



Univerza v Mariboru

---

Fakulteta za elektrotehniko,  
računalništvo in informatiko

# Projektna naloga

(ni imena?)

Študent:  
Domen Hribnik  
Denis Železnik  
Merisa Mustajbašić

Datum in mesto:  
28.05.2023, Maribor

## Kazalo

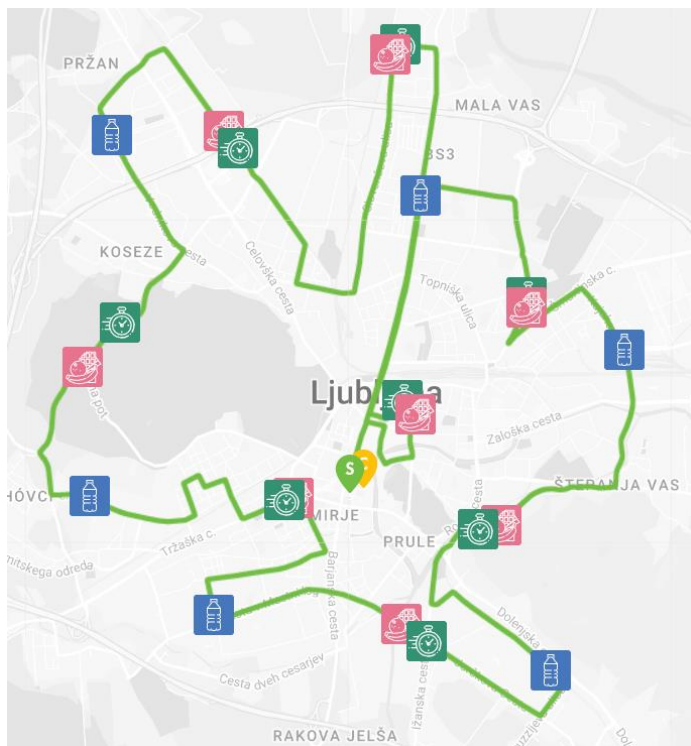
<b>1.</b>	<b><i>Analiza in načrtovanje jezika</i></b> .....	<b>1</b>
1.1.	Konstrukti za opis teka .....	1
1.2.	Definicija BNF jezika .....	2
1.3.	Testni primeri .....	3

# 1. Analiza in načrtovanje jezika

## 1.1. Konstrukti za opis teka

- run - osnovni element celotnega teka
- path - pot, ki jo pretečejo udeleženci
- start - točka, kjer se tek začne
- end - točka, kjer se tek konča
- tima - postaja na trasi, kjer se meri čas in preverja udeleženceva prisotnost
- food - postaja na trasi, kjer udeleženci dobijo hrano
- water - postaja na trasi, kjer udeleženci dobijo vodo

Namen jezika je avtomatizirati ustvarjanje poti na zemljevidu za tekaške dogodke. S pomočjo elementov, kot so "path", "start", "end", "vmesne postaje, ki niso obvezne" (kot so "time", "food", "water"), jezik omogoča opisovanje in ustvarjanje trase tekaškega dogodka. Element "run" se uporablja kot osnovni element za celoten tek, ki določa začetek in konec teka v tem elementu lahko uporabljamo tudi programerske koncepte: spremenljivke, polja, procedure, vejitve in zanke.



Slika 1 - Primer končnega rezultata, ki bi ga radi dosegli

## 1.2. Definicija BNF jezika

```
Run ::= "run" string "{"Path Start End Time Food Water "}"
Path ::= "path" "{" Points "}"
Points ::= Point, Points | Point
Box ::= "(" Point, Point ")"
Point ::= "(" number "," number ")"

Start ::= "start" "{" Point "}"
End ::= "end" "{" Point "}"
Time ::= "time" "{" Box "}" Time | ε
Food ::= "food" "{" Box "}" Food | ε
Water ::= "water " "{" Box "}" Water | ε

For ::= "for" "(" string "in" number range number ")" "{" Statement
      "}"
If ::= "if" "(" Condition ")" "{" Statement "}"
List ::= string "[" ListValues "]"
Procedure ::= "proc " string "(" Parameters ")" "{" statements "}"
           "end"
Assign ::= string "=" Expression

Parameters ::= string Parameters | ε
Statements ::= Statement Statements | ε
Statement ::= Assign | For | If | Procedure_call
Procedure_call ::= string "(" Arguments ")"
ListValues ::= Value | Value "," ListValues
Value ::= number | string | Point | Variable
Arguments ::= Expressions
Expressions ::= Expression Expressions | ε
Expression ::= number | Variable | List | Procedure_call | Operation
Increment ::= string operator | string operator Expression
Condition ::= Expression comparison_operator Expression
Operation ::= Expression operator Expression
Variable ::= string

comparison_operator ::= "<" | ">" | "=="
operator ::= "+" | "-" | "*" | "/" | "%" | "++" | "--"
string ::= [a-zA-Z]+
number ::= [0-9]+
range ::= :
```

