Dr Dinu Dragan

PROGRAMIRANJE I PROGRAMSKI JEZICI

KO IZVODI NASTAVU

Nastavnik:

Docent dr Dinu Dragan

Kabinet: NTP 330

Telefon: ---

E-mejl: dinud@uns.ac.rs

Asistent:

MSc Radovan Turović

Kabinet: NTP 328

Telefon: ---

E-mejl: radovan.turovic@uns.ac.rs

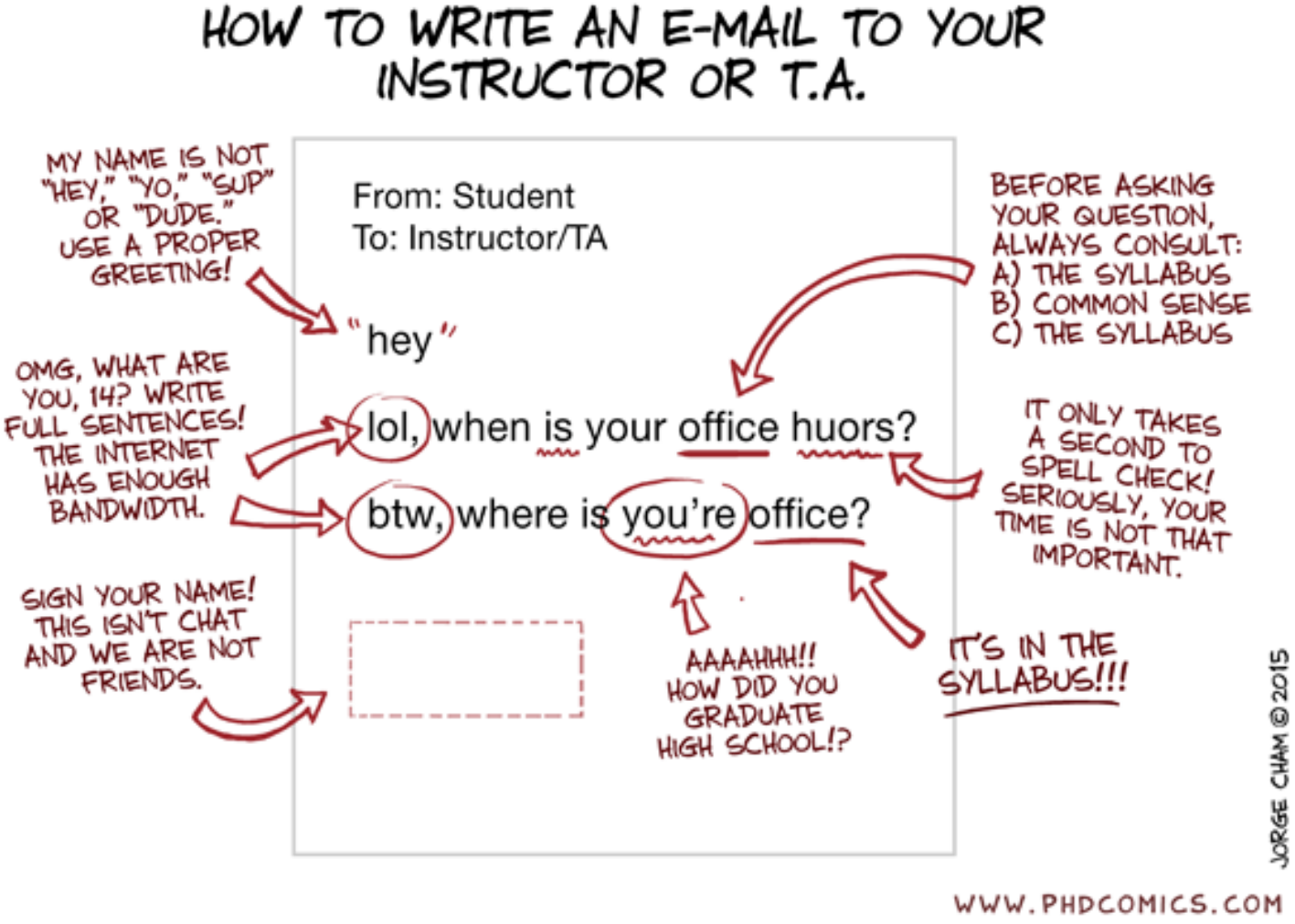
http://?

