



الجمهورية العربية السورية

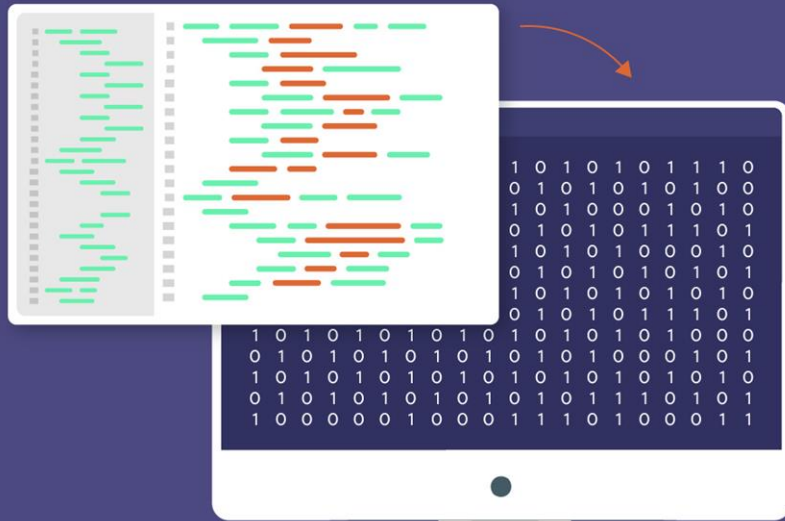
جامعة دمشق

كلية الهندسة المعلوماتية

قسم هندسة البرمجيات ونظم المعلومات

- Compiler Project -

codequotient



إعداد الطلاب:

ليلى خيرالله العودة

أحمد جهاد المبارك

مصطفى دياب حرب

صبا محمد طليعة

بإشراف:

د. باسم قصيبة

م. هدى جمال

جدول المحتويات

Contents

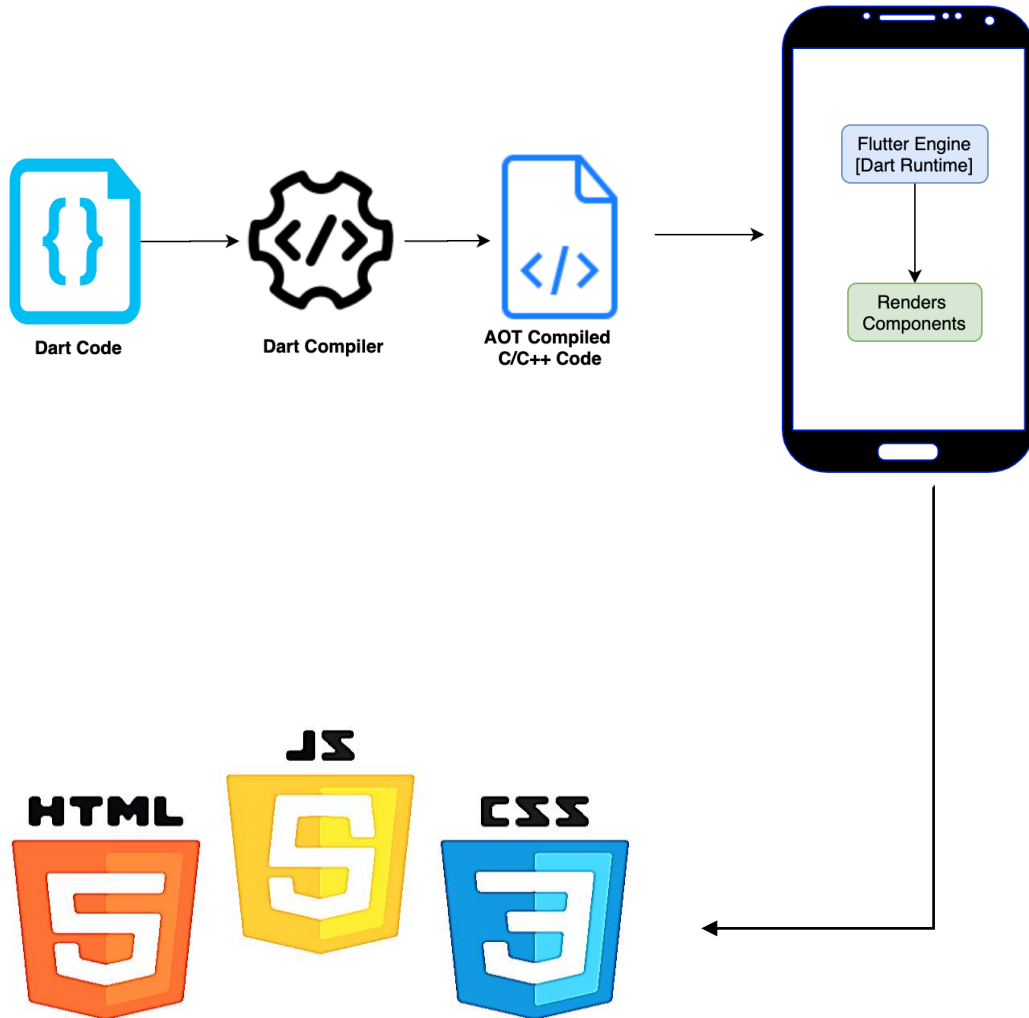
Error! Bookmark not defined.....	اللغة المصدر	❖
Error! Bookmark not defined.....	بنية اللغة :	•
Error! Bookmark not defined.....	شرح عن قدرات اللغة :	•
Error! Bookmark not defined.....	قواعد اللغة :	•
Error! Bookmark not defined.....	Lexer and Parser :	▪
Error! Bookmark not defined.....	بنى المعطيات الخاصة بجدول الرموز : .	❖
defined.		
12	بنى المعطيات الخاصة بالشجرة المولدة :	❖
14	بنية اللغة الهدف:	❖
15	خوارزمية التحويل إلى اللغة الهدف :	❖
Error! Bookmark not defined.....	HtmlGenerator :	✓
Error! Bookmark not defined.....	java script Generator:	✓
Error! Bookmark not defined.....	css generator	❖

