

PROGRAMIRANJE 1

Programiranje – temeljni pojmovi

- Programiranje, programska podrška
 - Sistemska
 - Aplikacijska
- Program, instrukcija
- Programski jezici
 - strojni
 - asembleri
 - programski jezici visoke razine
- Programi prevoditelji
 - Kompilatori
 - Interpretatori

Programiranje

- **Programiranje** je postupak kojim se problem kojeg treba riješiti oblikuje tako da ga se može računalno obraditi.
- Oblikovanje ovdje podrazumijeva stvaranje slijeda instrukcija prema kojima računalno izvodi aritmetičke i logičke operacije.
- Programiranje na najvišoj razini se dijeli na:
 - Definiranje problema (utvrđivanje načina rješavanja i pogodnosti za računalnu obradu)
 - **Rješavanje problema** i njegovo oblikovanje u najprikladnijem programskom jeziku
 - Provjeravanje ispravnosti programa.

- Da bi računalo riješilo zadatak, treba ga pretvoriti u oblik koji uključuje samo pamćenje, logičko rasuđivanje.
- Da bi si olakšao pretvorbu zadatka iz svijeta koji ga okružuje u oblik prihvatljiv računalu, čovjek se služi mnogim pomoćnim postupcima.
- Što je zadatak složeniji, to je u načelu potrebno više pomoćnih postupaka.
- Konačni će rezultat svih tih postupaka biti računalni program

Program - Instrukcija

- Želimo li da računalo nešto uradi, moramo mu potanko obrazložiti što hoćemo – problem je točno izraziti svoje “želje”. Računalo nam može pomoći u traženju rješenja, ali samo ako ga pravilno uputimo
- Za svaku zadaću moramo dati detaljne i točne upute kako računalo ne bi bilo u nedoumici što uraditi. Upute koje računalo razumije zovu se program
 - **Računalni program** je, dakle, organizirani skup instrukcija koje se izvode određenim redoslijedom i s određenim ciljem.
 - Bez programa računalo ne može ništa učiniti.

- Program se sastoji se od liste varijabli i liste instrukcija koje kažu računalu što učiniti s varijablama.
 - **Instrukcija** (naredba, uputa) je iskaz s kojim se kaže računalu da nešto učini. Sastoji se od operacije koju treba izvesti.
 - Svaka instrukcija obavi djelić postavljenog zadatka.

- Računalo može riješiti postavljeni zadatak samo ako dobije uputstvo (niz instrukcija) kako to učiniti.
 - Kombinacijom instrukcija nastaje program.
- **Računalni program** je skup **instrukcija** nanizanih strogo utvrđenim redoslijedom, čijim se izvršenjem obavlja željeni posao
- Čovjek koji piše program naziva se **programer**, a postupak nastanka programa naziva se **programiranje**
 - Programer pišući program niže instrukcije u smislenom redoslijedu.
- Instrukcije moraju biti u obliku u kojem će ih računalo razumjeti.

- Vic:

Poslala žena programera u trgovinu: “Kupi mi margarin, a ako bude jaja, kupi 10”



Poslije pola sata, vraća se programer s 10 paketića margarina i govori ženi: “Bilo je jaja”



- Vic:

Programer hvata slona prema slijedećem postupku:

1. otići u Afriku
2. početi od Rta Dobre nade
3. ići sa kraja na kraj kontinenta (lijevo i desno), krećući se se postepeno k sjeveru
4. u svakom prolasku s kraja na kraj kontinenta:
 - a) uhvatiti svaku životinju koju ugledate
 - b) usporediti njene osobine sa osobinama nekog poznatog slona
 - c) zaustaviti se kada se detektira podudaranje
 - d) u protivnom nastaviti dalje

Iskusni programer: prije nego što primjeni opisani algoritam, postavi jednog slona u Kairo, kako bi bio siguran da će se algoritam sigurno završiti.

