

Föreläsning 1

Objektorienterad programmering DD1332

Introduktion till Java
Kompilering, exekvering,
variabler, styrstrukturer

1

Kompilering och exekvering

- Ett program måste översättas till datorns språk för att datorn ska kunna slutföra det. Översättningen görs m.h.a **kompilatorn**.
- Men för att kompilatorn ska kunna översätta så måste programmets olika delar följa språkets regler och syntaxform.

2

Kompilering och exekvering

- Kompilatorn har följande uppgifter:
 1. Kontrollera att programmet har följt språkets regler och syntaxform.
 2. Översätta koden till det språk som kan användas av datorn och spara i en ny fil.
- Man säger att man har kompilerat koden. (Det gör man med kommandot javac)
- När kompileringen inte ger några felmeddelande det betyder att man kan exekvera (köra) filen som skapats av kompilatorn.(java)

3

En kompilerbar och körbar java-kod

```
class ProgEtt{  
    public static void main(String[] args){  
        System.out.println("java är enkel");  
    }  
}
```

4

Kompileringsfel

Följande program ger följande kompileringsfel

```
class ProgEtt{  
    public static void main(String[] args){  
        String mening="java är enkel";  
        System.out.println(mining);  
    }  
}
```

```
progEtt.java:4: error: cannot find symbol  
    System.out.println(mining);  
                        ^
```

```
symbol:   variable mining  
location: class ProgEtt  
1 error
```

5

Variabel, datatyp och litteral

- Variabel är en identifierare i ett program som representerar en plats i datorns minne.
- I Java finns det flera olika datatyper. T.ex: "2" , '2' , 2, 3.14
- Litteral är ett angivet värde i ett program.

6

Datatyper

Primitiva datatyper	Referens datatyper
byte	String
short	Scanner
int	System
long	Integer
double	Double
float	Boolean
Char	...
boolean	

7

Deklarering, tilldelning och initiering

- Deklarering av variabler betyder att man talar om för datorn vilken datatyp variabeln kommer att ha under hela programmet.
- Tilldelning är då en variabel får ett värde.
- Initiering är då en variabel får sitt absolut första värde.

8

Namngivning av variabler

Följande konvention för namngivning av variabler används i java.

- Första tecknet på namnet är **gemen**.
- Om variabelnamnet består av flera ord, första tecknet på alla ord förutom det första ska vara **versal**.

9

Exempel

```
Integer ålder ;
```

← Deklaration av ålder

```
ålder = 12;
```

← Initiering av ålder

```
Double längd = 1.70;
```

← Deklaration och initiering av längd på samma gång

10

Reserverade ord

- Följande ord är reserverad och får (ska) inte användas som identifierare.

abstract	default	(goto)	(operator)	synchronized
boolean	do	if	(outer)	this
break	double	implements	package	throw
byte	else	import	private	throws
(byvalue)	extends	(inner)	protected	transient
case	false	instanceof	public	true
(cast)	final	int	(rest)	try
catch	finally	interface	return	(var)
char	float	long	short	void
class	for	native	static	volatile
(const)	(future)	new	super	while
continue	(generic)	null	switch	

11

Sats och block

- Ett program består av en eller flera satser. En sats i java avslutas med `;` och i vissa fall inom ett block `{ }`
- Block `{ }` används bl.a. för att flera satser ska hålla ihop. När en sak är gemensam för flera satser som kommer efter varandra.

12

Utskrift satser

- `System.out.print()`
 - En sats som skriver ut parameternsvärde.
- `System.out.println()`
 - En sats som skriver ut parameternsvärde och byter automatiskt rad fteråt.

13

Inläsning från tangentbord

```
import java.util.Scanner;

class FintNamn{
    public static void main(String[] args){
        Scanner tgb = new Scanner(System.in);
        System.out.print("Vad heter du? ");
        String namnet = tgb.nextLine();
        System.out.println(namnet + " är ett fint namn");
    }
}
```

14

Operatörer

Op	Exempel	Op	Exempel
+	<code>x+y</code>	+=	<code>x+=2, x+=y</code>
-	<code>y-x</code>	-=	<code>x-=3, x-=y</code>
*	<code>x*y</code>	!	<code>!isEmpty</code>
/	<code>x/y</code>	*=	<code>x*=2, x*=y</code>
++	<code>x++</code> eller <code>++x</code>	/=	<code>x/=2, x/=y</code>
--	<code>y--</code> eller <code>--y</code>		
%	<code>x%y</code>		
==	<code>x==y</code>		
!=	<code>x!=y</code>		

