

# VPRAŠANJA

- Kaj je program?
  - je niz logično povezanih ukazov, inštrukcij, ki računalniku zapovedujejo, kako naj obdela določene podatke, da bo rešil neki problem.
- Kaj je programiranje?
  - Je pisanje nizov logično povezanih ukazov, inštrukcij, ki računalniku napovedujejo, kako naj obdela določene podatke, da bo rešil neki problem.

# VPRAŠANJA

- Naštej stopnje med programiranjem?
  - Opredeljimo nalogo
  - razčlenimo in najdemo rešitev
  - preverjeno zapišemo v programskem jeziku
  - dodatna preverjanja, pregled in testiranje
  - izdelava dokumentacije

# VPRAŠANJA

- Kaj je to strukturirano programiranje?
  - opredelitev problema (postopek čisto jasen kako nalogo rešimo)
  - razgrajevanje problema na niz zaporednih majhnih korakov – postopkov (neodvisnih problemov)
  - sestavljanje razgrajenih problemov nazaj v celoto in preverjanje, če je problem ustrezen

# VPRAŠANJA

- Prednosti razgrajevanja?
  - omogočajo enostavnejšo definicijo glavne naloge oz. problema
  - vsak posamezen korak se lahko zapiše in testira samostojno
  - postopki oz. podproblemi se lahko v glavnem poteku programa uporabljajo večkrat

# VPRAŠANJA

- Kaj je to algoritem?
  - Algoritem je postopek, opremljen z nizom navodil za reševanje naloge. Biti mora nedvoumen ustavljiv in zapisan z elementarnimi navodili.
- Kdaj je algoritem nedvoumen?
  - Nedvoumen je takrat, ko je izvajanje navodil v vseh okoliščinah natančno določeno.
- Kdaj je algoritem ustavljiv?
  - Ustavljiv je, če se v vseh okoliščinah konča v merljivem času.
- Kakšna so elementarna navodila?
  - Elementarno navodilo je tisto, ki ga oblikovalec algoritma razume in zna izpolniti.

# VPRAŠANJA

- Na kakšen način lahko zapišemo algoritem?
  - z besednim opisom
  - z grafično predstavitvijo
  - s psevdokodom
  - s programskim jezikom
- Iz česa je sestavljen vsak računalniški program?
  - iz algoritma in
  - podatkov

# VPRAŠANJA

- Kaj je to spremenljivka?
  - spremenljivka je sestavljena iz dveh delov
    - ime spremenljivke
    - podatek v spremenljivki
  - ime spremenljivke je konstantno čez celoten algoritem
  - vsebina spremenljivke se z izvajanjem algoritma spreminja
  - sprememba – sprememba vsebina in iz tukaj izvira ime spremenljivka

# VPRAŠANJA

- Vrednost spremenljivk na začetku dela so?
  - Na začetku dela so vrednosti spremenljivk nedoločene, zato jim moramo (vsaj enkrat) vrednosti predhodno prirediti.
- Zakaj raje uporabljamo grafični zapis algoritma?
  - besedni zapis algoritma je nepregleden, zato uporabimo grafični zapis z dogovorjenimi bloki – diagram poteka.

# VPRAŠANJA

- Kaj je to strojni jezik?
  - To je jezik, katerega razume računalnik
- Ali lahko programiramo v strojnem jeziku?
  - DA!
- Kako zgleda programiranje v strojnem jeziku?
  - Pisanje ničel in enic
- Ali je strojni jezik pomagal širiti uporabo računalnikov?
  - Strojni jezik je kvečjemu oteževal širjenje uporabe računalnikov, ker je bil zelo kompliciran.

# VPRAŠANJA

- Razlogi za oteževanje širjenja uporabe računalnikov pri uporabi strojnega jezika?
  - pisanje programov z zaporedjem ničel in enic je nepregledno in zato na tak način lahko rešujemo le najpreprostejše probleme
  - strojni jezik se razlikuje glede na vrsto računalnikov, kar pomeni nezdržljivost (neprilagojenost, medsebojno neusklajenost, neprenosljivost) izdelanih programov.

# VPRAŠANJA

- Kako je sestavljen strojni jezik?
  - Sestavlja ga niz strojnih inštrukcij, a vsaka inštrukcija sestoji iz niza dvojiških številk določene dolžine. Ta niz se deli na dva dela:
    - operacijskega in
    - naslovnega
- Kaj predstavlja operacijski del inštrukcije?
  - Operacijski del inštrukcije predstavlja kodirano obliko določene elementarne operacije.
- Kaj predstavlja naslovni del inštrukcije?
  - naslovni del predstavlja kodirano obliko efektivnih naslovov v spominu in/ali registrih , na katerih vsebinah se želi opravljati določena elementarna operacija