C++ programer: izvršava opisani algoritam na rukama i koljenima.

Programska podrška

- **Računalni sustav** se sastoji od **programske podrške** (eng. Software) i **tehničke podrške** (eng. Hardware)
- Programska podrška (eng. software) je termin za sve programe koje računalo može izvršiti kao i podaci s kojima ovi programi mogu izvršiti različite operacije.
- **Programska podrška** se dijeli u dvije osnovne grupe:
 - **sistemska programska podrška** (eng. system software)
 - proizvođač računalne opreme je isporučuje korisniku zajedno sa sustavom tehničke podrške, te je ona prilagođena njegovoj konfiguraciji
 - **aplikacijska programska podrška** (eng. application software)
 - odabire korisnik u skladu sa zadacima i poslovima koje želi obaviti pomoću računala i zbog čega je, konačno, i izvršio nabavku računalne opreme.

Sistemska programska podrška

- **Sistemska programska podrška** obuhvaća:
 - operacijski sustav,
 - programe prevoditelje
 - uslužne programe.

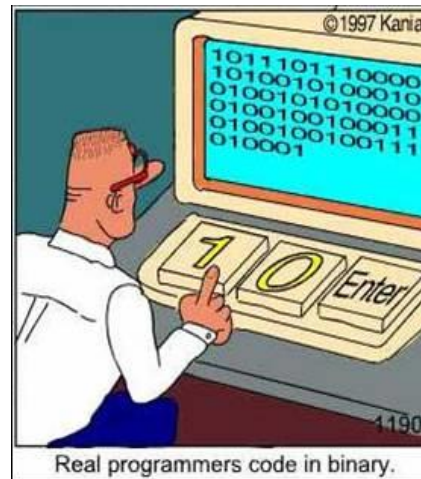
Programski jezici

- **Programski jezik** računala je jezik koji računalno razumije i to su jezici kojima se pišu računalni programi, tj. to je skup sintaktičkih i semantičkih pravila koja definiraju računalni program.
- Postoji mnogo programskih jezika, a svaki od njih ima područje primjene, prednosti i nedostatke.
- **Programski jezici** se mogu podijeliti na:
 - **strojne jezike,**
 - **programski jezike niske razine,**
 - **programski jezike visoke razine.**

- Svaki **programski jezik** ima vlastiti ograničeni skup riječi koje imaju posebna značenja.
 - Takve se riječi nazivaju **ključnim riječima**.
- Za svaki su **programski jezik** propisana pravila slaganja **ključnih riječi** u **instrukcije**.
 - Takva se pravila nazivaju **sintaksa**.
 - Ako se ne zadovolji propisana sintaksa, program će biti neispravan i neće se moći izvršiti.
- **Semantika** programskog jezika označava značenje koje ima u programu neka pravilna konstrukcija programskog jezika

Strojni jezik

- **Strojni jezik** (eng. machine language) je binarni prikaz programa za računalu.
- to je jedini oblik programa koji računalu doslovno "razumije" (nema potrebe ni za kakvim prevođenjem)
- najniža razina programskog jezika
- iskazan je skupovima binarnih brojeva koji sadrže direktne instrukcije za izvođenje operacija na računalu
- stroj razumije takav jezik bez ikakvih tumačenja i točno izvršava takve instrukcije



- Pisanje programa strojnim jezikom je složeno i zahtijeva dobro poznavanje građe računala te se njime bave usko specijalizirani stručnjaci.
 - Pri programiranju je potrebna izuzetna pažnja, velike su mogućnosti pogrešaka i pisanje programa traje vrlo dugo
- Strojni je jezik vezan uz građu računala, odnosno ovisan je o središnjoj jedinici za obradu.
 - Svaki procesor ima sebi svojstven strojni jezik i ne može se koristiti na drugom procesoru, čak ni na različitim procesorima istog proizvođača.
 - rad u strojnom jeziku zahtijeva dobro poznavanje strukture tehničke podrške računala