Ako postoje pitanja, iskoristiti za to:

- Slobodno vreme na predavanjima i vežbama
- Individualne konsultacije (dogovor tokom prvih nedelja)
- E-mejl

Asistenti nisu dibager !

O PISANJU E-MEJLOVA



O NASTAVNIKU - *Dr Dinu Dragan*

- 1998. - 2003. – Elektrotehnika, smer Računarstvo i upravljanje sistemima, usmerenje Računarska nauka i informatika, FTN, Novi Sad
- 2003. – Diplomski rad, “Primena OMR algoritama u DMS sistemima,” FTN
- 2008. – Magistarska teza, “Enkapsulacija JPEG2000 kompresione tehnike u DICOM standard,” FTN
- 2013. – Doktorska disertacija, “Metrika prihvatljivosti kompresione tehnike mirne slike u implementaciji PACS sistema,” FTN
- Od februara 2014. godine docent na Departmanu za računarstvo i automatiku Fakulteta tehničkih nauka
- Interesovanja: kompresija podataka, HCI, računarska grafika, računarska vizija, multimedija, fotogrametrija, upravljanje projektima, …
- **Praktični projekti**

TEME IZUČAVANJA

1. Kratak pregled organizacije računara
2. Proces razvoja programa – apstrakcija, tipovi podataka i kôd
3. Struktura C programa
4. Tipovi podataka C jezika
5. Pokazivači i dinamička alokacija memorije
6. Operatori i izrazi
7. Programske upravljačke strukture
8. Funkcije – parametri, makroi
9. Datoteke
10. Komunikacija sa okolinom – tastatura/miš/monitor, komandna linija

NAČIN POLAGANJA (PST)

1. 2 zadatka na vežbama, ukupno 50 poena
   - 1. zadatak 25 poena
   - 2. zadatak 25 poena
   - svaki zadatak se brani
2. Testovi na predavanjima, ukupno 20 poena
3. Usmeni ispit, ukupno 30 poena
4. Aktivnost na predavanjima i vežbama
5. Za potpis potrebno 36 poena iz predispitnih obaveza

Broj bodova	Ocena
51 – 60	Šest
61 – 70	Sedam
71 – 80	Osam
81 – 90	Devet
91 – 100	Deset

NAČIN POLAGANJA (MEH)

1. 2 zadatka na vežbama, ukupno 50 poena
   - 1. zadatak 25 poena
   - 2. zadatak 25 poena
   - svaki zadatak se brani
2. Testovi na predavanjima, ukupno 20 poena
3. Usmeni ispit, ukupno 30 poena
4. Aktivnost na predavanjima i vežbama
5. Za potpis potrebno 36 poena iz predispitnih obaveza

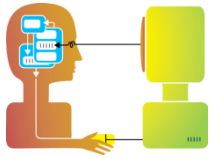
Broj bodova	Ocena
51 – 60	Šest
61 – 70	Sedam
71 – 80	Osam
81 – 90	Devet
91 – 100	Deset

LITERATURA

1. **Brošura sa slajdovima, prateći kod + vaše beleške**
2. Dragan Ivetić, Strukturirani pristup programiranju: inženjering, algoritmi i programski jezici Paskal i C, FTN Novi Sad, 2005.,
3. Laslo Kraus, Programski jezik C sa rešenim zadacima, Akademska misao, 2001.,
4. Brian W. Kernighan, Dennis M. Ritchie, The C Programming Language, 2nd ed., Pretince Hall, Savremena, 1988.,
5. Augie Hansen, Programiranje na C jeziku – potpuni vodić za programski jezik C, Mikro knjiga, 1991.

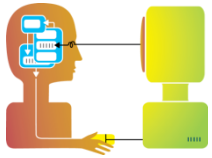
UVOD

PISANJE PROGRAMA



- Nije lako napisati program
- Morate reći računaru šta treba da radi i pri tom mu opisati svaki korak koji treba da se izvrši
- Nema mesta za greške, jer računar ne zna da interpretira, slepo izvršava svaku naredbu
- Dobar program je rezultat razmišljanja na pravi način
- **Program nije ništa drugo nego formalno izražena ideja**
  - Kreće se od ideje šta se želi (šta program/računar treba da radi)
  - Okvir onog što će se (što treba da se) uradi, izražava se kroz algoritam
  - Algoritam omogućuje ne samo da se izrazi ideja o zadatku, nego i o koracima koji se moraju preći da bi zadatak rešio (iz čega proizilazi program

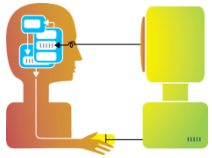
ALGORITAM …



- Algoritam je precizno definisana procedura za rešavanje nekog problema.

