

# بنام خدا

## فاز اول پروژه کامپایلر

علی دیندامل ۹۷۷۳۱۱۸

کلمات کلیدی:

{class, while, if, else, main, INT, STR, integer, bool, True, False, string, or, and, not, input, output}

عملگرها:

{+, -, =, >, <, ==}

جدا کننده ها :

{:, {, }, (, ), "}

توابع از پیش تعریف شده:

{input, output, STR, INT}

توضیحات:

این زبان سه نوع تایپ دارد که شامل bool و integer و string میشود. یه کلاس بنام main وجود دارد که شروع اجرای برنامه و تمام مراحل برنامه در آن نوشته میشود. توابع از پیش آماده شده ی STR و INT برای تبدیل نوع تعریف شده اند و توابع input و output هم برای ورودی و خروجی استاندارد استفاده میشوند. زبان شامل یکسری اعمال ریاضی و منطقی هم میشود.

مثال هایی از برنامه:

مثلا اول :

```
class main{
    integer a = 10;
    while(10 == a){
        output(a);
    }
}
```

مثلا دوم :

```
class main{
    integer num = INT(input());
    if(5 < num){
        output("input is : ");
        output(num);
    }
}
```

مثلا سوم :

```
class main{
    string s = input();
    if(s == "YES"){
        output("yes");
    }
    else{
        output("no");
    }
}
```

مثلا چهارم :

```
class main{
    string a_string = input();
    integer b_integer = INT(input());
    integer a_integer = INT(a_string);
    string b_string = STR(b_integer);
    integer c = a_integer + 2;
    string d = b_string + "2";

    output(c);
    output(d);
}
```

\* برای هر دو مرحله ی پروژه از کتاب خانه ی ply برای python استفاده شده.  
\* کتاب خانه ی ply بر اساس LR 1 است.

گرامر های استفاده شده برای قسمت پارسر:

```
class : CLASS MAIN LCBRACK statements RCBRACK
statements : statement
           | statements statement
statement : assign_s SEMICOLON
           | output SEMICOLON
           | if_s
           | input SEMICOLON
           | type_cast SEMICOLON
           | while_s
while_s : WHILE LPAREN logic_o RPAREN LCBRACK statement RCBRACK
        | WHILE LPAREN comparison RPAREN LCBRACK statement RCBRACK
if_s : IF LPAREN logic_o RPAREN LCBRACK statements RCBRACK ELSE LCBRACK
statements RCBRACK
      | IF LPAREN logic_o RPAREN LCBRACK statements RCBRACK
```

```

    | IF LPAREN comparison RPAREN LCBRACK statements RCBRACK ELSE
LCBRACK statements RCBRACK
    | IF LPAREN comparison RPAREN LCBRACK statements RCBRACK
assign_s : INTEGER_TYPE ID ASSIGN arithmetic_o
    | BOOL_TYPE ID ASSIGN logic_o
    | STRING_TYPE ID ASSIGN string_o
comparison : arithmetic_o GTS arithmetic_o
    | arithmetic_o LTS arithmetic_o
    | arithmetic_o EQ arithmetic_o
    | string_o EQ string_o
string_o : string_o PLUS z
    | z
    z : variable
    | STRING
logic_o : logic_o AND m
    | m
    m : m OR n
    | n
    n : LPAREN logic_o RPAREN
    | comparison
    | NOT LPAREN logic_o RPAREN
    | variable
    | BOOL
arithmetic_o : arithmetic_o PLUS f
    | arithmetic_o MINUS f
    | f
    f : LPAREN arithmetic_o RPAREN
    | INTEGER
    | variable
variable : ID
input : INTEGER_TYPE ID ASSIGN INPUT LPAREN RPAREN
    | STRING_TYPE ID ASSIGN INPUT LPAREN RPAREN
    | BOOL_TYPE ID ASSIGN INPUT LPAREN RPAREN
output : OUTPUT LPAREN ID RPAREN
    | OUTPUT LPAREN STRING RPAREN
    | OUTPUT LPAREN BOOL RPAREN
    | OUTPUT LPAREN INTEGER RPAREN
type_cast : INTEGER_TYPE ID ASSIGN INT LPAREN INPUT LPAREN RPAREN
RPAREN
    | INTEGER_TYPE ID ASSIGN INT LPAREN ID RPAREN

```

| STRING\_TYPE ID ASSIGN STR LPAREN INPUT LPAREN RPAREN RPAREN  
| STRING\_TYPE ID ASSIGN STR LPAREN ID RPAREN

first ها و follow های گرامر:

first(class) = {CLASS}  
first(statements) = {INTEGER\_TYPE, BOOL\_TYPE, STRING\_TYPE, OUTPUT, IF, WHILE}  
first(statement) = {INTEGER\_TYPE, BOOL\_TYPE, STRING\_TYPE, OUTPUT, IF, WHILE}  
first(while\_s) = {WHILE}  
first(if\_s) = {IF}  
first(assign\_s) = {INTEGER\_TYPE, BOOL\_TYPE, STRING\_TYPE}  
first(comparison) = {LPAREN, INTEGER, STRING, ID}  
first(string\_o) = {STRING, ID}  
first(logic\_o) = {LPAREN, NOT, BOOL, ID, INTEGER, STRING}  
first(arithmetic\_o) = {LPAREN, INTEGER, ID}  
first(variable) = {ID}  
first(input) = {INTEGER\_TYPE, STRING\_TYPE, BOOL\_TYPE}  
first(output) = {OUTPUT}  
first(type\_cast) = {INTEGER\_TYPE, STRING\_TYPE}

follow(class) = {\$}  
follow(statements) =  
{RCBRACK, INTEGER\_TYPE, BOOL\_TYPE, STRING\_TYPE, OUTPUT, IF, WHILE}  
follow(statement) =  
{RCBRACK, INTEGER\_TYPE, BOOL\_TYPE, STRING\_TYPE, OUTPUT, IF, WHILE}  
follow(while\_s) =  
{RCBRACK, INTEGER\_TYPE, BOOL\_TYPE, STRING\_TYPE, OUTPUT, IF, WHILE}  
follow(if\_s) =  
{RCBRACK, INTEGER\_TYPE, BOOL\_TYPE, STRING\_TYPE, OUTPUT, IF, WHILE}  
follow(assign\_s) = {SEMICOLON}  
follow(comparison) = {RPAREN, SEMICOLON, AND, OR}  
follow(string\_o) = {SEMICOLON, EQ, RPAREN, PLUS, AND, OR}  
follow(logic\_o) = {RPAREN, SEMICOLON, AND}  
follow(arithmetic\_o) = {SEMICOLON, GTS, RPAREN, LTS, EQ, PLUS, MINUS, AND, OR}  
follow(variable) = {SEMICOLON, EQ, RPAREN, PLUS, AND, OR, GTS, LTS, MINUS}  
follow(input) = {SEMICOLON}  
follow(output) = {SEMICOLON}  
follow(type\_cast) = {SEMICOLON}