



Grupa 1

Kreirati **jflex** specifikaciju za generisanje leksičkog analizatora jezika koji je definisan sledećom gramatikom:

Program → **main** *Block* **exit**

Block → *Declarations* *Expressions*

Declarations → *Declarations* *Declaration* | *Declaration*

Declaration → **Type** **ID** ;

Type → **int** | **float** | **bool**

Expressions → *Expressions* ; *Expression* | *Expression*

Expression → *Assignment* | *ApplyExpression*

Assignment → **ID** := *ArithmeticExpression*

ArithmeticExpression → *ArithmeticExpression* + *TermExpression*
| *ArithmeticExpression* – *TermExpression*
| *TermExpression*

TermExpression → **CONST** | **ID**

ApplyExpression → **for** **ID** **in** [*NameList*] **apply** *Expression*

NameList → *NameList* , **ID** | **ID**

Terminalni simbol **ID** u ovom programskom jeziku označava identifikator (niz slova i cifara u kojem prvi znak ne može da bude cifra), a simbol **CONST** konstantu koja može da bude zadata u jednom od sledećih formata:

1. Konstante tipa **int**:

[<oznaka_osnove>]<niz_cifara_zadate_osnove>

Pri čemu oznaka osnove može biti

0 – za brojni sistem sa osnovom 8,

0x – za brojni sistem sa osnovom 16,

Ukoliko je oznaka osnove izostavljena, podrazumeva se osnova 10.

2. Konstante tipa **float**:

0. <niz_cifara>[E[±]<niz_cifara>]

3. Konstante tipa **bool**:

true ili **false**

Komentari u ovom programskom jeziku počinju simbolom **//** i završavaju se prelazom na novi red.



Grupa 2

Kreirati **jflex** specifikaciju za generisanje leksičkog analizatora jezika koji je definisan sledećom gramatikom:

Program → **main** () *Block*
Block → { *Variables* *Statements* }
Variables → *Variables* *Variable* | *Variable*
Variable → *Type* **ID** ;
Type → **int** | **real** | **boolean**
Statements → *Statements* ; *Statement* | *Statement*
Statement → *Assignment* | *IfStatement*
IfStatement → **if** (*RelExpression*) : *Block* *ElsePart*
ElsePart → *ElifList* *ElseStatement* | *ElifList* | *ElseStatement* | ε
ElifList → *ElifList* *Elif* | *Elif*
Elif → **elif** (*RelExpression*) : *Block*
ElseStatement → **else** : *Block*
RelExpression → *Term* *RelOp* *Term* | *Term*
Term → **ID** | **CONST**
RelOp → < | <= | == | <> | > | >=
Assignment → **ID** := *Term*

Terminalni simbol **ID** u ovom programskom jeziku označava identifikator (niz slova i cifara u kojem prvi znak ne može da bude cifra), a simbol **CONST** konstantu koja može da bude zadata u jednom od sledećih formata:

1. Konstante tipa **int**:

[<osnova>]#<niz_cifara_zadate_osnove>

Pri čemu osnova može da bude u opsegu 2-16. Ukoliko je osnova izostavljena podrazumeva se 16, a ukoliko je i simbol '#' izostavljen, podrazumeva se osnova 10.

2. Konstante tipa **real**:

<niz_cifara>.[<niz_cifara>][E[±]<niz_cifara>]

3. Konstante tipa **boolean**:

true ili **false**

Komentari u ovom programskom jeziku počinju simbolom **/**** i završavaju se simbolom ***/**.



Grupa 3

Kreirati **jflex** specifikaciju za generisanje leksičkog analizatora jezika koji je definisan sledećom gramatikom:

Program → **program** *Block* **return**

Block → **begin** *VarList* *Statements* **end**

VarList → *VarList* *Var* | *Var*

Var → **ID** : *Type* ;

Type → **integer** | **char** | **string** | **file**

Statements → *Statements* ; *Statement* | *Statement*

Statement → *Assignment* | *ReadExpression*

Assignment → **ID** = *Expression*

Expression → *ArithmeticExpression* | **open** (*PrimaryExpression*)

ArithmeticExpression → *ArithmeticExpression* + *PrimaryExpression*

| *ArithmeticExpression* – *PrimaryExpression*

| *PrimaryExpression*

PrimaryExpression → **ID** | **CONST**

ReadExpression → **read** (**ID** in **ID**) **do** *Block*

Terminalni simbol **ID** u ovom programskom jeziku označava identifikator (niz slova, donjih crta i cifara u kojem prvi znak ne može da bude cifra), a simbol **CONST** konstantu koja može da bude zadata u jednom od sledećih formata:

1. Konstante tipa **integer**:

[<oznaka_osnove>]<niz_cifara_zadate_osnove>

Pri čemu oznaka osnove može biti

0 – za brojni sistem sa osnovom 8,

0x – za brojni sistem sa osnovom 16,

Ukoliko je oznaka osnove izostavljena, podrazumeva se osnova 10.