# VPRAŠANJA

- V katere dve skupine delimo simbolične programske jezike?
  - nižje programske jezike in
  - višje programske jezike
- Kakšna je razlika med nižjimi in višjimi programskimi jeziki?
  - Nižji programski jeziki so bližje računalniškemu jeziku in bolj oddaljeni človeku
  - Višji programski jeziki so pa bolj oddaljeni računalniškemu jeziku in bližje človeku

# VPRAŠANJA

- Kateri programski jezik je značilen za nižje programske jezike?
  - Zbirni jezik – assembly language
- Kakšna je razlika med strojnim jezikom in zbirnim jezikom?
  - simbolni jezik uporablja ničle in enice za predstavitev operacijskega in naslovnega dela
  - zbirni jezik uporablja simbole za predstavitev operacijskega in naslovnega dela
- Kaj potrebujemo da program izdelan v zbirnem jeziku prevedemo v strojni jezik?
  - prevajalnik tako imenovani zbirnik - assembler

# VPRAŠANJA

- Ali je zbirni jezik enak za vse tipe računalnikov?
  - Zbirnik je ravno tako, kot strojni jezik prilagojen samo za določeni tip računalnikov in ni kompatibilen eden z drugim
- Kakšna je relacija med simboli v zbirnem jeziku in ukazi v strojnem jeziku?
  - Relacija med simboli v zbirnem jeziku in ukazi v strojnem jeziku je 1:1, torej en simbol v zbirnem jeziku pomeni en ukaz v strojnem jeziku

# VPRAŠANJA

- Prednosti in slabosti zbirnega jezika?
  - veliko možnosti za natančen nadzor nad uporabo
  - hitro izvajanje programov izdelanih v zbirnem jeziku
  - neprenosljivost
  - težave pri usvajanju programskega jezika
- Kaj so to makroji?
  - Makroji so okrajšave za standardne nize ukazov; programerjem olajšujejo delo v zbirnem jeziku tam, kjer bi se moral niz kakih ukazov ponoviti na več mestih programa. Makroukaz se vključi v program v zbirniku skupaj z normalnimi ukazi, pri čemer jih prevajalnik za zbirni jezik prevede v niz normalnih ukazov.

# VPRAŠANJA

- Kakšna je relacija med višjimi programskimi jeziki in strojnim jezikom?
  - Višji programski jeziki so problemsko usmirjeni , navadno so tudi prirejeni posebnim področjem. Vsebujejo pomensko simbolične ukaze, kateri se prevedejo v več strojnih ukazov. Torej relacija ni več 1:1 (kot zbirnik : strojni jezik), ampak 1:n (1 simbolični ukaz : več strojnih ukazov)

# VPRAŠANJA

- Kateri načini se uporabljajo za pretvorbo višjega programskega jezika v strojni jezik?
  - prevajalnik (compiler)
  - tolmač (interpreter)
- Kaj je to prevajanje?
  - za prevajanje se uporablja poseben program prevajalnik (compiler), ki celoten program prevede v strojni jezik. Tak način uporablja večina “klasičnih” jezikov

# VPRAŠANJA

- Kaj je to tolmačenje?
  - uporablja se poseben program tolmač (interpreter), ki pri izvajanju programa sproti pregleduje, ureja, javlja napake in prevaja oziroma tolmači posamezne stavke programa.
- Kakšna je hitrost izvajanja programov izdelanih z višjim programskim jezikom, ki so prevedeni v strojni jezik in programov, ki se interpretirajo?
  - Programi, ki se tolmačijo (interpretirajo) so navadno do 10x počasnejši od programov, ki so prevedeni v strojni jezik.

# VPRAŠANJA

- Opiši programski jezik Fortran!
  - (FORmula TRANslation) je nastal okrog leta 1955 kot program za prevajanje matematičnih postopkov v računalniške programe. Sprva zelo enostavni verziji so se kmalu pridružile izpeljanje, kot npr. fortran II, III, IV, V, fortran 77, fortran 90 in cela vrsta strukturiranih fortranov. Uporablja se predvsem za reševanje znanstvenih in tehničnih problemov.

# VPRAŠANJA

- Opiši programski jezik Cobol!
  - (COmmon Business Oriented Language) je nastal okrog leta 1960, vpeljan pa je bil sedem let kasneje. Namenjen je predvsem pisanju programov za poslovno-ekonomsko problematiko. Strukturiran je tako, da skuša biti čim bolj podoben naravnemu (angleškemu) jeziku in razumljiv tudi neprogramerjem, zlasti vodstvenim delavcem.

# VPRAŠANJA

- Opiši programski jezik BASIC!
  - (Beginner's All-purpose Symbolic Instruction Code) je nastal leta 1965 kot preprost jezik za učenje programiranja in kot stopnjička k fortranu. Zasnovali so ga kot jezik, ki bo primeren za najrazličnejše uporabnike: začetnike, študente pa tudi strokovnjake različnih strok. Njegova uporaba se je še posebno razmahnila, ko so se uveljavili hišni in osebni računalniki in to zlasti v šolstvu, saj je uporaba tolmača z že ugrajenim basicom (v ROMu) za začetno delo, posebno v osnovnih šolah, izredno ugodna. Čeprav basicu očitamo, da npr. nima dovolj preglednih programov, pa ta programski jezik zagotovo pozna največ ljudi.

