



KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ INŽINERIJOS KATEDRA

P175B124 PROGRAMAVIMO KALBŲ TEORIJA

## **KALBOS PAVADINIMAS:**

**FL# (FRIENDS-LANGUAGE-SHARP)**

**Grupė: Draugai Trys**

**Laboratorinių darbų dėstytojas:**

lekt. Guogis Evaldas  
doc. Sajavičius Svajūnas

**Studentai:**

Aleksas Juzukonis IFF-1/8  
Erikas Briauka IFF-1/8  
Matas Palujanskas IFF-1/8

Kaunas, 2023

# Turinys

1. Užduotis.....	3
2. Kalbos idėja ir pavadinimas.....	3
3. Esminės kalbos savybės.....	3
4. Kūrimo priemonės.....	3
5. Apribojimai.....	4
6. Galutinė kalbos gramatika.....	4
7. Kaip sukompiliuoti ir paleisti.....	6
8. Pavyzdinis kodas ir gauti rezultatai.....	7
9. Naudoti pavyzdžiai, šaltiniai.....	12

# 1. Užduotis

## 1.1. Užduoties tikslas: sukurti programavimo kalbą draugams.

- 1.1.1. Unikali savybė: sukurta bent viena unikali savybė;
- 1.1.2. Struktūra: funkcijos, eilė, kintamieji, išimčių valdymas;
- 1.1.3. Tipai: realizuoti keli tipai(pvz: int, string);
- 1.1.4. Skaitymas/rašymas į failą.

# 2. Kalbos idėja ir pavadinimas

FL# (FRIENDS-LANGUAGE-SHARP) - tai programavimo kalba skirta draugams. Kalbos pavadinimas kilo iš mūsų grupės pavadinimo. Tai grupinis darbas, o mes esame draugai, todėl ir kalba yra draugų.

# 3. Esminės kalbos savybės

Unikali savybė - papildomas kintamųjų tipas "operator". Su tipu "operator" galima priskirti kintamajam operatorių. Pvz:

```
operator a = |*|;  
5 a 6 = 30  
operator b = |+|;  
5 b 5 = 10
```

- Palaikomi duomenų tipai: string, int, boolean, double.
- Palaikomos struktūros: queue.
- Kalbos savybės: išimčių valdymas.
- Palaikomos konstrukcijos: if(), for(), while().
- Skaitymas iš failo, rašymas į failą.

# 4. Kūrimo priemonės

- Java, nes šia kalba yra sukurta daug pavyzdinio kodo, nuo kurio bus patogu "atsispirti" ir kurti savąjį;
- IntelliJ IDEA, nes jau esame susipažinę su šita aplinka, todėl bus patogu ir paprasta;
- Komandos komunikacijai naudojamas discord serveris.

## 5. Apribojimai

Išimčių valdymo metu programuotojas, negali pasirinkti kokia išimtis yra išspausdinama:

```
int a = 6;
try {
    a = a / 0;
}
catch {
    a = a + 2;
}
print(a);
```

```
<PROGRAM OUTPUT>
java.lang.ArithmeticException: / by zero
8
```

Skaitant iš failo galima nuskaityti tik vieną eilutę.  
Išimant reikšmę iš eilės negalima jos priskirti kintamajam.