- Računarstvo je nauka o algoritmima:
  - njihovim formalnim osobinama
    - tačnost, ograničenja, efikasnost, cena
  - njihovim hardverskim realizacijama
    - projektovanje računarskih sistema
  - njihovim jezičkim realizacijama
    - programiranje i programski jezici
  - njihovim primenama
    - inženjerstvo, astronomija, fizika, biologija, hemija...

… ALGORITAM

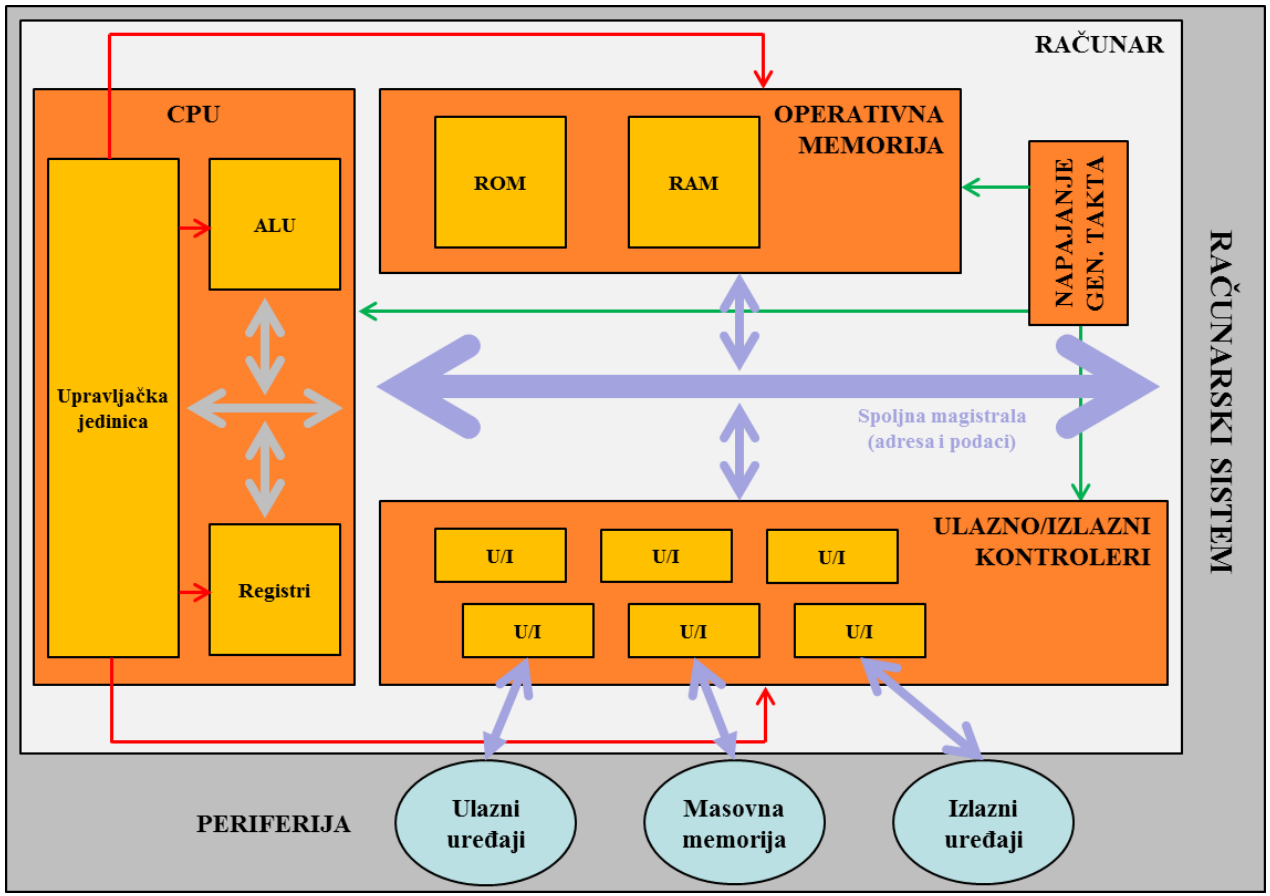


- Kada se opisuje algoritam, ne moraju svi delovi (koraci) da budu do detalja opisani (specificirani), sve dok su oni determinisani i dok vode ka jasnom rešenju