# VPRAŠANJA

- Opiši programski jezik pascal!
  - je nastal leta 1968, prvi prevajalnik pa dve leti kasneje. Njegov "oče" Niklaus Wirth; imenoval ga je po francoskem matematiku Blaisu Pascalu, ki je izumil enega prvih praktično uporabnih računalniških strojev. Za osnovo sta mu rabila ALGOL-60 in ALGOL-W. Wirth je izpustil večino težjih, manj uporabnih lastnosti algola, dodal pa je nekaj koristnih elementov. Za to je imel dva razloga: želel je napraviti jezik, ki bo imel učinkovit prevajalnik, uporabniki pa bi lahko pri pisanju programov izkoriščali polno moč jezika. Pascal se je hitro razširil po svetu in v nekaj letih postal eden pomembnejših programskih jezikov: je šolski primer za lepo izdelan in strukturiranemu programiranju naklonjen programski jezik. Ima vrsto koristnih prijemov, ki jih najdemo tudi pri drugih programskih jezikih in programerju naravnost ponuja možnost, da piše programe, kar se da strukturirano.

# VPRAŠANJA

- Opiši programski jezik C!
  - je zelo sposoben višji programski jezik (avtor Dennis Ritchie), ki se je razvil ob operacijskem sistemu UNIX. Večina sistemskih programov drugih operacijskih sistemov je namreč napisanih v nižjih programskih jezikih (zbirnem jeziku). Programski jezik c je posebno primeren za pisanje operacijskih sistemov in sistemskih servisnih programov.

# VPRAŠANJA

- Koliko višjih programskih jezikov poznamo?
  - Več kot 100!
- Kako delimo višje programske jezike
  - na jezike, s katerimi predvsem opisujemo kaj želimo narediti (nepostopkovni jeziki) in
  - na jezike, s katerimi tudi podrobneje povemo, kako to storiti (postopkovni jeziki)

# VPRAŠANJA

- Opiši programski jezik PROLOG!
  - (PROgramming in LOGic) – je bil prvi poskus oblikovanja jezika, ki naj bi programerjem omogočal definiranje rešitev s pomočjo logike. Zasnovali so ga že leta 1975 na Univerzi v Marseillu. To je preprost, vendar presenetljivo močan programski jezik, ki se vedno bolj uveljavlja in bo kmalu na široko prodril med uporabnike. Posebno učinkovit je pri simboličnem, nenumeričnem procesiranju in pri delu z bogatimi podatkovnimi strukturami.

# VPRAŠANJA

- Kakšne vrste napak poznaš in vsako posebej opiši!
- Kako posamezne vrste napak odpravljamo?
- Kaj je to testiranje programa in kako ga izvajamo?
- Kaj zavzema dokumentacija programa?

# VPRAŠANJA

- Izdelaj algoritem za množenje dveh celih števil
- Izdelaj algoritem za deljenje dveh celih števil
- Izdelaj algoritem, za ostanek dveh celih števil
- Izdelaj algoritem s katerim izračunamo vsoto sodih števil od N do M (N in M vnesemo preko tipkovnice in so pozitivna cela števila in predpostavimo da je N manjši od M).
- Izdelaj algoritem s katerim izračunamo vsoto lihih števil od N do M (N in M vnesemo preko tipkovnice in so pozitivna cela števila in predpostavimo da je N manjši od M).

# VPRAŠANJA

- Izdelaj algoritem, ki iz prebranega števila preko tipkovnice izpiše število števk.
- Izdelaj algoritem, ki iz prebranega števila preko tipkovnice izpiše vsoto števk.
- Izdelaj algoritem, ki iz prebranega števila preko tipkovnice najde največjo števko.
- Izdelaj algoritem, ki iz prebranega števila preko tipkovnice najde najmanjšo števko.
- Izdelaj algoritem, ki izračuna absolutno vrednost števil
- Izdelaj algoritem, ki pretvori prebrano desetiško število v dvojiški sistem

# VPRAŠANJA

- Izdelaj algoritem, ki med tremi števili poišče največje število
- Izdelaj algoritem, ki med 10 prebranimi števili izpisuje samo pozitivna števila
- Izdelaj diagram poteka, ki za prebrano število iz tipkovnice izpiše vse njegove deljitelje.
- Izdelaj algoritem, ki zamenja vrednosti dveh prebranih števil iz tipkovnice med seboj.

# VPRAŠANJA

- Napišite algoritem, ki iz tipkovnice prebere poljubno število in glede na prebrano vrednost na zaslon izpiše trikotnik oblike (primer če vnesemo število 5):

\*1

\*\*2

\*\*\*3

\*\*\*\*4

\*\*\*\*\*5