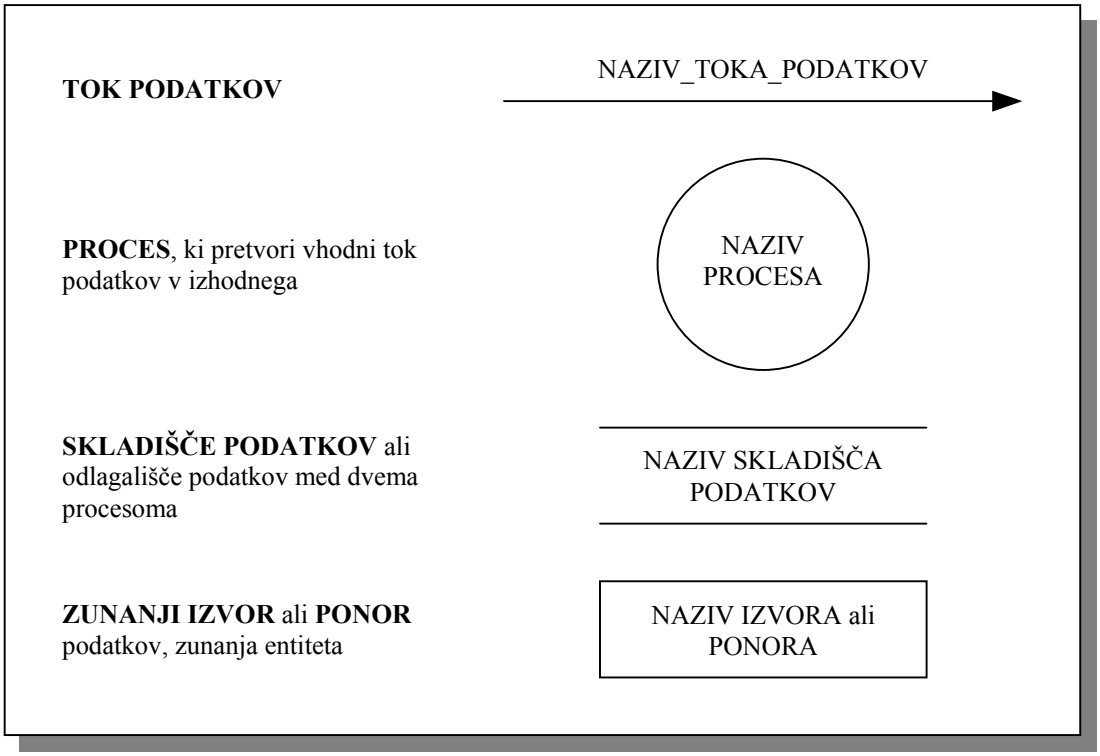
Diagram toka podatkov (Data Flow Diagram)



Gradniki diagrama toka podatkov (Yourdon-DeMarco notacija)

TOK PODATKOV

Konceptualni standardi

- Podatkovni tok ponazarja potek informacij od enega objekta (procesa, izvora, ponora ali skladišča) k drugemu objektu.
- Podatkovni tok predstavlja množico vhodnih ali izhodnih (glede na proces) podatkov, ki imajo enolično definirano vsebino in strukturo.
- Podatkovni tokovi ne morejo spremeniti podatkov.
- Podatki ne morejo biti ustvarjeni ali uničeni znotraj podatkovnega toka.
- Podatkovni tokovi ne morejo biti uporabljeni za povezovanje prožilnih ali odločitvenih zastavic.
- Podatkovni tokovi ne morejo neposredno prehajati od terminatorja (izvor ali ponor) k shrambi.
- Podatkovni tokovi ne morejo neposredno prehajati od terminatorja do terminatorja.

Notacijski standardi

- Podatkovni tok je predstavljen z usmerjeno črto, opremljeno z labelo (označbo).
- Vsak podatkovni tok mora biti označen.