مقدمة :

في هذا المشروع سنقوم بكتابة مترجم للغة برمجة (Dart) : وهي لغة مفتوحة المصدر تستخدم في برمجة الأجهزة الذكية وفي تطبيقات سطح المكتب وتعمل هذه اللغة على مختلف أنظمة التشغيل وكذلك تعمل مع مختلف متصفحات الإنترنت بكل سهولة ويسر وهي منصة قابلة للتوسع لبناء التطبيقات التي تعمل على شبكة الإنترنت.

مع إضافة قواعد جديدة تسمح بالانتقال بين الشاشات وتمرير البيانات بين الشاشات بالإضافة إلى اكتشاف وعرض الأخطاء الدلالية.

مع توليد كود مكافئ بلغة (HTML, JAVA SCRIPT, CSS) لشاشات التطبيق المكتوب بلغة المصدر التي قمنا بتطويرها.



قواعد اللغة:

```
grammar dart;

program: (libraryImport | libraryExport)* mainClass? ( widgetClass |
blocClass | stateClass | getClass | statement )* EOF ;

libraryImport
    :   IMPORT_   STRING_LITERAL (',' STRING_LITERAL)*   ';'
    ;

libraryExport
    :   EXPORT_   STRING_LITERAL (',' STRING_LITERAL)*   ';'
    ;

getClass:CLASS_ IDENTIFIER EXTENDS_ GETXCONTROLLER_ '{' (class_call
';' | statement)* '}' ;
blocClass : CLASS_ IDENTIFIER EXTENDS_ 'Bloc' '<' blocType ','
blocType '>' '{' blocClassArguments '}' ;
blocClassArguments : blocConstructor streamMethod;
blocConstructor : className '(' ')' ':' 'super' '(' expression ')'
';';
streamMethod : OVERRIDE_ 'Stream'<' blocType '>' 'mapEventToState'
'(' className IDENTIFIER ')' 'async*'
'{' 'yield' IDENTIFIER ';' '}' ;

widgetClass
    :   statelessWidget | statefulWidget ;

statefulWidget
    :   CLASS_ IDENTIFIER EXTENDS_ STATEFULWIDGET_ '{'
(stateClass|createState )* '}'
    ;
createState : OVERRIDE_ IDENTIFIER CREATESTATE_ '(' ')' '=>'
class_call ';';
stateClass
    :   CLASS_ IDENTIFIER EXTENDS_ ( 'State' | 'state') '<' IDENTIFIER
'>' '{' statement* widgetBuild? '}'
    ;

statelessWidget
    :   CLASS_ IDENTIFIER EXTENDS_ STATELESSWIDGET_ '{' (class_call ';' |
statement)* widgetBuild '}'
    ;
widgetBuild : OVERRIDE_ WIDGET_ BUILD_ '(' BUILDCONTEXT_ CONTEXT_
')' '{' statement* RETURN_ widgetLiteral ';' '}' ;

widgetLiteral: blocBuildWidget |obxWidget | scaffold | alertDialog |
textButton | textField | elevatedButton | materialApp | container |
sizedBox | gestureDetector | card | center | column | row | listview
| image | text;
materialApp : MaterialApp_ '(' (widgetAttribute (','
```

```

widgetAttribute)* )? ', '? ' )' ;
gestureDetector : GESTUREDETECTOR_ '(' (widgetAttribute (',' widgetAttribute)* )?
widgetAttribute)* )? ', '? ' )';

scaffold : SCAFFOLD_ '(' (widgetAttribute (',' widgetAttribute)* )?
', '? ' )';

container: CONTAINER_ '(' (widgetAttribute (',' widgetAttribute)* )?
', '? ' )';

card: CARD_ '(' (widgetAttribute (',' widgetAttribute)* )? ', '?
')';

textButton : TEXTBUTTON_ '(' (widgetAttribute (',' widgetAttribute)*
)? ', '? ' )';

textField: 'TextField' '(' 'controller' ':' IDENTIFIER ', '
'decoration' ':' 'InputDecoration'
  '(' 'labelText' ':' STRING_LITERAL ', '? ' )' ', '? ('obscureText'
':' constant)? ', '?
  ') ' ', '?;

elevatedButton : ELEVATEDBUTTON_ '(' elevatedButtonAttribute* ', '?
')';
elevatedButtonAttribute: ( onPressed | style | child );

sizedBox : SIZEDBOX_ '(' (widgetAttribute (',' widgetAttribute)* )?
', '? ' )';

alertDialog : ALERTDIALOG_ '(' title ', ' content ', ' actions ', '?
')';

center: CENTER_ '(' (widgetAttribute (',' widgetAttribute)* )? ', '?
')' ;

column: COLUMN_ '(' (widgetAttribute (',' widgetAttribute)* )? ', '?
')';

row: ROW_ '(' (widgetAttribute (',' widgetAttribute)* ', ' )? ', '?
')';

listview : LISTVIEW_ '(' (widgetAttribute (',' widgetAttribute)* )?
', '? ' )';

image : IMAGE_ ':' DECORATIONIMAGE_ '(' AssetImage | networkImage ')'
;

text : ('text' | 'Text') '(' stringInterpolation ', '? ' ) ' ', '? ;

obxWidget: 'Obx' '(' '(' ' )' ' => ' widgetLiteral ', '? ' ) ' ', '? ;
blocBuildWidget : 'BlocBuilder' '<' blocType ', ' blocType '>' '('
blocBuilder ')' ' ', '? ;
blocBuilder : 'builder' ':' '(' CONTEXT_ ', ' ( 'State' | 'state' ) ')'
'{' RETURN_ widgetLiteral ';' ' }' ' ', '? ;

assetImage : ASSETIMAGE_ '(' STRING_LITERAL ' )' ;
networkImage: NETWORKIMAGE_ '(' STRING_LITERAL ' )' ;

widgetAttribute :( appBar | body | onPressed | home |
mainAxisAlignment | style | color | hight | width | padding | margin

