

F1 - Introduktion

ID1004 Objektorienterad
programmering

Fredrik Kilander fki@kth.se

Viktiga resurser

- "Java Software Solutions", Lewis & Loftus, sjätte eller sjunde upplagan
- kth.se/social – meddelanden, frågor
- bilda.kth.se – inlämningsuppgiften
- Daisy (<http://daisy.ict.kth.se>) – div adm.
- Kursansvarig: Fredrik Kilander, fki@kth.se
- Bengt Koren (handledning)

Programmera

- Ge datorn instruktioner
- Instruktionerna skrivs i något språk
- FORTRAN, COBOL, LISP, Haskell, Python, Java, C, C++, Ada, C#, Pascal, Prolog...
- Inget språk är bäst på allt
- Val av språk beror ofta på:
 - vad programmet ska göra
 - vad man redan kan
 - vad som finns tillgängligt
 - vad chefen har bestämt

Olika programvarukategorier

- Användarprogram
- Middleware (förser vissa användarprogram med funktionalitet)

User mode

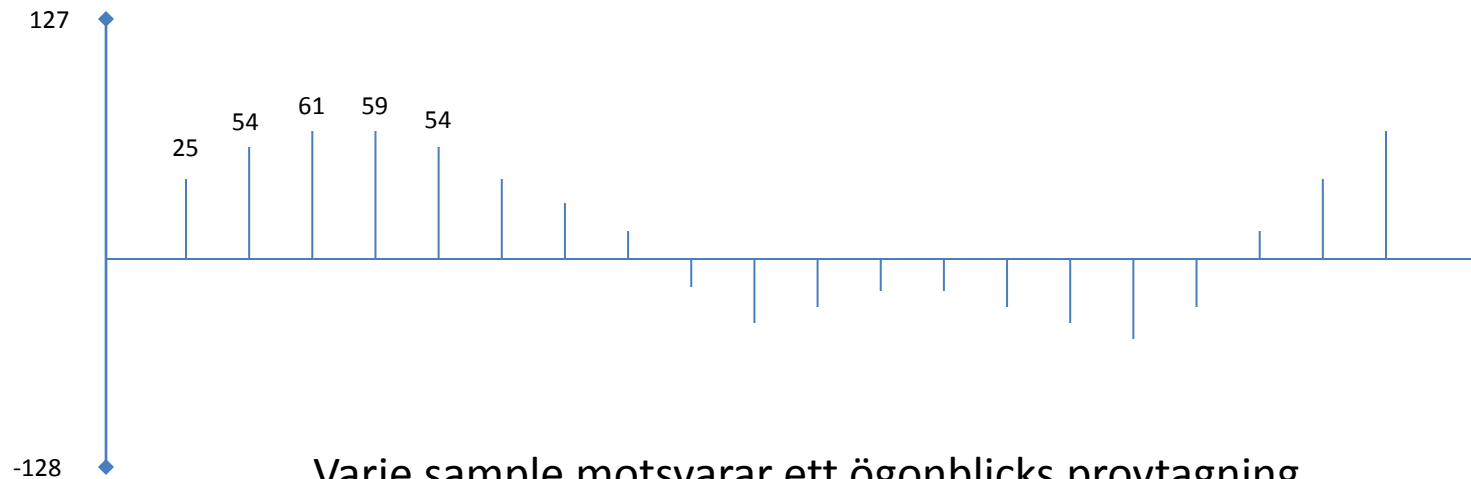
Supervisor mode

- Operativsystem (hanterar datorns resurser)
- Drivrutiner (OS \leftrightarrow kringutrustning)

Digitaliserad data

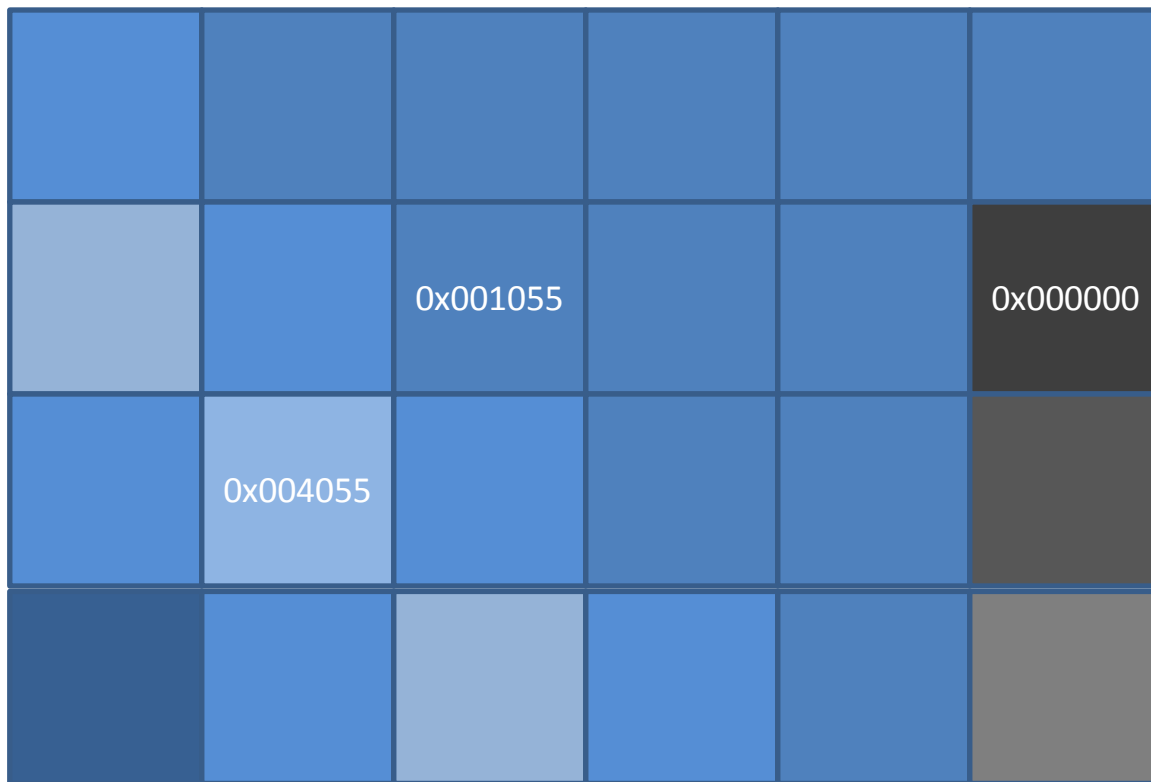
- Allting representeras av tal
- Ljud – samples
- Bild – pixlar
- Text – teckenkoder
- Heltal, flyttal, stora tal