Programski jezici niske razine

- Simbolički jezici su nastali kako bi ljudima olakšali programiranje jer ljudi lakše pamte simbole nego binarne brojeve.
- Programi pisani simboličkim jezikom su čovjeku čitljiviji i lakši za razumijevanje od binarnog zapisa.
- **Asembler** (eng. assembly language) je **programski jezik niske razine** u kome je svaka binarna instrukcija strojnog jezika predočena odgovarajućim simbolom (kraticom).
- Takvi se simboli lakše pamte i pisanje programa postalo je lakše, brže i sigurnije - mnemonici

- assembler ima istu strukturu i skup instrukcija kao i strojni jezik, ali on dopušta programeru da koristi nazive instrukcija umjesto njihovih brojčanih specifikacija
- svaka instrukcija na strojnom jeziku ima svoj par u assembleru (odnos 1:1)
- Program napisan u assembleru mora biti preveden u binarni oblik da bi ga središnja jedinica za obradu razumjela.
- operaciju prevođenja izvršava poseban program prevoditelj

Programski jezici visoke razine

- Da bi se još više olakšalo programiranje i da bi se isti program mogao izvršavati na različitim računalima (procesorima) stvoren je niz programskih jezika visoke razine.
- Programski jezici visoke razine su svi oni programski jezici kod kojih se veći ili manji skup naredaba strojnog jezika nadomješta jednom instrukcijom višeg programskog jezika.
- Instrukcije ovih jezika mnogo su više nalik govornom jeziku, lakše su za pamćenje i upotrebu.



- Prednosti:
 - nezavisnost jezika o stroju (program se može izvoditi na različitim strojevima)
 - lakoća i razumljivost: čitanja, pisanja i održavanja.
 - instrukcije su čovjeku lagane za razumijevanje
 - slično kodirani engleskom jeziku
- Program napisan **programskim jezikom visoke razine** zove se **izvorni kod** (eng. source code).
- Izvorni kod je moguće pisati u bilo kojem programu za uređivanje teksta (eng. text editor).

- Izvorne kodove treba prevesti u strojni jezik.
- Prevode ih programi koji se nazivaju jezični prevoditelji.
- **Programi prevoditelji** se međusobno razlikuju složenošću i djelotvornošću, a načelno se mogu podijeliti u dvije skupine:
 - **Interpretatori** (eng. interpreter),
 - **Kompilatori** (eng. compiler).
- Danas se uglavnom programi za pisanje izvornog koda objedinjuju u cjelinu sa prevodiocem i poveziivačem (integrirana razvojna okolina, IDE).

Zašto postoje prevoditelji?

- Što ako osoba koja govori samo hrvatskim jezikom želi komunicirati s osobom koja govori samo engleskim jezikom? Postoje dva moguća rješenja problema:
 - Osoba koja govori hrvatskim jezikom zapiše (na hrvatskom) ono što želi reći. Zatim prevoditelj cijeli dokument prevede na engleski jezik i isporuči osobi koja govori engleskim jezikom. Uočimo u ovoj situaciji dvije bitne činjenice:
 - Postoji malo kašnjenje na početku jer hrvatski dokument se u cijelosti mora prevesti prije čitanja na engleskom jeziku.
 - Prevedeni dokument se može čitati bilo kada nakon toga.
 - Osoba koja govori hrvatskim jezikom zaposli prevoditelja koji će prevesti rečenice s hrvatskog jezika u rečenice na engleskom jeziku čim se izgovore. Prevođenje ide sporo zbog pauza između rečenica, originalnih i prevedenih. Uočimo u ovoj situaciji dvije bitne činjenice:
 - Ne postoji kopija prijevoda.
 - Ako se iste rečenice žele ponovno kazati, prevođenje se mora ponoviti.
 - Primijenite analogiju na sljedeće: hrvatski jezik = IZVORNI KOD, engleski jezik = STROJNI KOD

- Čovjeku je stran jezik kojega "razumije" računalo. Zbog toga je bilo potrebno razviti stanovite posredničke mehanizme koji bi bili most između prirodnog jezika čovjeka i jezika računala.
- Najčešće se određenim posredničkim mehanizmima prirodni jezik čovjeka prevodi u odgovarajući jezik računala.
- Jezik računala za razliku od čovjekovog prirodnog jezika jest predstavljen binarnim brojevnim sustavom koji poznaje samo znamenke 0 i 1.