- Labela je samostalnik; glagolske oblike ne smejo biti uporabljene.
- Vsi podatkovni tokovi morajo nastopati v podatkovnem slovarju.
- Označbe se morajo izogibati generičnim računalniškim pojmom (podatek, vhod …).

Izjema

- Podatkovni tok, usmerjen proč od skladišča ali v skladišče, ne rabi biti poimenovan, če predstavlja skladišče v celoti.

PROCES

Konceptualni standardi

- Proces predstavlja množico aktivnosti, ki vhodne podatke pretvorijo v izhodne.
- Proces ne more niti ustvariti niti uničiti podatkov (vse izhode procesa je mogoče dobiti z nekim postopkom na podlagi vhodov).
- Proces dobi le tiste podatke, ki jih potrebuje (ne nakazujemo prehodnih podatkov).
- Proces lahko spremeni podatke na dva načina:
  - Fizično – vhod je le malo ali nič podoben izhodu.
  - Logično – vhod in izhod sta fizično enaka, vendar sta obravnavana različno.
- Procesi ne »pošiljajo ali potiskajo« podatkov, temveč predstavljajo njihovo preslikavo.

Notacijski standardi

- Proces je ponazorjen s krogom, ki vsebuje številko in označbo.
- Naziv procesa je glagol, glagolski samostalnik ali zaporedje besed, ki opisujejo vrsto dejavnosti.

SKLADIŠČE PODATKOV

Konceptualni standardi

- Skladišče (zbirka) podatkov je koncept, ki predstavlja prostor za shranjevanje podatkov iz nekega procesa, tako da so lahko ti na voljo drugim procesom.
  - Skladišče je začasen nosilec informacij.
  - Skladišče je obravnavano tudi kot podatkovni tok v mirovanju.
- Skladišče ne more ustvariti ali uničiti podatkov.
- Skladišče mora imeti vsaj en vhodni in vsaj en izhodni podatkovni tok.
  - Najmanj en proces mora pisati v skladišče.
  - Iz skladišča ni mogoče brati podatkov, ki niso bili vanj zapisani.
- Struktura in vsebina skladišča je lahko definirana s podatkovnim modelom, ki predstavlja podatkovni vidik vseh tokov, usmerjenih v skladišče.
- Med fazo analize pomenijo zbirke logične sklope podatkov v organizacijskem sistemu, v fazi implementacije pa lahko postanejo fizične datoteke, baze podatkov ali dokumenti.
- Neposreden pretok podatkov med zbirkami ni mogoč.
- Zbirka se lahko pojavi na kateremkoli nivoju, razen na kontekstnem.

Notacijski standardi

- Skladišče podatkov predstavljata dve vzporedni premici.
- Skladišče mora imeti oznako, ki natančno določa vsebino.
- Naziv skladišča je največkrat enak nazivu vhodnih podatkovnih tokov.
- Kadar nek proces hkrati piše v skladišče podatkov in iz njega tudi bere, je dvosmerni podatkovni tok predstavljen z dvema usmerjenima povezavama.

ZUNANJI IZVOR ali PONOR PODATKOV

Konceptualni standardi

- Izvori ali ponori podatkov so koncepti, ki predstavljajo zunanje procese ali zunanje organizacijske sisteme.
  - Zunanji izvori in ponori podatkov se nahajajo izven interesnega področja naše analize, tako da nas nihova struktura in obnašanje ne zanimata, zanimajo pa nas podatkovni tokovi, ki jih povezujejo z obravnavanimi notranjimi procesi.
- Nek organizacijski sistem je lahko istočasno zunanji izvor in zunanji ponor podatkovnih tokov enega ali več procesov.

Notacijski standardi

- Simbol zunanjega izvora ali ponora je pravokotnik z vpisanim nazivom.

