

Alati za razvoj softvera



PREDMET: RAZVOJ SOFTVERA 2
PROFESOR: VLADIMIR FILIPOVIĆ
STUDENT: MARIJA ŠEVKOVIĆ

Sadržaj



- Alati za dizajniranje (Design Tools)
- Alati za manipulaciju izvornog koda (Source-Code Tools)
- Alati za manipulaciju izvršnog koda (Executable-Code Tools)
- Alatno-orijentisana okruženja (Tool-oriented Environments)
- Izrada razvojnih alata
- Tool Fantasyland

Alati za dizajniranje



- Sastoje se uglavnom od grafičkih alata za crtanje dijagrama
 - Za realizaciju dizajna koriste standardne grafičke notacije:
 - UML
 - ER diagrams
 - Architecture block diagrams
 - Hierarchy charts, etc...
 - Sve što se može nacrtati ovim alatima, može se i ručno nacrtati
 - Alati dosta pojednostavljuju stvari, pogotovu kod prepravki dizajna
 - Neki od alata mogu izvesti kod direktno iz dizajna
-
- ❖ Visual Paradigm
 - ❖ MySQL Workbench

Alati za manipulaciju izvornog koda



- Editovanje izvornog koda
- Analiza kvaliteta
- Refaktorisanje
- Kontrola verzija

Editovanje izvornog koda



- **Integrirana razvojna okruženja (IDEs)**
 - Prednosti:
 - Kompilacija i detekcija grešaka iz editora
 - Dobar pregled strukture programa
 - Brza pretraga klase, paketa i ostalih delova projekta
 - Formatiranje koda
 - Napredna pretraga (regularni izrazi, memory strings)
 - Search and replace kroz fajlove
 -
 - ❖ Eclipse
 - ❖ Visual Studio
 - ❖ IntelliJ IDEA
 - ❖ MonoDevelop

Editovanje izvornog koda



- **Diff Tools**

- Ova funkcionalnost je obično deo sistema za kontrolu verzija
- Pravi poredjenje trenutne verzije fajla sa poslednjom verzijom fajla na kojoj je radjeno i označava razlike

- **Merge Tools**

1. Zaključava resurs tako da samo jedna osoba može da modifikuje fajl u odredjenom trenutku
2. Dozvoljava da više ljudi simultano radi sa fajlom. Jednostavna spajanja se vrše automatski, dok konfliktna mora da razreši korisnik

Editovanje izvornog koda



- **Source-Code Beautifiers**
 - Formatiraju kod tako da izgleda konzistentno
 - Dva tipa formatiranja:
 1. Formatiranje bez menjanja originalnog koda
 2. Formatiranje koje menja kod (formatiranje liste parametara) korisno kada se radi sa legacy code
- **Interface Documentation Tools**
 - Izdvajaju dokumentaciju iz koda
 - JavaDoc

Editovanje izvornog koda



- **Templates**
 - Podstiču konzistentan stil kodiranja i pravljenja dokumentacije
- **Cross-Reference Tools**
 - Izlistavaju varijable i sva mesta na kojima se koriste(web strane)
- **Class Hierarchy Generators**
 - Proizvode informacije o stablu nasledjivanja
 - Korisni za analizu strukture programa
 - Dostupni u nekim IDEs

Analiza kvaliteta



- **Sintaksni i semantički analizatori**
 - Detaljnije proučavaju kod od kompajlera
 - `while(i = 0)` , `while(i == 0)`
- **Metrički analizatori**
 - Analiziraju i ocenjuju kvalitet koda
 - Prate anomalije i korekcije koda i povezuju ih sa programerima

Refaktorisiranje



- **Refaktori**
 - Refaktorišu kod kao odvojeni programi ili integrisani u neki IDE
 - Promene koda su brze i manje podložne greškama
- **Rekonstruktori**
 - Eliminišu goto naredbe iz špageti koda
- **Prevodioci koda**
 - Prevode kod na drugi jezik

Kontrola verzija



- **Version Control System (VCS)**
 - Sistem za praćenje promena(revizija) nad fajlovima u projektu
 - Mogućnost vraćanja fajla na prethodnu reviziju
- ❖ GIT
- ❖ SVN
- ❖ Mercurial

Alati za manipulaciju izvršnog koda



- **Kompajleri i linkeri**
 - Kompajleri prevode izvorni kod u izvršni
 - Linkeri povezuju objektne fajlove generisane od strane kompajlera sa kodom potrebnim da se napravi izvršni fajl
 - Linkeri mogu da naprave izvršne fajlove koji učitavaju samo deo sebe u memoriju dok je ostalo na disku dok ne zatreba
- **Make**
 - Vezuje se za Unix i C,C++ jezike
 - Sadrži uputstva za kompilaciju programa

Alati za manipulaciju izvršnog koda



- **Biblioteke**
 - Omogućavaju pisanje kvalitetnog koda za kratko vreme
 - Mogu se naći biblioteke raznih funkcionalnosti:
 - ✦ Kompresija podataka
 - ✦ Matematičke operacije
 - ✦ Obrada slika
 - ✦ Sigurnost i enkripcija , ...
- **Code Generation Wizards**
 - Koriste se najviše kod aplikacija za rad sa bazama podataka
 - Korisni su i za pravljenje prototipova koda
- **Debugovanje i testiranje koda**

Alati za manipulaciju izvršnog koda



- **Profajleri**
 - Analiziraju kod dok se izvršava
 - Za svaku stavku se zna koliko puta se izvršila i koliko vremena se izvršavala
 - Daju uvid u kritične tačke programa
- **Assembler Listings and Disassemblers**
 - Neki kompajleri proizvode assembler listings
 - Disassemblers rekonstruišu asemblerski kod iz mašinskog
 - Asemblerski kod nam govori koliko efikasno kompajler prevodi kod višeg programskog jezika u mašinski kod

Alatno-orijentisana okruženja



- Vezuju se za Unix
- Unix okruženje je poznato po kolekciji malih alata koji zajedno dobro rade i lako se mogu sastavljati da izvršavaju složenije funkcije
 - Grep
 - Diff
 - Sort
 - Make
 - Tar, ...

Izrada razvojnih alata



- Skoro svaka veća kompanija(preko 1000 programera) ima svoje alate i grupe za podršku
- Alati specifični za projekat
 - Veći projekti zahtevaju alate jedinstvene za njih:
 - ✦ Generisanje specijalnih testova
 - ✦ Simulacija hardvera koji nije dostupan
- Skripte
 - Alati koji izvršavaju listu naredbi sekvencijalno
 - Obično su to compile-link sekvence ili bilo koje komande sa puno parametara

Tool Fantasyland



- U prethodnim decenijama pojavljivali su se alati koji su navodno eliminisali potrebu za programiranjem
 - Jezici 3.i 4. generacije
 - Automatsko programiranje
 - Vizuelno programiranje
- Nijedan nije uspeo

Zaključak



- Dobri alati mogu vam olakšati život
- Možete napraviti alate za specifične svrhe
- Dobri alati mogu olakšati naporne aspekte programiranja ali ne mogu eliminisati potrebu za programiranjem, mada mogu menjati pojam programiranja