```

```

| onTap | actions | theme | content | title | child | children);
appBar : APPBAR_ ':' APPBAR_ '(' title ')' ',' '?' ;
body : 'body' ':' (bodyArguments | blocProviderStatement ) ;
style : 'style' ':' ELEVATEDBUTTON_ DOT 'styleFrom' '(' primarySwatch
')' ',' '?' ;

bodyArguments : (CENTER_ | PADDING_ ) '(' (widgetAttribute (','
widgetAttribute)* )? ',' '?' ')' ;

onPressed : ONPRESSED_ ':' '(' '(' ')' '{' (navigator | statement |
class_call ) '}' | expression ) ',' '?' ;
mainAxisAlignment : MAINAXISALIGNMENT_ ':' MAINAXISALIGNMENT_ DOT
CENTER_ ;
child: CHILD_ class_call ',' '?' ;
children: CHILDREN_ ( class_call | '(' '[' class_call ( ','
class_call )* ',' '?' ']' ) ) ',' '?' ;
onTap: ONTAP_ ':' ONTAP_ ;
color: COLOR_ (HEX_COLOR | expression | boxcolor);
boxcolor : 'state' '==' IDENTIFIER DOT expression '?' 'Colors.' 'red'
':' 'Colors.' 'blue' ',' '?' ;
hight: HIGHT_ ':' '?' NUMBER;
width: WIDTH_ NUMBER;
padding: PADDING_ ':' '?' EDGEINSET_ DOT ( ALL_ '(' NUMBER ')' | CENTER_
) ',' '?' ;
margin: MARGIN_ EDGEINSET_ DOT ( ALL_ '(' NUMBER ')' | CENTER_ ) ;
title: TITLE_ ':' (STRING_LITERAL | ('text' | 'Text') '('
STRING_LITERAL ')' ) ',' '?' ;
content : CONTENT_ ':' ('text' | 'Text') '(' STRING_LITERAL ')' ;
theme : THEME_ ':' THEMEDATA_ '(' primarySwatch ')' ',' '?' ;
primarySwatch : ('primarySwatch' | 'primary' ) ':'
'Colors.' (HEX_COLOR | ( 'red' | 'blue' )) ',' '?' ;
home : HOME_ ':' blocProviderStatement ;
actions : ACTIONS_ ':' ( class_call | '[' class_call ( ','
class_call )* ',' ']' ) ',' '?' ;

blocProviderStatement: 'BlocProvider' ('<' blocType '>')? '('
blocProviderArguments ')' ',' '?' ';?';
blocType: className;
blocProviderArguments: 'create' ':' createStatement ',' child ',' '?' ;
createStatement: '(' ('context' | IDENTIFIER)? ')' '=>' className '('
')' ;

navigator: navigatortype ';' ;
navigatortype: (navigatorPop | navigatorPush |
navigatorPushReplacement) ;
navigatorPop : NAVIGATOR_ '.' 'of' '(' CONTEXT_ ')' '.pop()' ;
navigatorPush : NAVIGATOR_ '.' PUSH_ '(' CONTEXT_ ','
materialPageRoute ',' '?' ')' ;
navigatorPushReplacement : NAVIGATOR_ '.' PUSHREPLACEMENT_ '('
CONTEXT_ ',' materialPageRoute ',' '?' ')' ;

materialPageRoute : MATERIALPAGEROUTE_ '(' BUILDER_ ':' '(' CONTEXT_
')' '=>' className '(' ( IDENTIFIER ':' IDENTIFIER )? ')' ',' '?' ')'
',' '?' ;

mainClass : VOID_ MAIN_ '(' ')' '{' runApp statement* '}' ;
runApp : 'runApp' '(' IDENTIFIER '(' ')' ')' ';' ;
className : IDENTIFIER ;