## 6. Galutinė kalbos gramatika

```
grammar GLang;

program : statement+ (returnStatement | expressionStatement)? EOF ;

statement
    : variableDeclaration ';'
    | assignment ';'
    | ifStatement
    | printStatement ';'
    | printfStatement ';'
    | whileLoop
    | forLoop
    | queueDeclaration ';'
    | enqueueStatement ';'
    | dequeueStatement ';'
    | functionDeclaration
    | functionCall
    | tryCatchStatement
    | returnStatement
```

```

| readFStatement ';'
;

variableDeclaration : TYPE ID '=' expression;

assignment : ID '=' expression;

queueDeclaration : 'queue' ID '=' '[' ']' ;

enqueueStatement : ID '.' 'enqueue' '(' expression ')' ;

dequeueStatement : ID '.' 'dequeue' '(' ']' ')' ;

expression
: INT #intExpression
| ID #idExpression
| DOUBLE #doubleExpression
| BOOLEAN #booleanExpression
| STRING #stringExpression
| OPERATOR #operatorExpression
| '(' expression ')' #parenthesesExpression
| expression intMultiOp expression #intMultiOpExpression
| expression intAddOp expression #intAddOpExpression
| expression doubleMultiOp expression #doubleMultiOpExpression
| expression doubleAddOp expression #doubleAddOpExpression
| 'sum' '(' expression (',' expression)* ')' #sumExpression
| functionCall #functionCallExpression
;

intMultiOp : '*' | '/' | '%';
intAddOp : '+' | '-' | '*' | '/' | '%';

doubleMultiOp : '*' | '/' | '%';
doubleAddOp : '+' | '-' ;

whileLoop : 'while' '(' expression relationOp expression ')' '{' statement* '}'
;

ifStatement : 'if' '(' expression relationOp expression ')' '{' statement+ '}'
elseStatement? ;

elseStatement : 'else' '{' statement+ '}' ;

tryCatchStatement : 'try' '{' statement * '}' catchClause ;

catchClause : 'catch' '{' statement * '}' ;

forLoop : 'for' '(' variableDeclaration? ';' expression relationOp expression
';' assignment ')' '{' statement+ '}' ;

relationOp : '==' | '!=' | '>' | '<' | '>=' | '<=';

printfStatement : PRINTF '(' STRING, 'expression' ')' ;

readFStatement : TYPE? ID '=' READF '(' STRING ')' ;

```

```

printStatement : PRINT '(' expression ')' ;

functionDeclaration : 'function' ID '(' parameterList? ')' '{' statement*
returnStatement? '}' ;

returnStatement : 'return' expression ';' ;

expressionStatement : expression ';' ;

parameterList : parameter (',' parameter)* ;

parameter : TYPE ID ;

functionCall : ID '(' argumentList? ')' (assignment)? ;

argumentList : expression (',' expression)* ;

TYPE : 'int'
      | 'boolean'
      | 'double'
      | 'string'
      | 'operator'
      ;

STRING : '"' (~["\r\n\\"] | '\\\' .)* '"' ; // string literal
BOOLEAN : 'true' | 'false' ;
PRINT : 'print' ;
PRINTF : 'printf' ;
READF : 'readf' ;
ID : [a-zA-Z]+ ;
INT : [0-9]+ ;
DOUBLE : [0-9]+'.'[0-9]+ ;
OPERATOR : '|' '+' | '|' '-' | '|' '*' | '|' '/' | '|' '%' ;

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

```

## 7. Kaip sukompiliuoti ir paleisti

- 7.1. Nurodyti interpretatoriaus vietą prie paleidžiamų konfigūracijų;
  - 7.2. Nurodyti 20 sdk modelių;
  - 7.3. Nurodyti ką norima paleisti, jei testinį kodą tai argumentų laukelyje rašyti -f ir failo lokaciją bei jo pavadinimą, pvz: -f samples/FibonacciRecursive.
- Jei norima paleisti console tuomet įrašyti -i ir rašyti viską run atsivėrusiame laukelyje.
- 7.4. Paspaudus run jei viskas teisingai atlikta, programa turėtų pasileisti.