PODATKOVNI SLOVAR

Podatkovni slovar razčleni podatkovne tokove in shrambe podatkov do podatkovnih prvin. Primitivnemu, nesestavljenemu elementu (.DE – »data element«) mora biti prirejen eden od standardnih podatkovnih tipov (integer, float, text, date …). Notacija:

=	izenačitev (kompozicija)
+	logični IN
[…] |	izbira (logični ALI)
{…}	ponavljanje
(…)	možnost (opcija)
*	komentar

Dovoljeno je gnezdenje. Pri ponavljanju smeta biti podani spodnja in zgornja meja števila ponovitev. Elementi v oklepaju, ki označuje opcijo, so lahko bodisi prisotni bodisi odsotni.

Zgled:

knjiga = avtor + naslov + jezik + leto_izida + {ključna_beseda}
kriteriji_iskanja_knjig = (avtor) + (naslov) + (jezik) + (leto_izida) + {ključna_beseda}
rezultati_iskanja_knjig = {avtor + naslov + {ključna_beseda} + {izvod}}
izvod = signatura + status + (rok_vrnitve)
status = [prost | izposojen | za_čitalnico]

$$\text{telefonska\_številka} = \left[\begin{array}{c} \text{lokalna\_številka} \\ 0 + \text{zunanja\_številka} \\ 9 * \text{hišna centrala} * \end{array}\right] * \text{velja za FERI} *$$

lokalna_številka = 5 + 3{poljubna_številka}3
poljubna_številka = [1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0]

MODELIRANJE PROCESNE LOGIKE (MINISPECIFIKACIJE)

Minispecifikacije podamo za tiste procese, ki niso razgrajeni, so torej atomarni procesi, imenovani tudi funkcionalne primitive. Z njimi opišemo preslikavo vhodnih podatkov v izhodne. Posežemo lahko po naslednjih tehnikah:

- strukturirani jezik,
- odločitvene tabele,
- odločitvena drevesa,
- diagrami prehajanja stanj.

STRUKTURIRANI JEZIK

Uporablja omejen slovar in omejeno sintakso. Lastnost strukturiranosti mu daje oblika, temelječa na zamikih, s katerimi so nakazani tisti deli opisa, ki pomensko sodijo skupaj. Podpira tri osnovne proceduralne konstrukte: zaporedje ukazov, iteracijo in pogojni stavek.

Zaporedje

```
začetek
   stavek 1
   stavek 2
   ...
   stavek n
konec
```

Pogojni stavek

```
/* pogojni stavek
če (pogoj) potem
   stavek ali blok stavkov
sicer
   stavek ali blok stavkov
konec pogoja

/* stikalo
izbira
   izbor 1 (pogoj 1)
      stavek ali blok stavkov
   ...
   izbor n (pogoj n)
      stavek ali blok stavkov
   ostalo (nobeden od pogojev ni izpolnjen)
      stavek ali blok stavkov
konec izbire
```

Iteracija

```
/* ponavljaj stavke
ponavljaj x krat
   stavek ali blok stavkov
konec ponavljanja

/* ponavljaj stavke, dokler velja pogoj
dokler velja (pogoj) ponavljaj
   stavek ali blok stavkov
konec ponavljanja

/* ponavljaj stavke, dokler ni pogoj izpolnjen
ponavljaj
   stavek ali blok stavkov
dokler (pogoj) ni izpolnjen
```