```

```

function_call : IDENTIFIER '(' ( class_call| expression)? ')' ;

class_call : ((CONST_)? IDENTIFIER '(' classArgument? ')' |
widgetLiteral | blocCall | function_call );
classArgument : (expression | namedArgument | required );

blocCall : 'BlocProvider' DOT 'of' '<' blocType '>' '(' (CONTEXT_ |
blocProviderArguments ) ')' ;

namedArgument
  : IDENTIFIER ':' expression
  ;

required : '{' 'required' 'this' DOT IDENTIFIER '}' ;

statement : ifStatement
  | showDialogStatement
  | enumStatement
  | varDeclaration
  | whileStatement
  | printStatement
  | varAssignment
  | blockStatement
  | methodcallStatement
  | methodStatement
  | setStateStatement
  | navigator
  ;

setStateStatement: SETSTATE_ '(' '(' ')' '{' statement* '}' ')' ';' ;
methodStatement
: OVERRIDE ? type IDENTIFIER '(' (type IDENTIFIER (',' type
IDENTIFIER)*)? ')' '{' statement* '}' ;

showDialogStatement : SHOWDIALOG_ '(' showDialogProperties ',' '?' ')'
';';
showDialogProperties : CONTEXT_ ':' CONTEXT_ ',' showDialogBuilder;
showDialogBuilder : BUILDER_ ':' '(' 'BuildContext' CONTEXT_ ')' '{'
RETURN_ widgetLiteral ';' '}' ;

methodcallStatement: IDENTIFIER '.' (function_call | IDENTIFIER) ';' ;
blockStatement : '{' (statement)* '}' ;
varDeclaration : FINAL_ ? type IDENTIFIER ( '=' ( expression |
class_call ) )? ';' ;
ifStatement : IF_ '(' expression ')' statement (elseCond)? ;
elseCond : ELSE_ statement ;
whileStatement: WHILE_ '(' expression ')' statement* ;
printStatement : PRINT_ '(' expression ')' ';' ;
varAssignment : IDENTIFIER '=' expression ';' ;
enumStatement : 'enum' IDENTIFIER '{' constant (',' constant)* '}' ;

expression
  : '(' expression ')'
#parenExpr
  | '!' expression
#notExpr
  | expression ('*' | '/' | '÷') expression
#multiplication
  | expression ('+' | '-') expression
#addition

```

```

    | expression '<' expression
#lessThan
    | expression '&&' expression
#and
    | expression '==' expression
#equal
    | expression '!=' expression
#notequal
    | expression '[' expression ']'
#indexArray
    | expression '.' ('length' | 'text' | IDENTIFIER)
#propertyAccess
    | expression '(' (expression (',' expression)*)? ')'
#methodCall
    | expression '?' expression ':' expression
#ternary
    | constant
#constantVal
    | objectAccess (IDENTIFIER|constant)
#fieldAccess
    | IDENTIFIER
#idCall
    | NEW_ INT '[' expression ']'
#newArray
    | NEW_ className '(' ' ' ')'
#newClass
    | '{' expression '}'
#exprBlock
    ;

objectAccess: (IDENTIFIER '.')* ;

type : INT '[' ']'
    | VAR
    | BOOLEAN
    | INT
    | FLOAT
    | VOID_
    | FINAL_
    | className
    ;

constant : colorName | TRUE | FALSE | NULL | NUMBER | STRING_LITERAL;

colorName : ( 'red' | 'blue' ) ;

IMPORT_:'import';
EXPORT_:'export';
WIDGET_:'Widget';
GESTUREDETECTOR_:'GestureDetector';
STATELESSWIDGET_:'StatelessWidget';
STATEFULWIDGET_:'StatefulWidget';
BUILD_:'build';
BUILDCONTEXT_:'BuildContext';
CONTEXT_:'context';

```



```

CHILD_:'child:';
CHILDREN_:'children:';
CARD_:'card';
CENTER_:( 'center' | 'Center' );
COLUMN_:'column' | 'Column';
ROW_:( 'Row' | 'row' );
MaterialApp_:'MaterialApp';
FINAL_:'final';
LISTVIEW_:'listview';
IMAGE_:'Image';
DECORATIONIMAGE_:'DecorationImage';
ASSETIMAGE_:'Image:AssetImage';
NETWORKIMAGE_:'Image:NetworkImage';
ONTAP_:'onTap';
ONPRESSED_:'onPressed';
COLOR_:'color:';
HIGHT_:( 'height:' | 'hight:' );
WIDTH_:'width:';
PADDING_:( 'padding' | 'Padding' );
EDGEINSET_:( 'EdgeInset' | 'EdgeInsets' );
ALL_:'all';
MARGIN_:'margin';
TITLE_:'title';
THEME_:'theme';
THEMEDATA_:'ThemeData';
HOME_:'home';
IF_:'if';
ELSE_:'else';
WHILE_:'while';
PRINT_:'print';
ACTIONS_:'actions';
DOT_:'.';
CONTAINER_:'Container';
OVERRIDE_:'@override';
EXTANDS_:'extends';
RETURN_:'return';
NEW_:'new';
PUBLIC_:'public';
CLASS_:'class';
MAIN_:'main';
VOID_:'void';
FLOAT_:'float';
INT_:'int';
VAR_:'var';
BOOLEAN_:( 'boolean' | 'bool' );
FLOAT_LITERAL_:'g';
NULL_:( 'null' | 'NULL' );
TRUE_:'true';
FALSE_:'false';
CONST_:'const';
CREATESTATE_:'createState';
SETSTATE_:'setState';
NAVIGATOR_:'Navigator';
PUSH_:'push';
MATERIALPAGEROUTE_:'MaterialPageRoute';
BUILDER_:'builder';
PUSHREPLACEMENT_:'pushReplacement';
TEXTBUTTON_:'TextButton';
SHOWDIALOG_:'showDialog';
ALERTDIALOG_:'AlertDialog';
CONTENT_:'content';