### 1.3. Testni primeri

#### Primer 1

```
run "Mariborski tek" {  
  path {  
    (0,0),  
    (1,2),  
    (3,2),  
    (4,4),  
    (5,6),  
    (7,8),  
    (9,9)  
  }  
  start { (0,0) }  
  end { (9,9) }  
  time { box((3.5, 3.5), (4.5, 4.5)) }  
  food { box((2.5, 1.5), (3.5, 2.5)) }  
  water { box((4.5, 5.5), (5.5, 6.5)) }  
}
```

Slika 2 - Osnovni program

#### Primer 2

```
run "Mariborski tek" {  
  t1 = 1  
  t2 = 2  
  path {  
    (0,0),  
    (t1, t2),  
    (3,2),  
    (4,4),  
    (5,6),  
    (7,8),  
    (9,9)  
  }  
  start { (0,0) }  
  end { (9,9) }  
}
```

Slika 3 - Program s spremenljivko

### Primer 3

```
run "Mariborski tek" {  
  var = 5  
  Path {  
    (0,0),  
    (1,2),  
    (var - 2,2),  
    (4,4),  
    (var,6),  
    (7, var + 3),  
    (9, (var-2) * 3)  
  }  
  start { (0,0) }  
  end { (9,9) }  
  time { box((3.5, 3.5),(4.5, 4.5)) }  
  food { box((2.5, 1.5), (3.5, 2.5)) }  
  water { box((4.5, 5.5), (5.5, 6.5)) }  
}
```

Slika 4 - Program s spremenljivko in izrazi

### Primer 4

```
run "Mariborski tek" {  
  list = [1, 2, 3, 4]  
  path {  
    for (x in 0:3) {  
      (list[x], list[x])  
    }  
  }  
  start { (0,0) }  
  end { (9,9) }  
  time { box((3.5, 3.5),(4.5, 4.5)) }  
  food { box((2.5, 1.5), (3.5, 2.5)) }  
  water { box((4.5, 5.5), (5.5, 6.5)) }  
}
```

Slika 5 - Program s poljem

## Primer 5

```
run "Mariborski tek" {  
  path {  
    for (x in 0:10) {  
      (x, x+3)  
    }  
  }  
  start { (0,0) }  
  end { (9,9) }  
  time { box((3.5, 3.5),(4.5, 4.5)) }  
  food { box((2.5, 1.5), (3.5, 2.5)) }  
}
```

Slika 6 - Program s for zanko

## Primer 6

```
run "Mariborski tek" {  
  path {  
    for (x in 0:10) {  
      if (x > 5) {  
        (5, x)  
      }  
    }  
  }  
  start { (0,0) }  
  end { (9,9) }  
  time { box((3.5, 3.5),(4.5, 4.5)) }  
  food { box((2.5, 1.5), (3.5, 2.5)) }  
  water { box((4.5, 5.5), (5.5, 6.5)) }  
}
```

Slika 7 - Program s spremenljivko, if stavkom in for zanko

## Primer 7

```
run "Mariborski tek" {  
  path {  
    for (x in 0:10) {  
      for (y in 0:10) {  
        if (x / 2 == 0) {  
          (x, y)  
        }  
      }  
    }  
  }  
  
  start { (0,0) }  
  end { (9,9) }  
  time { box((3.5, 3.5),(4.5, 4.5)) }  
  food { box((2.5, 1.5), (3.5, 2.5)) }  
  water { box((4.5, 5.5), (5.5, 6.5)) }  
}
```

Slika 8 - for zanka v for zanki

## Primer 8

```
proc drawWater(x, y) {  
  water { box ( (x,x),(y,y) ) }  
}  
  
run "Mariborski tek" {  
  path {  
    (0,0),  
    (1,2),  
    (3,2),  
    (4,4),  
    (5,6),  
    (7,8),  
    (9,9)  
  }  
  
  start { (0,0) }  
  end { (9,9) }  
  time { box((3.5, 3.5),(4.5, 4.5)) }  
  food { box((2.5, 1.5), (3.5, 2.5)) }  
  drawWater(3,2)  
}
```

Slika 9 - Program s proceduro



## Primer 9

```
run "Mariborski tek" {  
  razdalja = 21  
  path {  
    for (x in 0:razdalja) {  
      (x,x)  
    }  
  }  
  start { (0,0) }  
  end { (9,9) }  
  for (y in 0:4) {  
    time { box((y * 5 - 0.5, y * 5 - 0.5), (y * 5 + 0.5, y * 5 + 0.5)) }  
  }  
  food { box((2.5, 1.5), (3.5, 2.5)) }  
  water { box((4.5, 5.5), (5.5, 6.5)) }  
}
```

Slika 10 - Praktičen primer, postaja za čas se izpiše vsakih 5km

## Primer 10

```
run "Mariborski tek" {  
  path {  
    list = [0, 1, 2, 3, 4]  
    for (x in 0:4) {  
      (list[x], list[x])  
    }  
  }  
  start { (list[0], list[0]) }  
  end { (list[4], list[4]) }  
}
```

Slika 11 - Program s poljem, začetek in konec sta prvi in zadnji element polja