Prevoditelj kompilator

- **Kompilator** je program koji prevodi izvorni kôd programa zapisanog u jednom jeziku visoke razine u ekvivalentni strojni program – program zapisan u strojnom ili objektnom kodu.
 - Kompilator prevodi izvorni kod tako da analizira i prevede cjelokupni izvorni kod odjednom.
 - Ako pronade pogreške (eng. compile-time error), ispisuje poruke i upozorenja o njima.

- **Objektni kôd** (datoteka s ekstenzijom .obj) nije izvršni program i ne može se izravno izvršiti na računalu.
 - Objektni kod je međukorak do izvršnog koda i uz ostalo omogućava uključivanje gotovih dijelova programa iz drugih datoteka.
- Otkrivene pogreške treba ispraviti pa ponovo pokrenuti program za prevođenje.
- Objektni kôd je strojni kôd kojeg računalno može izvršiti, ali da bi postao program mora proći kroz **linker** (povezivač) - program koji povezuje objektnu datoteku s bibliotekama i drugim potrebnim datotekama
 - Takav se oblik programa naziva izvršni oblik (eng. executable - datoteka s ekstenzijom .exe) ili **izvršni program**.

- Kod kompajlera su izvorni kod i izvršni program potpuno odvojeni i pri izvođenju neovisni.
 - Izvršni program se može izvršavati bez postojanja izvornog koda.
- Kod kompilatora su izvorni kod i prevedeni program potpuno odvojeni i pri izvođenju neovisni. Ako se izmjeni izvorni kod, to se neće automatski odraziti na izvedbenom programu, nego ga je potrebno ponovno “kompajlirati”.
- Izvršni se program može izvršavati bez postojanja izvornog koda pa se korisniku najčešće i predaje samo izvršna inačica programa. Takva inačica programa ima oznaku .exe.

- Prednosti kompilatora:
 - brži rad
 - zaštićen izvorni kod
 - uočavanje sintaksnih pogrešaka tijekom “kompajliranja”
- Nedostaci kompilatora:
 - odvojenost prevedenog i izvornog programa.

Prevoditelj interpretator

- Interpretator svaku instrukciju izvornog koda prevodi u strojni oblik u trenutku izvođenja programa, instrukciju po instrukciju.
 - Simbolička instrukcija se prevodi u jednu ili više instrukciji strojnog jezika i zatim se izvrši.
 - Nakon toga se prevede sljedeća simbolička instrukcija i izvrši, i tako redom.
- Prevođenje instrukciju po instrukciju omogućava trenutno otkrivanje određene vrste pogrešaka i interaktivno ispravljanje.
- Nedostaci su relativno sporiji rad i nužnost isporuke izvornoga kôda programa korisniku.
- Izvorni kod je moguće izvršiti samo ako je na računalu prisutan i interpreter.

- Interpretatori prevode i odmah izvršavaju svaku instrukciju višeg programskog jezika. Prevodi se pri svakom izvođenju programa.
- Treba uočiti da je izvorni kod moguće izvršiti samo ako je na računalu prisutan i interpretator koji prevodi instrukcije u strojni oblik.

