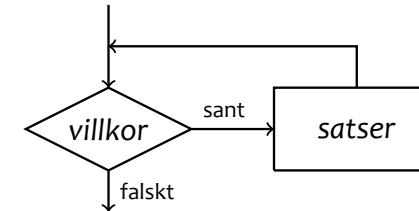


Föreläsning 2: Loopar och objekt

Christian.Soderberg@cs.lth.se

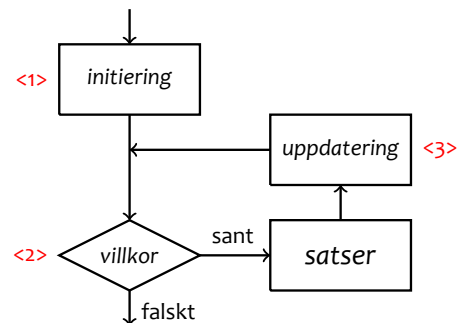
25 januari 2013

while-satsen



- Satserna inne i en while-sats utförs så länge villkoret är uppfyllt
- Villkoret testas bara före varje nytt varv, inte medan vi kör ett varv.

for-satsen



- Första ledet är det som görs första varvet.
- Andra ledet testas före varje varv, även det första.
- Tredje ledet utförs efter varje varv.

for eller while?

I princip kan vi använda vilken av de båda looparna som helst, men under de första veckorna av kursen kan vi använda:

for: när vi i förväg vet över vilka värden vi skall loopa.

while: när vi inte i förväg vet hur många varv vi skall loopa.

- ▶ Två sorters kommentarer:
 - ▶ Enradskommentarer, inleds med //
 - ▶ Flerradskommentarer, omges av /* och */
- ▶ Kommentera gärna *varför* du gör något i ditt program, men kommentera inte *vad* du gör (om det ändå är uppenbart)

- ▶ Det som skrivs innanför ett par klamrar utgör ett *block*.
- ▶ Vi skjuter *alltid* in satserna inuti ett block fyra steg längre än omgivande block.
- ▶ De variabler som deklarerats inne i ett block finns bara inuti blocket.
- ▶ Variabler som deklarerats i rubriken i en for-sats räknas som deklarerade inuti for-satsens block.

Mer om block

- ▶ Deklarera variabler så 'lokalt' som möjligt – vi får på så vis mindre att hålla reda på.
- ▶ Ge variabler startvärden när de deklarerats – om vi inte kan ge dem ett startvärde är det antagligen för tidigt att deklarera dem.

Objekt

Java har två slags typer:

- ▶ Enkla typer, som heltal och reella tal (dessa kallas *primitiva* typer).
- ▶ *Objekt*, som beskriver mer sammansatta värden, som exempelvis:
 - ▶ speltärningar,
 - ▶ personer,
 - ▶ bilar,
 - ▶ geometriska figurer,
 - ▶ ...

Objekt och referenser

- ▶ Vi skapar *objekt* med hjälp av *new*-satsen.
- ▶ Varje objekt är av en given *klass*.
- ▶ Klasserna beskrivs av *klass-specifikationer* – för kurspaketet finns dessa på kurshemsidan.
- ▶ *Konstruktorn* visar hur ett objekt skapas (med hjälp av ett *new*-anrop).
- ▶ Vanliga *operationer* (eller *metoder* eller *meddelanden*) visar vad vi kan göra med våra objekt.

Begreppet klass

- ▶ En *klass* beskriver objekt av ett givet slag, vi kan ha många objekt av varje klass (exempelvis många tärningar eller sköldpaddor).
- ▶ Klassen (typen) hos ett objekt avgör vad vi kan göra med objektet – de operationer som vi kan anropa på ett objekt beskrivs i *klassspecifikationen* för dess klass.

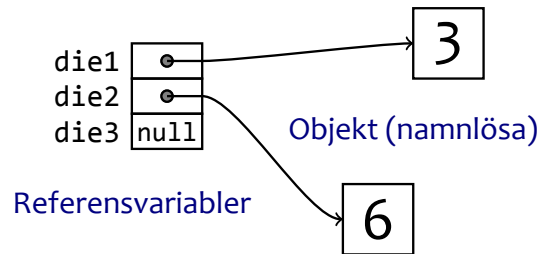
Allmänt om objekt

- ▶ Varje objekt har ett *tillstånd* (antal sidor, antal prickar som visas).
- ▶ Vi kan ändra tillståndet hos ett objekt genom att *anropa operationer* på objektet (exempelvis kasta tärningen).
- ▶ *Parametrarna* till operationerna är deklarerade så att vi skall veta vilket slags värde vi skall skicka in.
- ▶ Vi kan hämta information om objektens tillstånd genom att anropa operationer som returnerar värden – *returtypen* anger vilket slags värde vi får tillbaka, *void*-operationer returnerar inget värde alls.

Klasser i kapitel 3

- ▶ *Die*: beskriver speltärningar.
- ▶ *GraphicsWindow*: beskriver ett slags grafiskt fönster.
- ▶ *Turtle*: beskriver små 'sköldpaddor' som kan krypa omkring i *GraphicsWindow*-fönster.
- ▶ Olika slags geometriska figurer (rektangel, ellips, etc).
- ▶ *DotWindow*: ett annat slags grafiskt fönster som vi kan sätta ut prickar i.
- ▶ *Scanner*: låter oss läsa in från tangentbordet.
- ▶ *Random*: låter oss dra olika slags slumpstal.

Objekt och referenser



- Objekten själva är namnlösa.
- Vi kommer åt våra objekt med hjälp av *referenser*.
- Referenser lagras i referensvariabler.
- En referensvariabel kan referera till ett objekt, eller ha värdet `null`.
- För att anropa en operation på ett objekt använder vi en referens till objektet:

```
die2.roll();
```

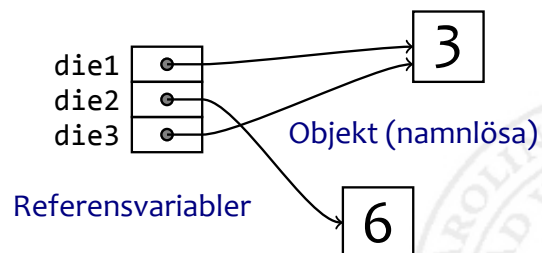
Att skapa objekt

- Objekt skapas med `new`-anrop (som i sin tur anropar konstruktorn).
- Ett objekt får ett väldefinierat starttillstånd när vi anropar konstruktorn.
- När vi skapar ett objekt får vi tillbaka en referens till det nya objektet.

Mer om referenser

Flera referensvariabler kan referera till samma objekt:

```
Die die1 = new Die();  
Die die2 = new Die();  
Die die3 = die1;
```



Klassen Random

- Vi kan skapa slumpital med hjälp av standardklassen `Random`.
- Typiska operationer:
 - `int nextInt(int n)`
 - `double nextDouble()`