15

Styrstrukturer

- Java kod exekveras radvis och sekventiell.
- Med hjälp av styrstrukturer kan man ändra den sekventiella exekveringen.
- Några exempel på styrstrukturer är:
 - if, if-else, for, while, do-while

16

if-sats

Problem: Skriv ett program som frågar efter pris på boken och skriver ut "dyr bok, ingen affär" om det priset är mer än 500:-

17

if-else

else använder man tillsammans med en if-sats när man vill göra ett val mellan två olika saker.

Exempel: Skriv ett program som frågar efter bokenspris och skriver ut "dyr, ingen affär om priset är mer än 500 kr annars skriver programmet ut "

18

Slinga (loop)

- Slingor eller loop används då man vill en sak ska upprepas ett antal gånger.
- I java finns det for-slingor och while-slingor för detta ändamål

19

Exempel

Skriv ett program som frågar efter användarens ålder. om användaren matar ett in tal mindre än 20 då ska texten skrivas ut:

"tyvärr, men du är välkommen när du blir 20"

Detta upprepas tills användaren matar in ett tal större eller lika med 20.

Då ska skrivas ut på skärmen

"Välkommen till systembolaget".

och därefter programmet avslutas.

20

For-sats

for-sats består av följande delar

1. for
 - 2.(A;B;C)
- A är oftast deklaration och initiering av variabel
 - B är ett villkor-sats (Condition)
 - C oftast är en stegning-sats
 - 3. uttryck (statement)

```
for ( Integer i=0 ; i< 10; i++) {  
    System.out.println("Java är enkelt");  
}
```

21

while sats

while-sats består av följande delar:

1. while
2. Villkor-sats (condition)
3. uttryck (statement)

```
Integer i=0;  
while (i<10) {  
    System.out.println("programmering är kul");  
    i++;  
}
```

22

do-while sats

while-sats består av följande delar:

1. do
2. uttryck
- 3.while
- 4.Villkor-sats (condition)

Integer i=0;

```
do {  
    System.out.println("programmering är kul");  
    i++;  
} while (i>10);
```

23

Exempel

- Skriv ett program som läser in ett heltal från tangentbordet hela tiden och kontrollerar om talet är positiv eller negativ.
- Skriv ett program som skriver ut alla udda nummer mellan 1-99 inklusive 1 och 99.

24

Exempel

Skriv ett program som frågar efter användarens ålder. om användaren matar ett in tal mindre än 20 då ska texten nedan skrivas ut:
"tyvärr, men du är välkommen när du blir 20"
Detta upprepas tills användaren matar in ett tal större eller lika med 20.
Då ska skrivas ut på skärmen
"Välkommen till systembolaget".
och därefter programmet avslutas.

25

API:n

- API står för Application Programming Interface.
- Är egentligen en samling av datatyper som underlättar programmeringen.
- Det finns en bra dokumentation på nätet där kan man få fram information om hur man använder en viss klass och dess metoder.

26

Indexerade variabler

- Det förekommer ofta att programmeraren inte har någon aning om antal data som användaren kommer att använda som indata till programmet.
- Programmet ska inte behöva modifieras om användaren eller antal indata ändras.
- Därför finns det indexerade variabler som vi oftast kallar de vektorer.

27

Array (Hakvektor)

- Beteckningen [] betyder att vi har med en **array** att göra.
- I en variabel av typen `String` kan man bara lagra ett namn åt gången men i en variabeln av typen `String[]` (man läser `String`-array eller array av `String`) kan man lagra flera namn samtidigt. Hur?

28

Exempel

```
String kursbeteckning = "oop15";
```

```
String[] kursBeteckningar = new String[3];  
kursBeteckningar[0] = "oop15";  
kursBeteckningar[1] = "Matematik I";  
kursBeteckningar[2] = "Matematik II";
```

29

Storlek på en array

Varje array har ett attribut som är av typen **int** och visar antal element i arrayen

```
String[] kursBeteckningar = new String[3];  
int storlek = kursBeteckningar.length;
```

OBS! blanda inte *attributen* `length` med *metoden* `length()`
Som finns i klassen `String`

30