```

```

SCAFFOLD_ : 'Scaffold';
APPBAR_ : ('appBar' | 'AppBar');
MAINAXISALIGNMENT_ : ('mainAxisAlignment' | 'MainAxisAlignment');
SIZEDBOX_ : 'SizedBox';
ELEVATEDBUTTON_ : 'ElevatedButton';
GETXCONTROLLER_ : 'GetXController';
//COLOR_NAME: ( 'red' | 'blue' );

stringInterpolation
: STRING_LITERAL
| STRING_LITERAL interpolationPart
;

interpolationPart
: '$' IDENTIFIER
;

STRING_LITERAL
: '"' (ESC_SEQ | ~('\\"|'"'))* '"'
| '\'' (ESC_SEQ | ~('\\"|'"'))* '\''
;

fragment
ESC_SEQ
: '\\\"'
| '\\\\'
// | '\\/'
| '\\b'
| '\\f'
| '\\n'
| '\\r'
| '\\t'
;

HEX_COLOR: '#' (HEX_DIGIT HEX_DIGIT);
NUMBER: DIGIT+ ('.' DIGIT+)?;
fragment DIGIT: '0'..'9';
fragment HEX_DIGIT: DIGIT | 'a'..'f' | 'A'..'F';
IDENTIFIER
: '_'? [a-zA-Z_] [a-zA-Z0-9_]*
;

WS: [ \t\r\n]+ -> skip;
LINE_COMMENT : '//' ~('\n' | '\r')* -> skip;
//MULTI_COMMENT : '/*' .* '*/' -> skip;

```

```

appBar:AppBar BRACOPEN title BRACCLOSE COMA;
title:Title COLUMN tExt ;
tExt:Text BRACOPEN (STRING[CHAR] COMA style* BRACCLOSE COMA;
style:STYLE COLUMN tExtStyle COMA;
tExtStyle:TEXTSTYLE BRACOPEN fontSize BRACCLOSE;
fontSize:FONTSIZE COLUMN NUMBER COMA;
cOntainer:CONTAINER BRACOPEN (width|height|margin|padding|decoration)* child? BRACCLOSE COMA;
child:CHILD COLUMN widgets;
width:WIDTH COLUMN (NUMBER|NAME (DOT NAME)*) COMA;
height:HEIGHT COLUMN NUMBER COMA;
margin:MARGIN COLUMN EdgeInsets DOT ALL BRACOPEN NUMBER BRACCLOSE COMA;
padding:PADDING COLUMN EdgeInsets DOT ALL BRACOPEN NUMBER BRACCLOSE COMA;
decoration:DECORATION COLUMN BoxDecoration COMA;
bOxDecoration:BoxDecoration BRACOPEN border? borderRadius? BRACCLOSE;
border:BORDER COLUMN Border DOT ALL BRACOPEN (NUMBER|width|height)* BRACCLOSE COMA;
borderRadius:BORDERRADIUS COLUMN BorderRadius DOT Circular BRACOPEN NUMBER BRACCLOSE COMA;
rOw:ROW BRACOPEN children mainAxisSize? BRACCLOSE COMA;
children:CHILDREN COLUMN SEQOPEN widgets* SEQCLOSE COMA;
mainAxisSize:MainAxisSize COLUMN MainAxisSize DOT (MIN|MAX) COMA;
eXpanded:EXPANDED BRACOPEN child BRACCLOSE COMA;
cOLUMN:COLUMN BRACOPEN mainAxisSize? children BRACCLOSE COMA;
cArD:CARD BRACOPEN child style? BRACCLOSE COMA;
lIstView:ListView BRACOPEN itemBuilder separatorBuilder itemCount BRACCLOSE COMA;
iTemBuilder:ItemBuilder COLUMN BRACOPEN CONTEXT COMA NAME BRACCLOSE (CURLYOPEN body_element* CURLYCLOSE|ATTRIBUTION cull_fun COMA);
separatorBuilder:SeparatorBuilder COLUMN BRACOPEN CONTEXT COMA NAME BRACCLOSE (CURLYOPEN body_element* CURLYCLOSE|ATTRIBUTION widgets);
iTemCount:ItemCount COLUMN NAME (DOT NAME)* COMA;
iMage:IMAGE BRACOPEN image fit? BRACCLOSE COMA;

