

AlgDat 12

Forelesning 2

Forrige forelesning

- Følg med på *hiof.no/algdat*, ikke minst beskjedsida!
- Algdat: *Fundamentalt*, klassisk, morsomt, ...*krevende* :)
- Pensum: *Forelesningene, oppgavene* (pluss deler av boka)
- Tårnet i Hanoi
 - Illustrerer to måter å lage algoritmer på:
 - *Iterativt*
 - *Rekursivt*
 - Også eksempel på *eksponensiell vekst*, noe som gjør at det ikke er mulig å løse oppgaven med mange ringer
- *Effektiv søking* i mengder er en absolutt forutsetning for de fleste applikasjoner
 - Løses ved varianter av *binærsøk*, som igjen er avhengig av
 - *effektiv sortering*
 - Ineffektiv studentsortering ser ut til å medføre N^2 sammenlikninger (N: antall elementer i mengden)

Denne forelesningen

- Litt mer Google Maps
- Litt mer sortering
- Algoritmer og datastrukturer
- Java repetisjon
- IDE eller teksteditor + kommandolinje?
- Java Collections and Generics

Datastrukturer

- **Definisjon 1:** En datastruktur er den måten en samling data er organisert på.
- **Definisjon 2:** En datastruktur består av en samling elementer, ofte kalt noder, med innbyrdes relasjoner, ofte kalt referanser. Den viktigste oppgaven til en datastruktur er å knytte beslektede elementer sammen på en effektiv og hensiktsmessig måte.
- Tre hovedtyper:
 1. Lineære (en-til-en)
 2. Hierarkiske (en-til-flere)
 3. Nettverk (flere-til-flere)
- Ofte en nær sammenheng mellom datastrukturene og algoritmene

Datastrukturer i hverdagen

Forslag fra studentene:

- ...

Algoritmer

- **Definisjon:** En algoritme er en beskrivelse av hvordan man løser et
 - veldefinert problem med en
 - presist formulert sekvens av
 - et endelig antall
 - enkle, utvetydige og
 - tidsbegrensede steg.
- Som en god fortelling har en algoritme gjerne en
 - innledende del (hode),
 - en gjennomføringsdel (kropp), ofte med en eller flere løkker
 - og en avslutning (hale)

Algoritmer i hverdagen

Forslag fra studentene:

- ...

Programvareutvikling

- En mengde mer eller mindre veldefinerte metoder (software engineering):
 - Fossefall
 - Iterative metoder
 - Smidig utvikling (Agile Development)
 - Extreme programming
 - SCRUM
- Felles for alle: Tre hovedkomponenter:
 1. Analyse
 2. Design
 3. Utvikling (programmering)

Analyse

- Forstå problemet/oppgaven
- Finne og vurdere ulike løsningsstrategier
- Finne og vurdere allerede eksisterende løsninger
- ...selvfølgelig i tett kontakt med oppdragsgiver/prosjekteier

Design

- Skisse av komponenter og data/meldingsflyt (arkitektur)
- Valg av plattform(er)
- Funksjonalitet: Veldefinert beskrivelse av input/output
- Bruk av formelle metoder for beskrivelse, typisk UML

Implementasjon

- Valg av språk
- Koding
- Testing

Hvorfor Java?

- Plattformuavhengig: Kompilert bytekode kan f.eks. kjøres på Linux, Mac m. fl. uten problemer (hvis man passer på å ikke bruke plattformavhengige mekanismer, f.eks. kall på operativsystem-funksjoner)
- OO
- Enkelt
- Oversiktelig?
- Mye ferdig:
 - Datastrukturer med algoritmer
 - Filhåndtering, nettverksoperasjoner
 - Grensesnitt-komponenter

- En mer naturlig / intuitiv / virkelighetsnær måte å programmere på (?)
- Oppmunttrer til modularisering av kode: Oversiktelig, enkelt å bruke: Applikasjoner kan bygges ved å sette sammen ferdige komponenter
- Arv: Eksisterende klasser kan utvides/spesialiseres for nye anvendelser
- Interface:
 - Nye klasser kan implementeres som har samme oppførsel, men som utfører oppgavene på andre måter
 - Dermed kan man bytte ut komponenter i eksisterende applikasjoner, uten å endre selve koden
- Men: Kan paradoksalt nok ble uoversiktelig: Store arvehierarkier kan gjøre det vanskelig å finne ut hva som faktisk skjer, og hvor

Java programmering

- Java dokumentasjon
 - API
 - Tutorial
 - Kode eksempler
- Google:
 - Kodesnutter
 - Hjelpepakker
 - Open source prosjekter

Abstrakte datatyper (ADT)

- Vi kan si at en datatype er en gruppe verdier og operasjonene som er definert for disse verdiene.
- Skal vi kunne gjøre noe med heltall i en datamaskin, må vi ha en datatype som representerer heltall og operasjoner som kan utføres på datatypen.
- Tilsvarende: Skal vi kunne bruke samlingen Set, må vi lage en datatype som representerer samlingen.
- Dette kalles en abstrakt datatype. En abstrakt datatype er en datatype der verdiene og operasjonene i utgangspunktet ikke er definert i programmeringsspråket. Abstrakt fordi detaljer rundt implementasjon bør være skjult for brukeren.
- I Java kan en abstrakt datatype defineres i et grensesnitt (interface). Her beskriver vi *hvilke* operasjoner som skal kunne utføres, men ikke *hvordan* operasjonene skal utføres.

Neste forelesning

- Mer java
 - Lineære strukturer
 - Tabeller
 - Lenkede lister
 - Java Collections
 - Java Generics