- Mogu se pisati na bilo kom jeziku, mogu biti grafički i/ili tekstualni

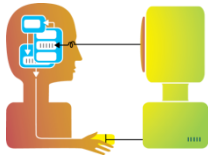
- Popunjavanjem svih detalja algoritam postaje program

ORGANIZACIJA RAČUNARA



- brzi procesor/memorija naspram sporih izlazno/ulaznih uređaja
- preprogramirani set instrukcija
- **Računar nije pametan, samo je dobar u brzom ponavljanju rutina**

PROGRAMIRANJE



- **Programiranje == davanje uputstva računaru**
- Procesorske (engl. Central Processing Unit – CPU) instrukcije:

c = a + b

Pročitaj vrednost sa lokacije **a**

Pročitaj vrednost sa lokacije **b**

Saberi

Upiši vrednost na lokaciju **c**

- Programske jezike je lakše razumeti nego CPU instrukcije
- Potrebno je prevesti kod na CPU instrukcije da bi ga CPU razumeo

GENERACIJE PROGRAMSKIH JEZIKA ...

1. GL – MAŠINSKI JEZICI

- programiranje korišćenjem binarnog koda (0 i 1)
- uporedo sa prvim komercijalnim računarima
- komande zavise od arhitekture procesora, nema portovanja
- potrebna ekspertiza u u elektronici i digitalnim sistemima
- svi programski jezici se na kraju prevode na mašinski

2. GL – ASEMBLERSKI JEZICI

- simboli ili mnemonici koji odgovaraju instrukcijama mašinskog jezika
- prvi put se javlja kompajliranje (prevođenje na mašinski jezik)
- manja zavisnost od hardverske platforme
- i danas se koriste za pisanje pojedinih delova sistemskog softvera (**! C**)

... GENERACIJE PROGRAMSKIH JEZIKA

3. GL – JEZICI VIŠEG NIVOA (KOMPAJLERSKI I INTEPRETERSKI)

- nezavisni od fizičke arhitekture
- dorađena verzija jezika druge generacije (proširen alfabet)
- jedna naredba prevodi se u više naredbi mašinskog jezika
- jezičke strukture – utvrđena struktura programskog koda
- podrazumevaju kompajler ili interpreter

