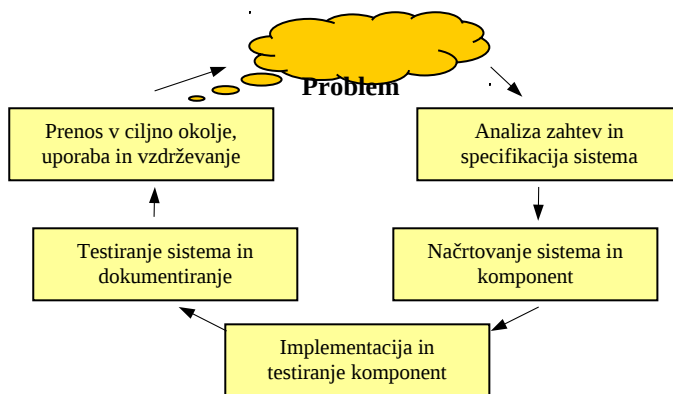


Uvod v programske jezike

Faze življenjskega cikla programske opreme

Življenjski cikel programske opreme delimo na naslednje faze

- Analiza zahtev in specifikacija sistema.
- Načrtovanje sistema in komponent.
- Implementacija in testiranje komponent.
- Testiranje sistema in dokumentiranje.
- Prenos v ciljno okolje, uporaba in vzdrževanje.



Analiza zahtev in specifikacija sistema

V tej fazi razvijalci in uporabniki dobijo odgovor na vprašanje: KAJ POTREBUJEMO oz. kaj naj bi zgrajeni programski sistem zagotavljal.

Ta faza vključuje:

- analiziranje programskega problema (funkcionalen opis) in opis produkta. Rezultat tega dela je popolno razumevanje problemskega področja;
- specifikacije želenega obnašanja grajenega sistema (funkcionalne zahteve in specifikacije). Rezultat tega dela je celovit dokument programskih specifikacij (SZPO).

Obe aktivnosti se izvajata sočasno. SZPO opisuje funkcionalne zahteve, značilnosti strojnega okolja, osnovno obliko uporabniških vmesnikov in performančne cilje oz. zahtevane zmogljivosti.

Načrtovanje sistema in komponent

V tej fazi odgovorimo na vprašanje: KAKO NAREDITI oz. kako bomo zadovoljili identificirane zahteve -obnašanje sistema.

Ta faza vključuje

- preliminarno načrtovanje - izvede se dekompozicija (razčlenitev) programskega sistema na njegove dejanske konsistentne komponente in interaktivna razgradnja teh komponent v vedno manjše podkomponente, dokler niso dovolj majhne, da jih lahko ljudje brez težav razumejo; vsak modul je dokumentiran - opisani so vhodi, izhodi in funkcije preoblikovanja podatkov;
- podrobno načrtovanje - za vsak modul definiramo in dokumentiramo algoritme.

Kot rezultat te faze je narejena:

- modularna razgradnja programskega sistema,
- definicija strukture podatkov,
- definicija formata datotek in
- opis pomembnejših algoritmov.

Implementacija in testiranje komponent sistema

V tej fazi se izvede kodiranje algoritmov. Algoritmi se transformirajo v računalniku razumljiv jezik (praviloma se uporabljajo višji programski jeziki). Testirajo in čistijo se napake vsakega modula, specificiranega pri načrtovanju.

Po potrebi se v tej fazi tudi kreira in inicializira podatkovna baza.

Testiranje sistema

Pri testiranju posameznih modulov se osredotočimo na del programa, da lažje ugotovimo in odstranimo napake. Hkrati kontroliramo tudi obnašanje modula glede na podane specifikacije (funkcionalno testiranje).

V tej fazi že preverjene module integriramo oz. povežemo (združimo) v enotno programsko strukturo ter jih testiramo kot celoto. Sledi še sistemsko testiranje, s katerim preverimo, ali se celoten programski sistem, postavljen v določeno strojno okolje, obnaša ustrezno podanim specifikacijam zahtev programske opreme.

Prenos v ciljno okolje, uporaba in vzdrževanje

Programska oprema in pripadajoča dokumentacija se izroči uporabnikom. Začne se uporaba sistema. V primeru napak ali pomanjkljivosti, se pripravijo in implementirajo ustrezni popravki in nadgradnje sistema.

Pomen in vloga programskih jezikov

Programski jezik je skupek pravil, podatkovnih struktur in programerskih vmesnikov, ki programerju služijo kot orodje za izdelavo programa. Programski jeziki se uporabljajo zato, ker sta človekova in računalnikova »govorica« tako različni, da je pisati programje v jeziku računalnika (strojni kodi) nepregledno in nasploh mukotrpno.

Naloga programskih jezikov je omogočiti opis problema in opis postopka za njegovo rešitev

- Opis problema: opis izhodiščnih podatkov, rezultatov in relacij med njimi
- Opis postopka: opis zaporedja korakov (algoritma), ki ustreza relaciji med podatki in rezultati in ki nas pripelje do rešitve problema.

Računalniško programiranje je dejavnost, katere cilj je ustvarjanje novih računalniških programov ali njihovih sestavnih delov na temelju določenih pravil. Računalniški program je algoritem napisan v izvorni kodi določenega programskega jezika, pri delu pa si programer lahko pomaga z ustreznimi razvojnimi orodji.