```

بنى المعطيات الخاصة بجدول الرموز:

بالنسبة widget:

تمّ بناء جدول يحوي كل widget بحيث كل widget تحوي على معلوماتها وكل widget تحوي على التاغ الخاص بها

بالنسبة widget attribute :

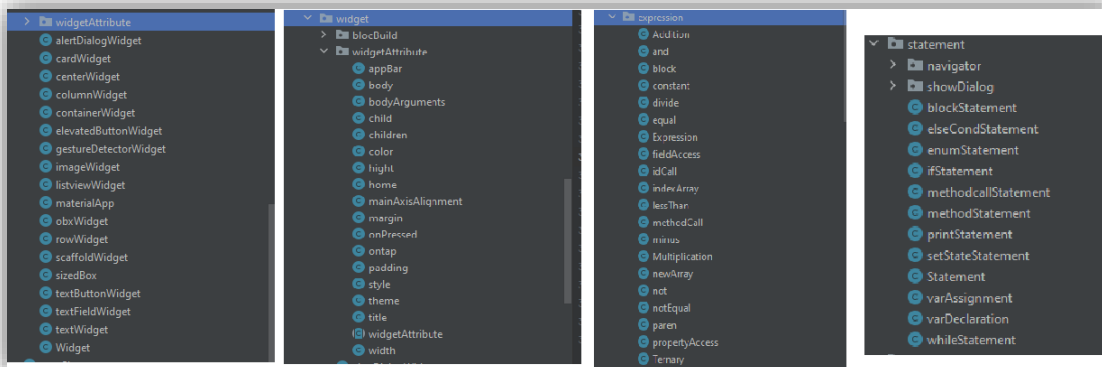
تمّ بناء جدول يحوي كل widget attribute بحيث كل widget attribute تحوي على معلوماتها

بالنسبة statements :

تمّ بناء جدول يحوي كل : statements بحيث كل : statements تحوي على معلوماتها

بالنسبة expression :

تمّ بناء جدول يحوي كل expression بحيث كل expression تحوي على معلوماتها



❖ بنى المعطيات الخاصة بالشجرة المولدة :

بالنسبة widget:

alertDialogWidget

materialApp

obxWidget

rowWidget

scaffoldWidget

sizedBox

textButtonWidget

textFieldWidget

textWidget

cardWidget

centerWidget

columnWidget

containerWidget

elevatedButtonWidget

gestureDetectorWidget

imageWidget

listviewWidget

وكلهم ييرثوا من الكلاس Widget

بالنسبة widget attribute :

bodyArguments

title

style

home

child
children
onPressed
margin
body
theme
color
ontap
hight
width
padding
mainAxisAlignment
appBar

وجميعهم يرثون من الكلاس widgetAttribute

بالنسبة statements :

whileStatement
varAssignment
printStatement
ifStatement
varDeclaration
elseCondStatement
methodcallStatement
setStateStatement
methodStatement
enumStatement
blockStatement

وجميعهم يرثون من الكلاس statement

بالنسبة expression :

equal

paren

divide

Multiplication

idCall

not

fieldAccess

Addition

block

Than less

methodCall

Ternary

indexArray

constant

and

minus

propertyAccess

notEqual

newArray

وجميعهم يرثون من الكلاس expression

❖ بنية اللغة الهدف:

تمّ معالجة التعليمات التالية :

style – header – head - title - body - script –Html ✓

❖ خوارزمية التحويل إلى اللغة الهدف :

