("XML file compiled successfully with %d errors!\n", errors);

}

fclose(yyin);

return 0;

}