## 8. Pavyzdinis kodas ir gauti rezultatai

- 8.1. Unikali savybės “operator” kodo pavyzdys:

```
function f(operator a) {  
    int b = 9;  
    int c = 6;  
    print(b a c);  
}  
operator a = |+|;  
f(a)  
a = |-|;  
f(a)  
a = |*|;  
f(a)  
a = |/|;  
f(a)  
a = |%|;  
f(a)
```

Gautas rezultatas:

```
<PROGRAM OUTPUT>  
15  
3  
54  
1  
3
```

8.2. Išimčių valdymo kodo pavyzdys:

```
int a = 6;
try {
a = a / 0;
}
catch {
a = a + 2;
}
print(a);
```

Gautas rezultatas:

8.3. While ciklo kodo pavyzdys:

```
int fib = 9000;
int x = 1;
int y = 1;
while (y < fib){
print(y);
int next = x + y;
x = y;
y = next;
}
```

Gautas rezultatas:



<PROGRAM OUTPUT>

1  
2  
3  
5  
8  
13  
21  
34  
55  
89  
144  
233  
377  
610  
987  
1597  
2584  
4181  
6765

8.4. For ciklo kodo pavyzdys:

```
for (int i = 0; i < 10; i = i + 1 ){  
    print(i);  
}
```

Gautas rezultatas:

<PROGRAM OUTPUT>

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

8.5. Kintamųjų apimties kodo pavyzdys:

```

for (int i = 0; i < 5; i = i + 1){
    if (2 == 2){
        int a = 5;
        print(a);
    }
    int a = 2;
    print(a);
}
int i = 1;
print(i);

```

Gautas rezultatas:

```

<PROGRAM OUTPUT>
5
2
5
2
5
2
5
2
5
2
5
2
1

```

8.6. If sąlygos kodo pavyzdys:

```

int a = 6;
if (a > 7){
    operator b = |*|;
    print(3 b a);
}
else {
    operator b = |+|;
    print(3 b a);
}

```

Gautas rezultatas:

<PROGRAM OUTPUT>

9

8.7. Funkcijos kodo pavyzdys:

```
1  function fibona(int y) {  
2  int x=1;  
3      if (y < 9){  
4          x = x + fibona(y+1);  
5      }  
6      return x;  
7  }  
8  string a="abc";  
9  int x=2;  
10 print(x);  
11 x=fibona(7);  
12 print(x);
```

Gautas rezultatas

<PROGRAM OUTPUT>

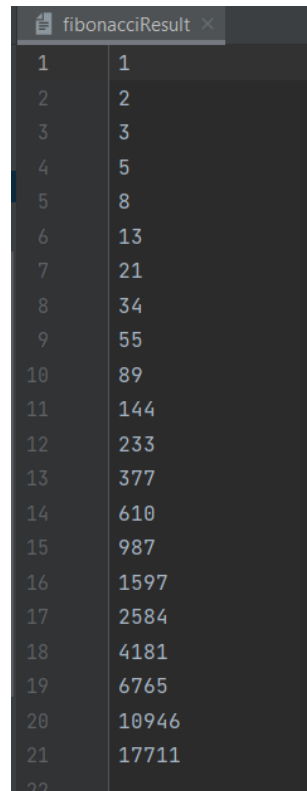
2

3

8.8. Skaitymas ir rašymas į failą:

```
1  function fibonacci(int x, int y, int fib) {  
2      if (y < fib){  
3          printf("C:\Studijos\2 kursas\2 semestras\Programavimo kalbu teorija\Grupės projektas\glang-interpreter-master\resultFiles\fibonacciResult", y);  
4          fibonacci(y, x + y, fib)  
5      }  
6  }  
7  int fib=0;  
8  fib = readf("C:\Studijos\2 kursas\2 semestras\Programavimo kalbu teorija\Grupės projektas\glang-interpreter-master\dataFiles\fibonacci");  
9  fibonacci(1, 1, fib)  
10
```

Gautas rezultatas:



1	1
2	2
3	3
4	5
5	8
6	13
7	21
8	34
9	55
10	89
11	144
12	233
13	377
14	610
15	987
16	1597
17	2584
18	4181
19	6765
20	10946
21	17711
22	

## 9. Naudoti pavyzdžiai, šaltiniai

- <https://github.com/P175B124/glang-interpreter>;
- Dėstytojo Evaldo Guogio tiesioginė transliacija platformoje twitch.