ضمن جذر شجرة ال ast يتم انشاء ال tags لملف ال html ضمن تابع يسمى

generateCode

يذكر مرة واحدة فقط عند الجذر

ولدى انشاء ال tags ولكي لا يتم فتح tag واغلاق tag عند عملية توليد الكود

تم تقسيم كود التوليد عند جذر الشجرة بحيث

ضمن head tag

نقوم بإنشاء ال style tag ونضمنه نقوم باستدعاء تابع لتوليد كود css

generateCSScode

وبعد فتح body tag نقوم باستدعاء تابع توليد ال html

generateHTMLcode

وبعد انتهاء توليد كود ال html نقوم بفتح script tag لنقوم بوضع كود

الجافاسكريبت فيه عبر استدعاء تابع generateJScode بداخله

وهكذا نكون فصلنا اللوجك عن الرسم ولكل كلاس ضمن الشجرة اصبح لديه 3

توابع لتوليد الكود

```
Usage
public String generateCode() {
    StringBuilder stringBuilder = new StringBuilder();

    stringBuilder.append("<!DOCTYPE html>\n");
    stringBuilder.append("<html>\n");
    stringBuilder.append("<head>\n");
    stringBuilder.append("<style>\n");

    for (int i = 0; i < widgetClasses.size(); i++) {
        widgetClass widgetClass = widgetClasses.get(i);
        if (widgetClass != null) {
            stringBuilder.append(widgetClass.generateCSSCode());
        }
    }

    for (int i = 0; i < stateClasses.size(); i++) {
        stateClass stateClass = stateClasses.get(i);
        if (stateClass != null) {
            stringBuilder.append(stateClass.generateCSSCode());
        }
    }
}
```

```
stringBuilder.append("</style>\n");
stringBuilder.append("</head>\n");

stringBuilder.append("<body>\n");

for (int i = 0; i < widgetClasses.size(); i++) {
    widgetClass widgetClass = widgetClasses.get(i);
    if (widgetClass != null) {
        stringBuilder.append(widgetClass.generateHTMLCode());
    }
}

for (int i = 0; i < stateClasses.size(); i++) {
    stateClass stateClass = stateClasses.get(i);
    if (stateClass != null) {
        stringBuilder.append(stateClass.generateHTMLCode());
    }
}

for (int i = 0; i < statements.size(); i++) {
    statement statement = statements.get(i);
    if (statement != null) {
        stringBuilder.append(statement.generateJScode());
    }
}
```

```
stringBuilder.append("\n<script>");

for (int i = 0; i < widgetClasses.size(); i++) {
    widgetClass widgetClass = widgetClasses.get(i);
    if (widgetClass != null) {
        stringBuilder.append(widgetClass.generateJScode());
    }
}

for (int i = 0; i < stateClasses.size(); i++) {
    stateClass stateClass = stateClasses.get(i);
    if (stateClass != null) {
        stringBuilder.append(stateClass.generateJScode());
    }
}

for (int i = 0; i < statements.size(); i++) {
    Statement statement = statements.get(i);
    if (statement != null) {
        stringBuilder.append(statement.generateJScode());
    }
}
```


❖ تقسيم العمل :

بداية قمنا بتعديل قواعد اللغة التي وضعناها بالفصل الاول لكي تتضمن التعديلات التي طلبها الدكتور مثل (statemangment , navigator)
قام بعمل هذه الخطوة **مصطفى حرب و صبا طليعة**

ثم قمنا بتحليل كيف يجب للكود ان يكون عليه بعد التوليد وايجاد كل كومبونانت في الفلاتر مايقابله بالويب
قام بهذه الخطوة **ليلى العودة**

قمنا بإنشاء الكلاسات المسؤالة عن القواعد الجديدة المستحدثة في الفصل الحالي بالاضافة لتوابع توليد الكود
قام بتنفيذ الخطوة **أحمد المبارك**

ثم بعد ذلك قمنا بإنشاء كلاس لاكتشاف الاخطاء في الكود وضمنه توابع لايجاد كل خطأ على حدى
قام بالتنفيذ **احمد المبارك ومصطفى حرب**

قام بكتابة التقرير والعرض التقديمي **ليلى العودة و صبا طليعة**



Print ("The End");
buf ("LPG EUQ");