Razširitev strukturiranega jezika za delo s tabelami in uporabo SQL-a

Branje in pisanje v tabelo	
Preberi zapis iz tabele v spremenljivko V	`V = [TABELA.STOLPEC]`
Zapiši vrednost V v tabelo	`[TABELA.STOLPEC] = V`
Navigacija med zapisi	
Premakni se na naslednji zapis	`[TABELA].naprej`
Premakni se na prejšnji zapis	`[TABELA].nazaj`
Premakni se na prvi zapis	`[TABELA].prvi`
Premakni se na zadnji zapis	`[TABELA].zadnji`
Dodajanje in brisanje zapisa	
Kreiraj nov zapis v tabeli	`[TABELA].nov_zapis`
Briši zapis iz tabele	`[TABELA].briši`
Delo s skupinami zapisov	
za vse zapise iz [TABELA] **kjer velja** (pogoj) **izvedi** stavek ali blok stavkov **konec izvajanja**	

```
/* SQL deklaracija
SQL začetek
   declare cursor Cx
      select stolpec1, stolpec2 ...
      from tabela1, tabela2 ...
      where pogoj
SQL konec
/* nadaljujemo v navadnem načinu
za vsak zapis iz Cx izvedi
   stavek ali blok stavkov
konec izvajanja
```

```
/* SQL izvedba
SQL začetek
   update TABELA
   set kol = kol * 1.5
   where ID_ARTIKEL = 'A001'
SQL konec
```

Podprogrami

```
/* definiranje podprograma
podprogram Izračunaj (p1, p2)
   Izračunaj = p1 + p2
vrnitev
-------------------------------------------------
/* klic podprograma
vsota = pokliči( Izračunaj( a, b ) )
```

Pravila pisanja v strukturiranem jeziku

1. Celotno logiko predstavi z uporabo zaporedja, iteracije in pogojnega stavka.
2. Uporabljaj zamikanje stavkov, da prikažeš hierarhično strukturo procesne logike.
3. Besede, ki predstavljajo konstrukte strukturirane slovenščine (začetek, konec, če, potem, sicer …), podčrtuj.
4. Besede, ki so definirane v slovarju ali repozitoriju in identificirajo objekte, piši z velikimi črkami. Primer: ime shrambe – KNJIGE, ime toka – KNJIGA, ime prvine – KNJ_ISBN, oznaka procesa – P_VNOS_KNJIGE …
5. Celotna imena procesov iz funkcionalne dekompozicije piši z veliko začetnico. Primer: ime procesa – Izdajanje opominov članom.
6. Vse konkretne vrednosti piši v navednicah. Primer: KNJ_ISBN = '0-201-36082-5'.

Zgled – izračun davka za določeno obdobje

Navadno strukturirano besedilo

```
davek = 0
dokler velja (obstajajo zapisi v RAČUN) ponavljaj
  če (DATUM_PRODAJE v izbranem obdobju) potem
    dokler velja (obstajajo elementi prodaje v POSTAVKA_RAČUNA) ponavljaj
      izberi
        izbor 1 (TIP_ELEMENTA = 's')
          davku prištej 6.5% od CENA
        izbor 2 (TIP_ELEMENTA = 'a')
          davku prištej 20% od CENA
        ostalo
          izpiši sporočilo o napaki
      konec izbire
    konec ponavljanja
  konec pogoja
konec ponavljanja
izpiši davek
```

Razširjen strukturiran jezik – brez uporabe SQL-a

```
davek = 0
za vse zapise iz [PRODAJA]
kjer velja ([DATUM_PRODAJE] v izbranem obdobju) izvedi
  za vse zapise iz [ELEMENT_PRODAJE]
  kjer velja ([PRODAJA.ID] = [ELEMENT_PRODAJE.ID]) izvedi
    izberi
      izbor 1 ([TIP_ELEMENTA] = 's')
        davek = davek + [CENA] * 0.065
      izbor 2 ([TIP_ELEMENTA] = 'a')
        davek = davek + [CENA] * 0.20
      ostalo
        izpiši sporočilo o napaki
    konec izbire
  konec izvajanja
konec izvajanja
izpiši davek
```

Razširjen strukturiran jezik – z uporabo SQL-a (izračun za obdobje od 1. 3. 03 do 16. 3. 03)

```
SQL začetek
  declare cursor C_davek
    select sum(decode(e.TIP_EL, 's', 0.065 * e.CENA, 'a', 0.20 * e.CENA))
    from PRODAJA p, ELEMENT_PRODAJE e
    where DATUM_PRODAJE >= '1.3.03' and
          DATUM_PRODAJE < '16.3.03' and
          e.ID = p.ID
SQL konec
izpiši C_davek
```