

F20 - Sammanfattning

ID1004 Objektorienterad
programmering

Fredrik Kilander fki@kth.se

Viktiga begrepp - klass

- datatyp
- deklaration – i egen fil, inuti en annan klass, anonym
- objekt, instans
- arv, datatyp hierarkin
- abstract – inkomplett, mall för subklasser
- interface, specificerar metoder, men inte hur

Viktiga begrepp - metoder

- metoder bor i en klass
- metoder har en signatur
 - returtyp
 - namn
 - parameterlista
- överlagring (overload)
 - `void foo(int i)`
 - `void foo(double d)`
- åsidosättning (override)
- polymorfism (hänger ihop med override/arv)

Viktiga begrepp – variabler(fields)

- Deklareras i en klass, "klassvariabler"
- Deklareras i ett block {}, lokala variabler (inklusive parametrar)
- variabler måste ha en datatyp (men 'cast' finns)
- primitiva datatyper (lagras i variabeln)
- referensdatatyper (refererar till objekt)
- skuggning bör undvikas (lokala variabler med samma namn som klassvariabler)

Viktiga objekt

- String
- `String s = new String("foo");`
- `String s = "foo";` // syntaxen är anpassad
- `s.size(), s.equals("bar")` // även vanlig klass
- Arrayer – deklaration, användning
- `int [] ar = new int [14];`
- Arraytyp t ex `int[], double[], String[]`
- Elementtyp `int, double, String` .. `ar[8] : int`
- `Object [] ob = {"foo", new Integer(32) , ...}`

Viktiga objekt

- `java.util.Collections`
- `Listor`
 - `ArrayList`, `LinkedList`, ...
- `Mängder (Set)`
- `Hashtable`, `HashMap (Map)`

Objektorienterad design

- Klasser – dela upp i 'naturliga' delar
- Klasser är små eller stora beroende på syftet
- Låt delarna kommunicera med varandra mha metoder, och objekt/instanser
- Hitta gränslinjer i problemet

Objektorienterad design

- Inkapsling
- Skydda klasser från insyn, förändring
- publika metoder för kommunikation
- accessors and mutators (get- och set-metoder)
- kontroll av indata
- uppgradera insidan utan att ändra annan kod
- Använd inkapsling – var konservativ
- Börja med private och öppna vid behov

Skriva kod

- Inte för långa metoder
- När möjligt, dela upp arbetet i mindre metoder
- undvik duplicerad kod
- använd alltid {}, t ex efter if, while, for, ..
- använd alltid indentering
- skriv kommentarer (lagom)
- identifierare (klassnamn, variablenamn, metodnamn)
bör vara självförklarande `for(int i = 0 ; ...)`

Inlämningsuppgifterna

- deadline 7 jan – senare blir rättade senare
- frågor helst via kth/social, men epost är ok
- Pokerproblemet:
 - Analys av en stege: sh-analys (ett förslag)
- Spelautomaten och vinsttabellen
 - if-satser
 - Hashtable
- Nudge: cirkular datastruktur: ny position 0-9
 - $(i + 10 + \text{offset}) \% 10$
 - $0 + 10 + 1 = 11 \% 10 = 1$
 - $0 + 10 - 1 = 9 \% 10 = 9$
 - $9 + 10 + 1 = 20 \% 10 = 0$
 - $9 + 10 - 1 = 18 \% 10 = 8$

$(^{\wedge}_{-}\wedge')$

INGA FLER BILDER