- Prednosti interpretatora:
 - Prevođenje instrukciju po instrukciju omogućava trenutno otkrivanje i ispravljanje grešaka.
 - Ne troši se vrijeme na “kompajliranje”, odmah se može pokrenuti program.
 - Jednostavniji su.
 - Ne zauzimaju toliko prostora u memoriji za zapis kao kompilatori.
- Nedostaci interpretatora:
 - Prevođenje traje određeno vrijeme, čime se usporava izvršenje programa, što za složenije i opsežne programe može biti primjetno.
 - Sporiji su jer se program svaki puta kada ga želimo izvršiti mora ponovno prevesti interpretatorom.
 - Ne možemo dobiti program u strojnom jeziku.

Izrada mape koncepata predavanja

- Učiti istaknute koncepte
- Uočiti relacije među njima
- Napraviti mapu koncepata

Učiti istaknute koncepte i relacije

- **Programiranje** je postupak kojim se problem kojeg treba riješiti oblikuje tako da ga se može računalno obraditi. Programiranje na najvišoj razini se dijeli na:
 - Definiranje problema (utvrđivanje načina rješavanja i pogodnosti za računalnu obradu)
 - **Rješavanje problema** i njegovo oblikovanje u najprikladnijem programskom jeziku
 - Provjeravanje ispravnosti programa.
- **Računalni program** je, dakle, organizirani skup instrukcija koje se izvode određenim redoslijedom i s određenim ciljem.
- **Instrukcija** (naredba, uputa) je iskaz s kojim se kaže računalu da nešto učini. Sastoji se od operacije koju treba izvesti.
- **Računalni program** je skup **instrukcija** nanizanih strogo utvrđenim redoslijedom, čijim se izvršenjem obavlja željeni posao
- Čovjek koji piše program naziva se **programer**, a postupak nastanka programa naziva se **programiranje**

- **Računalni sustav** se sastoji od **programske podrške** (eng. Software) i **tehničke podrške** (eng. Hardware)
- **Programska podrška** se dijeli u dvije osnovne grupe:
 - **sistemska programska podrška** (eng. system software)
 - **aplikacijska programska podrška** (eng. application software)
- **Sistemska programska podrška** obuhvaća:
 - **operacijski sustav,**
 - **programe prevoditelje**
 - **uslužne programe.**
- **Programski jezik** računala je jezik koji računalo razumije i to su jezici kojima se pišu računalni programi, tj. to je skup sintaktičkih i semantičkih pravila koja definiraju računalni program.
- **Programski jezici** se mogu podijeliti na:
 - **strojne jezike,**
 - **programski jezike niske razine,**
 - **programski jezike visoke razine.**

- Svaki **programski jezik** ima vlastiti ograničeni skup riječi koje imaju posebna značenja. Takve se riječi nazivaju **ključnim riječima**.
- Za svaki su **programski jezik** propisana pravila slaganja **ključnih riječi** u **instrukcije**. Takva se pravila nazivaju **sintaksa**.
- **Semantika** programskog jezika označava značenje koje ima u programu neka pravilna konstrukcija programskog jezika
- **Asembler** (eng. assembly language) je **programski jezik niske razine** u kome je svaka binarna instrukcija strojnog jezika predložena odgovarajućim simbolom (kraticom).
- Program napisan **programskim jezikom visoke razine** zove se **izvorni kod** (eng. source code).
- **Programi prevoditelji** se međusobno razlikuju složenošću i djelotvornošću, a načelno se mogu podijeliti u dvije skupine:
 - **Interpretatori** (eng. interpreter),
 - **Kompilatori** (eng. compiler).

Istaknuti koncepti

- Programiranje
- Rješavanje problema
- Računalni program
- Instrukcija
- Programer
- Računalni sustav
- Programska podrška
- Tehnička podrška
- Sistemska programska podrška
- Aplikacijska programska podrška
- Operacijski sustav
- Program prevoditelj
- Uslužni program
- Programski jezik
- Strojni jezik
- Programski jezik niske razine
- Programski jezik visoke razine
- Ključna riječ
- Sintaksa
- Semantika
- Asembler
- Izvorni kod
- Interpretator
- Kompilator

Mapa konceptata