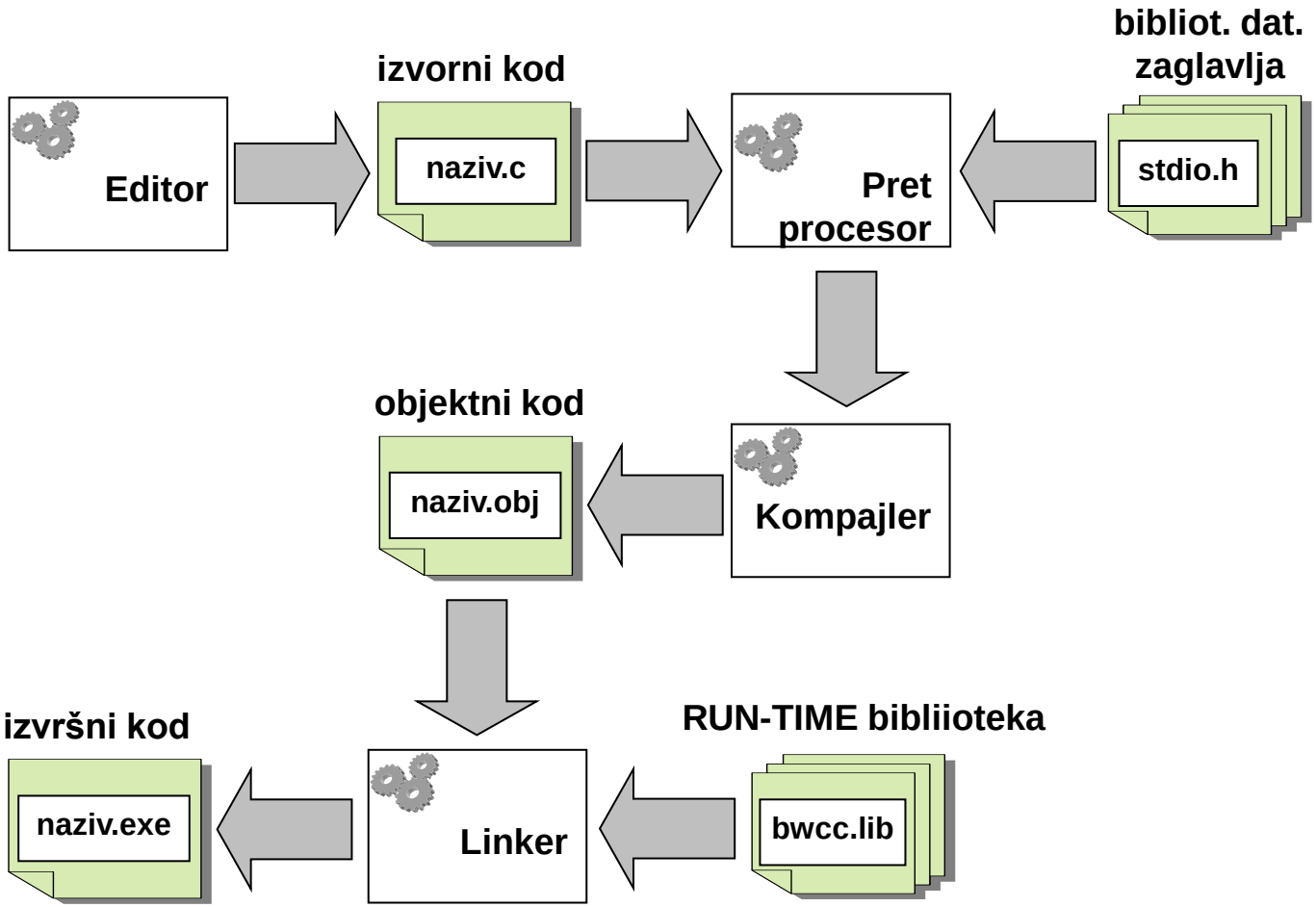
4. GL – PROBLEM ORIJENTISANI JEZICI (SQL)

- jezici posebne namene / problem orijentisani jezici (application specific)
- zahtevaju minimalno programersko znanje, ali je primena usko specifična
- neproceduralni jezici visokog nivoa izgrađeni oko baze podataka (SQL)

STILOVI PROGRAMIRANJA

- Pogled koji programer ima nad programom i njegovih izvršavanjem
- Četiri stila:
  1. **imperativni stil**
     - **program = algoritam + podaci (Fortran, Pascal, C, Basic)**
  2. funkcionalni stil
     - program = funkcija funkcija (Lisp)
     - deklarativni jezici, rekurzivno funkcije i vrednosti
  3. matematičko-logički
     - program = činjenice + pravila (Prolog)
     - primena matematičke logike
  4. objektno-orijentisani
     - program = objekti + poruke (SmallTalk, C++, Java, Pyton, C#)
     - programski objekti apstrakcije realnih objekata

OD KODA DO PROGRAMA



PROCES RAZVOJA PROGRAMA

PROCES RAZVOJA PROGRAMA

Prof. dr Dragan Ivetić:

„Kako izuzetno kompleksne, neformalne i često protivrečne realne probleme čoveka i njegovog okruženja rešavati (automatizovati) pomoću funkcionalno ograničene ali zato brze mehaničke (formalne) mašine – računara?„

„Etapnom transformacijom polaznog problema (njegove statike i dinamike) u objekte koji su interesantni sa date tačke gledišta (apstrakcija) a koji se mogu matematički (aritmetički i logički) rešiti – jezik računara!“

RAZVOJ PROGRAMA JEDNOSTAVNE FUNKCIONALNOSTI ...

1. ANALIZA PROBLEMA

- šta program treba da radi?
- **IDENTIFIKOVATI PROBLEM** – opisati problem u nekoliko rečenica, šta se očekuje od programa.
- **ODREĐIVANJE ULAZA I IZLAZA** – koje podatke program treba da dobije da bi obavio željenu radnju i koje podatke će program vratiti korisniku (uvek barem jedan izlaz)

2. FUNKCIONALNA DEKOMPOZICIJA PROBLEMA

- razlaganje problema na potprobleme, do elementarnih problema koji se znaju rešiti i implementirati
- usputno se identifikuju potrebni podaci

... RAZVOJ PROGRAMA JEDNOSTAVNE FUNKCIONALNOSTI ...

