

Ohjelmointi 1 / syksy 2007

10/20: IDE

Paavo Nieminen

nieminen@jyu.fi

Tietotekniikan laitos

Informaatioteknologian tiedekunta

Jyväskylän yliopisto

Tämän luennon rakenne

- Täydennystä aiempaan: parempi esimerkki huonosti toimivasta konsolisyytöstä.
- Mikä on IDE eli Integrated Development Environment
- Eclipse on yksi Javalle soveltuva IDE. Sitä opitaan käyttämään tällä kurssilla.

Konsolisytteen lukeminen: ongelma jos ei lueta koko riviä.

→ Katsotaan vielä Arvauspeli -ohjelmaa. Viimeksi yritettiin demonstroida virheellistä syötettä, mutta ohjelma tuntui toimivan. Katsotaan toisenlaista versiota, jossa ongelma ihan oikeasti ilmenee. Käyttäjä voi syöttää useita lukuja ennen kuin painaa enteriä! Scanner puolestaan lukee syöttövirrasta aina tyhjämerkkiin asti eikä pidemmälle. Huomioitava, tai toiminta voi vähintään näyttää oudolle tai joissain tilanteissa aiheuttaa jopa vaaranpaikkoja.

Luettakoon mielummin merkkirivi kerrallaan
(`java.util.Scanner.nextLine()`), ja tutkittakoon syötteen
oikeellisuus.

Mikä on IDE

IDE (eli Integrated Development Environment) eli ”Yhdistetty sovelluskehitysympäristö” tarjoaa samassa käyttöliittymässä yleensä seuraavat:

- Graafinen näkymä ohjelmiston osioihin
- Tekstieditori, jossa on Runsas tuki ohjelmointikielen syntaksille, semantiikalle ja apukirjastojen käytölle.
- Ajoympäristö testaamista varten
- Debuggeri koodin toiminnan tutkimista varten (TÄRKEÄ).
- Automaatioita, mm. koodipätkien ja kommenttien generointi ”nappia painamalla”, refaktorointi (HYÖDYLLISIÄ)
- Tukityökaluja, mm. versionhallinta. (ELINTÄRKEITÄ ISOISSA projekteissa)
- Plug-in -mahdollisuus. Mitä ikinä voi tarvita, on liitettävissä mukaan IDEen, jos sitä voi laajentaa lennosta.

IDEjä on erittäin paljon erilaisille ohjelmointikielille. Javallekin on useita, nimiltään mm. Eclipse, JBuilder, NetBeans, ... Valitsemme ohj1:llä Eclipsen.

Miksi IDE eikä pelkkä tekstieditori?

Miksi IDEä kannattaa käyttää:

- IDE helpottaa ja nopeuttaa ohjelmointia.
- Mahdollistaa isompien ja virheettömämpien ohjelmistojen tekemisen helpommin.
- Tekee automaattisesti mekaaniset ja työläät (eli ”tylsät”) ohjelmointivaiheet (joissa myös tulee helposti turhia virheitä)

Miksi emme ole vielä käyttäneet IDEä:

- IDE automatisoi toimintoja ja piilottaa yksityiskohtia graafisten rakenteiden alle.
- Alla oleva koneisto (tiedostorakenne, kääntäjä, virtuaalikone) täytyy ymmärtää ennen kuin on järkeä käyttää automaatiota, joka piilottaa sen.
- Ohjelmoinnin taito ei saa riippua IDEn klikkailun taidosta!
- Ohjelmointikyky ei saa rajoittua vain kieliin, joille on IDE; se on luksusta, johon ei aina ole mahdollisuutta!
- IDEn tarjoama välineistö on kova homma oppia jo itsessään, koska se on aika laaja.

Tilannekatsaus taakse ja eteen

- 50 % ohjelmointi 1:stä tehtiin ilman IDEä.
- Toivottavasti on ehditty oppia jotakin tietokoneen käskemisen ruohonjuuritasosta.
- Toinen 50 % tehdään IDEä hyödyntäen.
- Muistetaan kuitenkin: IDE on vain helppokäyttökerros, jonka alla on jatkuvasti tietokone, käyttöjärjestelmä, tiedostojärjestelmä, kääntäjä, kielen syntaksi ja rakenteiden (aliohjelma, lause, lauseke, operaattori, operandi, muuttuja, . . .) merkitys suorituksen kannalta!
- Näillä puheilla heitetään ruohonjuuritason yläpuolelle helppokäyttökerros (ja yritetään oppia vieläpä käyttämään sitä. . .)

Otetaan Eclipse käyttöön

→ Katsotaan, miltä Eclipse näyttää, mitä näkymiä siinä on, ja mitä kullakin niistä voi tehdä. Suhteutetaan nähtyä siihen, mitä jo ehkä pikkuhiljaa on alkanut hahmottua ohjelmoinnista.

Eclipse IDE:

Ensimmäisiä yksityiskohtia

Käytännön esimerkit seuraavista:

- Workspacen valinta: se on hakemisto, jonka alla sijaitsee mahdollisesti useita projekteja.
- Projektin luonti olemassaolevasta koodihakemistosta
- Java-version valinta (valitaan nyt SE 5 tai 6)
- Koodi-ikkuna ja rakennenäkymä
- Koodauskäytänteet ja IDE: tabulaattorimerkin hallinta, automaattinen muotoilu
- Ajaminen, konsolinäkymä, useat ajot yhtäaikaan
- Koodin tuominen muusta kuin projektin hakemistosta
- Nimien muuttaminen ja muita automaatioita (→ aiempi esimerkki `hirvea_kauhistus.java` korjaantuu kuin itsestään.)
- Miltä workspace näyttää tiedostojärjestelmässä. → HUOMAA KOKO AJAN: Klikkailu on helppoa, mutta se ei ole mystiikkaa, vaan tiedostojen ja tiedostorakenteen hallintaa työkalun avustuksella.