

## Rekurzivno programiranje

Iterativno programiranje:

Učenci poznajo iterativno programiranje. Kot primer izdelamo iterativni program za izračun fakultete ( $5! = 1 * 2 * 3 * 4 * 5$ ). V kolikor ne poznajo funkcije fakultete, se razloži, kako funkcijo ima fakuleta  $f(x) = 1 * 2 * 3 * 4 * \dots * x$ ;  $x > 1$

Primer za izračun fakultete števila 5:

$$5! = 1 * 2 * 3 * 4 * 5$$

```
program fakulteta;
function fak(a:integer):integer;
var i,st:integer;
begin
  for i := 1 to a do
    st := st * i;
  fak := st;
end;
begin
  writeln(fak(5));
end.
```

Nato se predstavi kaj je to rekurzija. Rekurzivni podprogrami so taki programi, ki kličejo sami sebe – rekurzivna definicija nekega problema uporablja le tega za opis problema samega.

Odgovor je v tem, da včasih lažje izrazimo rešitev problema tako, da se pri rešitvi sklicujemo na rešitev samo.

Izdelamo problem fakultete na rekurzivni način – rekurzivna funkcija  $f(x) = x * f(x-1)$ ;  $x > 1$

Rekurzivno definicijo sestavlja:

- splošen rekurzivni del
- robni pogoj (rekurzijska spremenljivka doseže mejno vrednost) – končnost rekurzivnih klicev

Rekurzivni klici sicer ne potrebujejo dodatnih spremenljivk, ampak se zaradi večkratnega izvajanja podprograma spremenljivke hranijo na skladu (torej potreba po

večjem pomnilniku) in čas shranjevanja spremenljivk je tudi daljši (počasnejše izvajanje) – slaba lastnost, ampak podprogram je razumljivejši – dobra lastnost.

```
program fakulteta;
function fak(a:integer):integer;
begin
  if a <= 1 then
    fak := 1
  else
    fak := a * fak(a -1);
  end;
begin
  writeln(fak(5));
end.
```

### **Primeri:**

Naredimo primer za fibbonacijeva števila. Prvi dve števili sta 1, vsa naslednja števila pa vsota prejšnjih dveh števil v zaporedju.

$F(x) := F(x-1) + F(x-2)$   $x > 1$ . Potrebno je izdelati program s pomočjo rekurzije.

Rekurzivna funkcija je elegantna in neposredno pove bistvo zaporedja.

```
program fibb;
function fib(a:integer):integer;
begin
  if a <= 2 then
    fib := 1
  else
    fib := fib(a-1) + fib(a -2);
  end;
begin
  writeln(fib(5));
end.
```

```

function fib(a:integer):integer;
Var i,f1,f2,p:integer;
Begin
  F1 := 1;f2:= 1;
  I := 2;
  while i < n do
  Begin
    I := i + 1;
    P := f2;
    F2 := f1 + f2;
    F1 := p;
  End;
  Fib := f2;
End;

```

Prikaz funkcije izračuna fakultete

$5! = 1 * 2 * 3 * 4 * 5 \quad \rightarrow$	$f(x) = x * f(x-1) \quad \text{za } N > 1$
<pre> program fakulteta; function fak(a:integer):integer; var i,st:integer; begin   for i := 1 to a do     st := st * i;   fak := st; end; begin   writeln(fak(5)); end. </pre>	<pre> program fakulteta; function fak(a:integer):integer; begin   if a &lt;=1 then     a := 1   else     fak := a * fak(a-1);   end; begin   writeln(fak(5)); end. </pre>

<pre> program fibb;  function fib(a:integer):integer;  begin   if a &lt;= 2 then     fib := 1   else     fib := fib(a-1) + fib(a -2);   end; begin   writeln(fib(5)); end. </pre>	<pre> Function fib(a:integer):integer;  Var i,f1,f2,p:integer;  Begin   F1 := 1;f2:= 1;   I := 2;   While i &lt; n do     Begin       I := I + 1;       P := F2;       F2 := F1 + f2;       F1 := P;     End;   Fib := f2; End; </pre>
---	--

[illegible]

- Primer palindroma (ko se beseda bere enako iz ene in druge strani).
- Primer seštevanje (iskanje vsote), odštevanja (iskanje razlike), množenja (iskanje produkta), deljenja (iskanje količnika)!
- Napiši program, ki bo prebrani niz izpisal v obratnem vrstnem redu (znak . naj označuje konec niza).