2. Konstante tipa **char**:

'<znak>'

3. Konstante tipa **string**:

"<niz_znakova>"

Komentari u ovom programskom jeziku počinju simbolom (***** i završavaju se simbolom *****).



Grupa 4

Kreirati **jflex** specifikaciju za generisanje leksičkog analizatora jezika koji je definisan sledećom gramatikom:

Program → **program** *Block* **end**

Block → { *Declarations* *Statements* }

Declarations → *Declarations* *Declaration* | *Declaration*

Declaration → *VariableDeclaration* | *FunctionDeclaration*

VariableDeclaration → **ID** : *Type* ;

Type → **int** | **float** | **char**

FunctionDeclaration → **ID** (*Parameters*) => *Expression* ;

Parameters → *Parameters* , *Parameter* | *Parameter*

Parameter → **ID** : *Type* | **ID** : *Type* = **CONST**

Statements → *Statements* ; *Assignment* | *Assignment*

Assignment → **ID** = *Expression*

Expression → **CONST** | **ID** | *FunctionCall*

FunctionCall → **ID** (*ArgumentsList*)

ArgumentsList → *ArgumentsList* , *Expression* | *Expression*

Terminalni simbol **ID** u ovom programskom jeziku označava identifikator (niz slova i cifara u kojem prvi znak ne može da bude cifra), a simbol **CONST** konstantu koja može da bude zadata u jednom od sledećih formata:

1. Konstante tipa **int**:

[<oznaka_osnove>]<niz_cifara_zadate_osnove>

Pri čemu oznaka osnove može biti

0 – za brojni sistem sa osnovom 8,

0x – za brojni sistem sa osnovom 16,

Ukoliko je oznaka osnove izostavljena, podrazumeva se osnova 10.

2. Konstante tipa **float**:

<niz_cifara>.[<niz_cifara>][E[±]<niz_cifara>] ili

.<niz_cifara>[E[±]<niz_cifara>]

3. Konstante tipa **char**:

'<znak>'

Komentari u ovom programskom jeziku počinju simbolom -- i završavaju se simbolom --.



Grupa 5

Kreirati **JFlex** specifikaciju za generisanje leksičkog analizatora jezika koji je definisan sledećom gramatikom:

Program → **main () Block**

Block → { *Declarations StatementList* }

Declarations → *Declarations VarDecl* | ε

VarDecl → *Type NameList* ;

NameList → **ID** | *NameList , ID*

Type → **int** | **char** | **float**

StatementList → *StatementList Statement* | *Statement*

Statement → *CaseStatement* | **ID = Expression ;** | *Block*

CaseStatement → **case (Expression) { WhenStatementList }**

WhenStatementList → *WhenStatementList WhenStatement* | *WhenStatement*

WhenStatement → **when CONST : Statement**

Expression → *Expression AddOperator Term* | *Term*

AddOperator → **+** | **-**

Term → **ID** | **CONST** | (*Expression*)

Terminalni simbol **ID** u ovom programskom jeziku označava identifikator (niz slova, cifara i znaka **_** "donja crta" u kojem prvi znak ne može da bude cifra), a simbol **CONST** konstantu koja može da bude zadata u jednom od sledećih formata:

1. Konstante tipa **int**:

[<oznaka_osnove>]<niz_cifara_zadate_osnove>

Pri čemu oznaka osnove može biti

Oct – za brojni sistem sa osnovom 8,

Hex – za brojni sistem sa osnovom 16,

Dec – za brojni sistem sa osnovom 10,

Ukoliko je oznaka osnove izostavljena, podrazumeva se osnova 10.

2. Konstante tipa **float**:

<niz_cifara>.[<niz_cifara>][E[±]<niz_cifara>]

3. Konstante tipa **char**:

'<znak>'

Komentari u ovom programskom jeziku počinju simbolom **%%** i završavaju se simbolom **%%**.



Grupa 6

Kreirati **JFlex** specifikaciju za generisanje leksičkog analizatora jezika koji je definisan sledećom gramatikom:

Program → **program** *Block* .

Block → **begin** *Variables StatementList* **end**

Variables → *Variables Declaration* | ϵ

Declaration → *NameList* : *Type* ;

NameList → *NameList* , **ID** | **ID**

Type → **integer** | **char** | **real** | **boolean**

StatementList → *Statement* | *StatementList* *Statement*

Statement → *WhileLoop* | **ID** := *Expression* ; | *Block*

WhileLoop → **while** *Expression* : *Statement* **else** *Statement*

Expression → *Expression* **or** *AndExpression* | *AndExpression*

AndExpression → *AndExpression* **and** *RelExpression* | *RelExpression*

RelExpression → *Term* *RelOp* *Term* | *Term*

RelOp → < | <= | == | <> | > | >=

Term → **ID** | **CONST** | (*Expression*)

Terminalni simbol ID u ovom programskom jeziku označava identifikator (niz slova, cifara i znaka \$ u kojem prvi znak ne može da bude cifra), a simbol CONST konstantu koja može da bude zadata u jednom od sledećih formata:

1. Konstante tipa **int**:

[<oznaka_osnove>]<niz_cifara_zadate_osnove>

Pri čemu oznaka osnove može biti

0 – za brojni sistem sa osnovom 8,

0x – za brojni sistem sa osnovom 16,

Ukoliko je oznaka osnove izostavljena, podrazumeva se osnova 10.