3. RAZVOJ ALGORITMA

- postupak (način razmišljanja) kojim se iz datih polaznih podataka, određenim konačnim nizom računskih i/ili logičkih postupaka dolazi do traženih rezultata
- osobine: diskretnost, rezultativnost, determinisanost, masovnost, (i) efikasnost
- primeri: recept, uputstvo za sklapanje, uputstvo...
- tekstualne ili grafičke notacije za opis algoritama
- nezavisno od programskog jezika (ili nametnuto progr. jezikom)
- piše se u skladu sa projektom razrađenim u prethodnim fazama

... RAZVOJ PROGRAMA JEDNOSTAVNE FUNKCIONALNOSTI

4. PROGRAMIRANJE

- pisanje programa prema algoritmu (preslikavanje 1:n)
- kompajlerski ili interpreterski jezici

5. TESTIRANJE

- izvršavanje programa nad test podacima s ciljem da se pronađu greške
- jedinično testiranje
- integrisano testiranje
- sistemsko testiranje

PRIMERI REŠAVANJA PROBLEMA

- **Primer 1.** Kalkulator celih brojeva za sabiranje i oduzimanje.
  - Šta su ulazi i izlazi?
  - Kako čuvati podatke?
  - Postoji li potreba za dekompozicijom?

- **Primer 2.** Sortiranje spiska zaposlenih po imenu ili jmbg-u.
  - Šta su ulazi i izlazi?
  - Kako čuvati podatke?
  - Postoji li potreba za dekompozicijom?

APSTRAKCIJA U RAZVOJU PROGRAMA …

- “Koncept ignorisanja svih svojstva subjekta koja nisu relevantna za tekuću svrhu sa ciljem koncentrisanja samo na ona koja to jestu”
- Svet koji nas okružuje može se shvatiti kao sistem entiteta koji su u međusobnom delovanju
- Grčki: entite == biće, postojanje
- Pod entitetom podrazumevamo klasu objekata posmatranja koji su važni sa određenog stanovišta i koja se može jedinstveno identifikovati
- Entitet može biti:
  - Realni objekat (osoba, kuća, dokument, automobil, ...)
  - Apstraktni koncept (veličina, boja, radno mesto, ...)
  - Događaj (rođenje, upis, isplata, ispit, ...)
  - Odnos (nastavnik-predmet, proizvod-prodacnica, ...)

… APSTRAKCIJA U RAZVOJU PROGRAMA …

- Svet poseduje različita svojstva – atributima ili obeležja
- Atributima se bliže određuje entitet
- Svaki entitet može imati više atributa, jedan atribut može se naći u više entiteta
- Svaki atribut ima svoje ime i vrednost do konkretnog i jedinstvenog predstavn
- Konkretizacija vrednosti atributa dovodi ika entiteta što se naziva **pojavom entiteta** ili **objektom**
  - U primeru: Pera Perić, ...

Entitet sa atributima

… APSTRAKCIJA U RAZVOJU PROGRAMA …

- Entitet se iz realnog sveta primenom **programske apstrakcije** preslikava na programsku predstavu (programski model) entiteta koja se naziva **tipom entita**
- Primeri:
  - ČOVEK => apstrakcija STUDENT
  - ČOVEK => apstrakcija VOZAČ

apstrakcija STUDENT

apstrakcija VOZAČ

TIPOVI ENTITETA

Prezime
Ime
Broj Indeksa
Adresa
Stud. program
Prosečna ocena

Programski modeli

Prezime
Ime
Adresa
Vozačke kategorije

… APSTRAKCIJA U RAZVOJU PROGRAMA

- Nad tipovima entiteta (tačnije nad njihovim pojavama) primenjuju se odgovarajuće programske operacije

TIP ENTITETA - STUDENT

Prezime
Ime
Broj Indeksa
Adresa
Stud. program
Prosečna ocena

primena

Programske operacije

novi student
nova adresa
novi stud. program
obrisi studenta
izvuci prosek
prikazi podatke
…

izvršavanje programa

Pera
Perić
8882
A
Stud. program
Prosečna ocena

Prezime
Ime
Broj Indeksa
Adresa
Stud. program
Prosečna ocena

Podaci - Informacija