Z uporabo različnih prevajalnikov lahko program v enem od programskih jezikov prevedemo v strojno kodo, to je v obliko, ki jo lahko računalnik oziroma njegov procesor izvede.

Strojna koda programov je vedno specializirana za izvajanje na določeni arhitekturi, ki je skupek lastnosti strojne in programske opreme. Strojna koda, ki jo programi vsebujejo, izvaja ukaze preko operacijskega sistema, ki deluje kot vmesnik med strojno in programsko opremo.

Izvršljive programske datoteke v okolju Microsoft Windows imajo končnico .EXE, ali pa .DLL, v DOS-u pa tudi .COM. Pri operacijskem sistemu LINUX izvršljive datoteke nimajo karakteristične končnice, pač pa je to razvidno iz lastnosti oziroma dovoljenj datotek.

Izbor programskega jezika je v veliki meri odvisen od tega, kaj nameravamo z njim početi. Tako se pri računsko intenzivnih programih še danes pogosto uporablja eden prvih programskih jezikov – FORTRAN (ime je skovano iz angleških besed Formula Translation). Za sistemsko programiranje (morda celo pisanje operacijskih sistemov) smo včasih uporabljali zbirni jezik, danes pa predvsem C, C++ ali C#. Za pisanje programa, ki bo neodvisen od operacijskega sistema in tudi od vrste strojne opreme, bomo uporabili programski jezik Java. Če pa se ukvarjamo z računalniško administracijo, bomo npr. uporabljali primeren skriptni jezik, na primer "bash" v primeru operacijskega sistema LINUX.

Delno je izbira programskega jezika pogojena tudi z našim predznanjem. Pravemu programerju je pravzaprav vseeno, v kakšnem jeziku bo programiral. Glede na to, da je programskih jezikov na stotine, pa je včasih odločitev za primeren jezik kar težavna.

Za učenje programiranja je pomembno da je jezik dovolj »čist« in preprost, da omogoča uporabo sodobnih programerskih prijemov, kot je na primer strukturirano in objektno usmerjeno programiranje, da so njegova orodja prosto dostopna in da omogoča tudi delo z grafiko. Vse naštetu velja tudi za jezik C# or. Visual C#, ki ga bomo uporabljali pri predmetu Programiranje I in Programiranje II.

Snovanje programov vsebuje elemente znanosti, umetnosti, matematike in tehnike.

Nekatere delitve programskih jezikov

Programski jezik se delijo na nizkonivojske, ti so bližje strojnemu jeziku in visokonivojske, ki pa so bližje naravnim jezikom.

C#, C++, C, Basic, Python in Java so na primer visokonivojski jeziki. Razlika med njimi je predvsem v tem, da moramo programe v jezikih C#, C++, C in Java pred izvajanjem prevesti, Python pa sodi med tako imenovane skriptne jezike (sem spadajo še PHP, Java Script, Bash, ...), ki potrebujejo za svoje izvajanje interpreter oziroma tolmač. Dobra stran skriptnih jezikov je predvsem to, da posamezne ukaze sproti tipkamo, računalnik jih z interpreterjem prevede in takoj izvede. Prav za vse skriptne jezike pa velja, da lahko zaporedja ukazov zapišemo kar v preprosto tekstovno (ASCII) datoteko, ki jo nato interpreter bere, razpozna in izvaja vrstico za vrstico. Slaba stran interpreterja pa je, da izvajanje programa postane precej počasnejše, saj mora interpreter vsako vrstico znova prevesti, četudi jo ponavlja, pa tudi dostop do programskih spremenljivk je počasnejši.

Na drugi strani pa je zbirni jezik (zbirnik), ki je najbližje strojnemu jeziku. Ta je večinoma izredno težko razumljiv človeku in je potrebno veliko učenja, de se ga navadi. Navadno je to skupek števil in črk (črke so le interpretacija števil). Edini razlog za njegovo uporabo, je (bila) njegova možnost, da z njim procesor uporabimo optimalno.

Avtorske pravice

Programne lahko izdelamo sami, vendar so mnogi programi tako obsežni, da je bolje, če jih kar kupimo. Z nakupom pa ne pridobimo lastniških pravic do programa, temveč le dovoljenje - licenco, ki nam daje pravico uporabe programa na enem računalniku. Z novo različico - verzijo programa kasneje že kupljeni program nadgradimo (*upgrade*) po ugodnejši ceni.

Programi	Uporaba
Javni (Public)	izdelek je predan v javno last (zastonj)
Prosti (Free)	izdelek je na voljo zastonj
Poskusni (Share)	oblika trženja brez posrednikov (nekaj 10 USD)
Tržni (Commercial)	podjetja sama ali s posredniki tržijo izdelke (več kot 50 USD)

Vsak avtor računalniškega programa ima samodejno avtorske pravice nad programom in ima možnost omejiti pravice kopiranja, uporabe ali spreminjanja svojega programa. Uporabnik lahko pridobi pravico do uporabe programa v skladu z dovoljenji avtorja. Licenca točno določa, kaj uporabnik lahko počne s programsko opremo, kakšne pogoje mora izpolnjevati in kakšne so njegove pravice. Glavni pogoj pa je seveda plačilo kupnine za programsko opremo.

Literatura: <http://colos.fri.uni-lj.si/ERI/INFORMATIKA/index.html>