2. Konstante tipa **real**:

<niz_cifara>.[<niz_cifara>][E[±]<niz_cifara>] i

.<niz_cifara>[E[±]<niz_cifara>]

3. Konstante tipa **char**:

'<znak>'

4. Konstante tipa **boolean**:

true i **false**

Komentari u ovom programskom jeziku počinju simbolom ** i završavaju se simbolom **.



Grupa 7

Kreirati **JFlex** specifikaciju za generisanje leksičkog analizatora jezika koji je definisan sledećom gramatikom:

Program → **main** () *Block*

Block → { *Declarations* *StatementList* }

Declarations → *Declarations* *VarDecl* | ε

VarDecl → *Type* *NameList* ;

NameList → **ID** | *NameList* , **ID**

Type → **int** | **char** | **float** | **bool**

StatementList → *StatementList* *Statement* | *Statement*

Statement → *RedoLoop* | **ID** = *Expression* ; | *Block*

RedoLoop → **loop** (*Expression*) { *Statement* **redo** (*Expression*) ; *Statement* }

Expression → *Expression* || *AndExpression* | *AndExpression*

AndExpression → *AndExpression* && *RelExpression* | *RelExpression*

RelExpression → *Term* *RelOp* *Term* | *Term*

RelOp → < | <= | == | != | > | >=

Term → **ID** | **CONST** | (*Expression*)

Terminalni simbol **ID** u ovom programskom jeziku označava identifikator (niz slova, cifara i znaka _ "donja crta" u kojem prvi znak ne može da bude cifra), a simbol **CONST** konstantu koja može da bude zadata u jednom od sledećih formata:

1. Konstante tipa **int**:

[<oznaka_osnove>]<niz_cifara_zadate_osnove>

Pri čemu oznaka osnove može biti

0#o – za brojni sistem sa osnovom 8,

0#x – za brojni sistem sa osnovom 16,

0#d – za brojni sistem sa osnovom 10,

Ukoliko je oznaka osnove izostavljena, podrazumeva se osnova 10.

2. Konstante tipa **float**:

0.[<niz_cifara>][E[±]<niz_cifara>]

3. Konstante tipa **char**:

'<znak>'

4. Konstante tipa **bool**:

true i **false**

Komentari u ovom programskom jeziku počinju simbolom % i završavaju se simbolom %.



Grupa 8

Kreirati **JFlex** specifikaciju za generisanje leksičkog analizatora jezika koji je definisan sledećom gramatikom:

Program → **program** *Block* .

Block → **begin** *Variables StatementList* **end**

Variables → *Variables Declaration* | ϵ

Declaration → *NameList* : *Type* ;

NameList → *NameList* , **ID** | **ID**

Type → **integer** | **char** | **real** | **boolean**

StatementList → *Statement* | *StatementList* *Statement*

Statement → *SelectStatement* | **ID** := *Expression* ; | *Block*

SelectStatement → **select begin** *CaseList* **end**

CaseList → *CaseList* *Case* | *Case*

Case → **case** *Expression* => *Statement*

Expression → *Expression* **or** *AndExpression* | *AndExpression*

AndExpression → *AndExpression* **and** *RelExpression* | *RelExpression*

RelExpression → *Term* *RelOp* *Term* | *Term*

RelOp → < | <= | == | <> | > | >=

Term → **ID** | **CONST** | (*Expression*)

Terminalni simbol **ID** u ovom programskom jeziku označava identifikator (niz slova, cifara i znaka **\$** u kojem prvi znak ne može da bude cifra), a simbol **CONST** konstantu koja može da bude zadata u jednom od sledećih formata:

1. Konstante tipa **int**: [*<oznaka_osnove>*]*<niz_cifara_zadate_osnove>*

Pri čemu oznaka osnove može biti

0 – za brojni sistem sa osnovom 8,

0x – za brojni sistem sa osnovom 16,

Ukoliko je oznaka osnove izostavljena, podrazumeva se osnova 10.

2. Konstante tipa **real**:

<niz_cifara> . [*<niz_cifara>*] [*E* [*±*] *<niz_cifara>*] i

. *<niz_cifara>* [*E* [*±*] *<niz_cifara>*]

3. Konstante tipa **char**: '*<znak>*'

4. Konstante tipa **boolean**:

true i **false**

Komentari u ovom programskom jeziku počinju simbolom **|*** i završavaju se simbolom ***|**.