Osnovni ukazi v jeziku Julia

Programe v jeziku Julia poženemo tako, da v terminalu poženemo ukaz `julia`

- **julia** zažene se Julia REPL (Read Eval Print Loop).
- **julia program.jl** Poženi kodo v datoteki `program.jl`.
- **julia –project="."** aktivira projekt v trenutnem direktoriju.

Julia REPL(Read Eval Print Loop)

Z vnosom naslednjih znakov se spremeni način REPL

- **`?`** Način dokumentacije (**`help>`**). Vnos **`?ime`** prikaže dokumentacijo za funkcijo `ime`.
- **`;`** Način zunanje lupine (**`shell>`**). Vnašamo lahko ukaze sistemske lupine.
- **`]`** Način paketov (**`pkg>`**). Vnašamo lahko ukaze iz modula `Pkg`.
- **`include("program.jl")`** Požene kodo iz datoteke `program.jl`.
- **`using MojPaket`** Naloži paket `MojPaket`.

Paketi

- **`import("Pkg")`** naloži modul `Pkg`

REPL način paketov (`pkg>`):

- **`activate Direktorij`** Aktviraj projekt v mapi `Direktorij`.
- **`add ImePaketa`** Namesti paket v trenutno aktivirani projekt.
- **`test`** Poženi teste definirane v `test/runtests.jl`.

Kontrolne strukture

```
if 1 == 2
  println("Ena je enako dva.")
end
```

```
for i=1:10
  println("Indeks v zanki je $i")
end
```

```
throw("Zgodilo se je nekaj pričakovano nepričakovanega.")
```

Funkcije

```
f(x,y) = x*y + x  # definicija v eni vrstici
```

```
function mojfun(x, y) # definicija v bloku
 return x*y + y
end
```

```
fun = (x, y) -> x*y + x # anonimna funkcija
```

Podatkovni tipi

```
# podatkovna struktura s polji x in y
struct Point2d {
  x
  y
}
```

```
point = Point2d(1.2, 2.3)
point.x
point.y
```

```
# dodajanje metod za specifične tipe
length(p::Point2d) = sqrt(p.x^2 + p.y^2)
```

Definicija Operatorjev

```
import Base: +
+(p::Point2d, q::Point2d) = Point2d(p.x + q.x, p.y + q.y)
```