Digitaliserad signal



Varje sample motsvarar ett ögonblicks provtagning.
Om insignalen är bandbredds begränsad $f < S/2$ så
går alla frekvenser att fånga och rekonstruera.

Digitaliserad bild



Provtagning i två dimensioner, med tre färgkanaler i varje mätpunkt.

Binära tal

- Datorns minne och register hanterar bitar
- Bit = 'binary digit', 0 eller 1
- Man har bara två siffror

Decimalt	Binärt
0	0
1	1
2	10
3	11
4	100

Antal bitar och ordlängd

- Varje minnescell eller register i datorn har en begränsad ordlängd (antal bitar)
- Ofta delas denna upp i mindre delar, eller kombineras till längre enheter

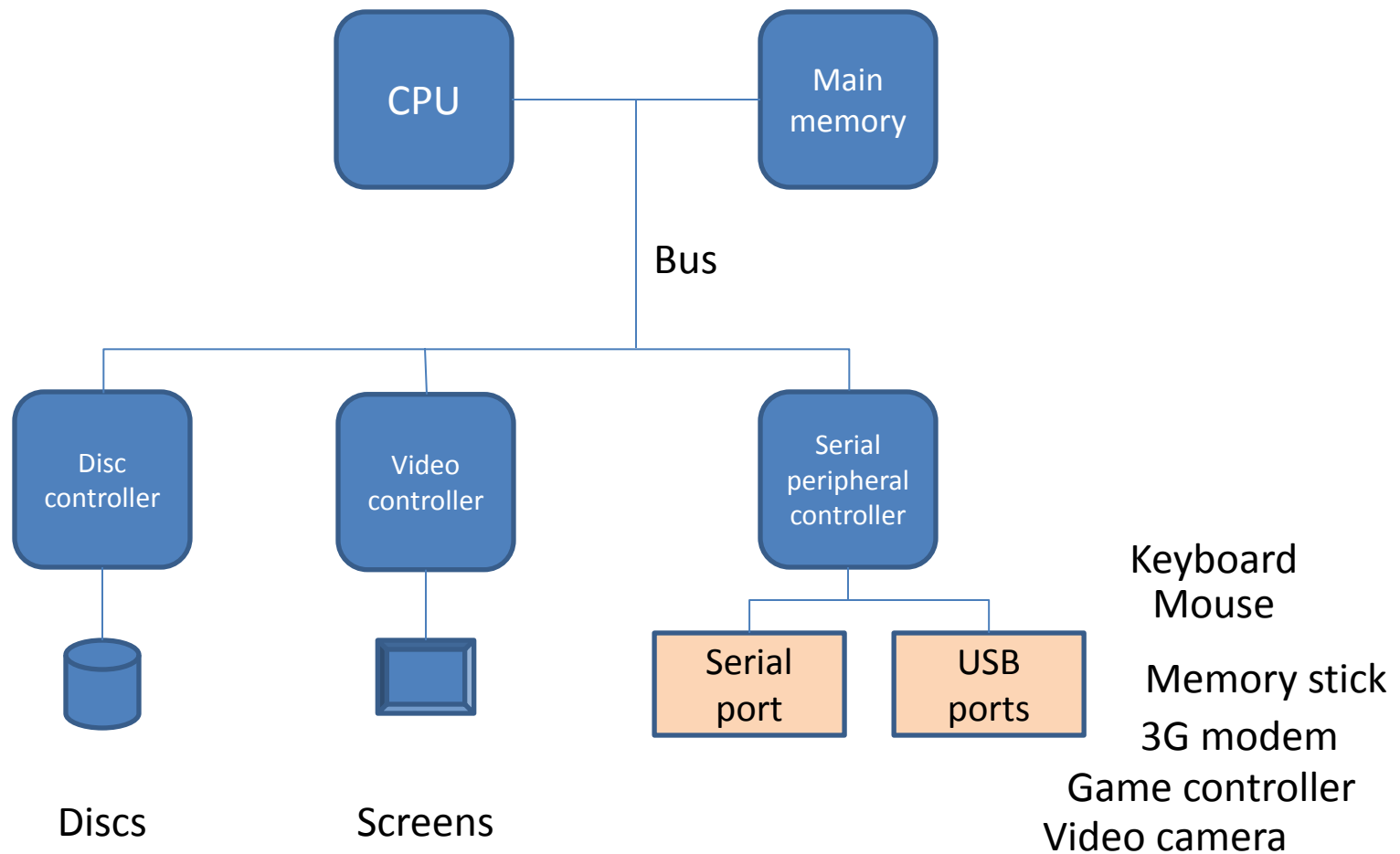


Antalet bitar bestämmer antalet värden

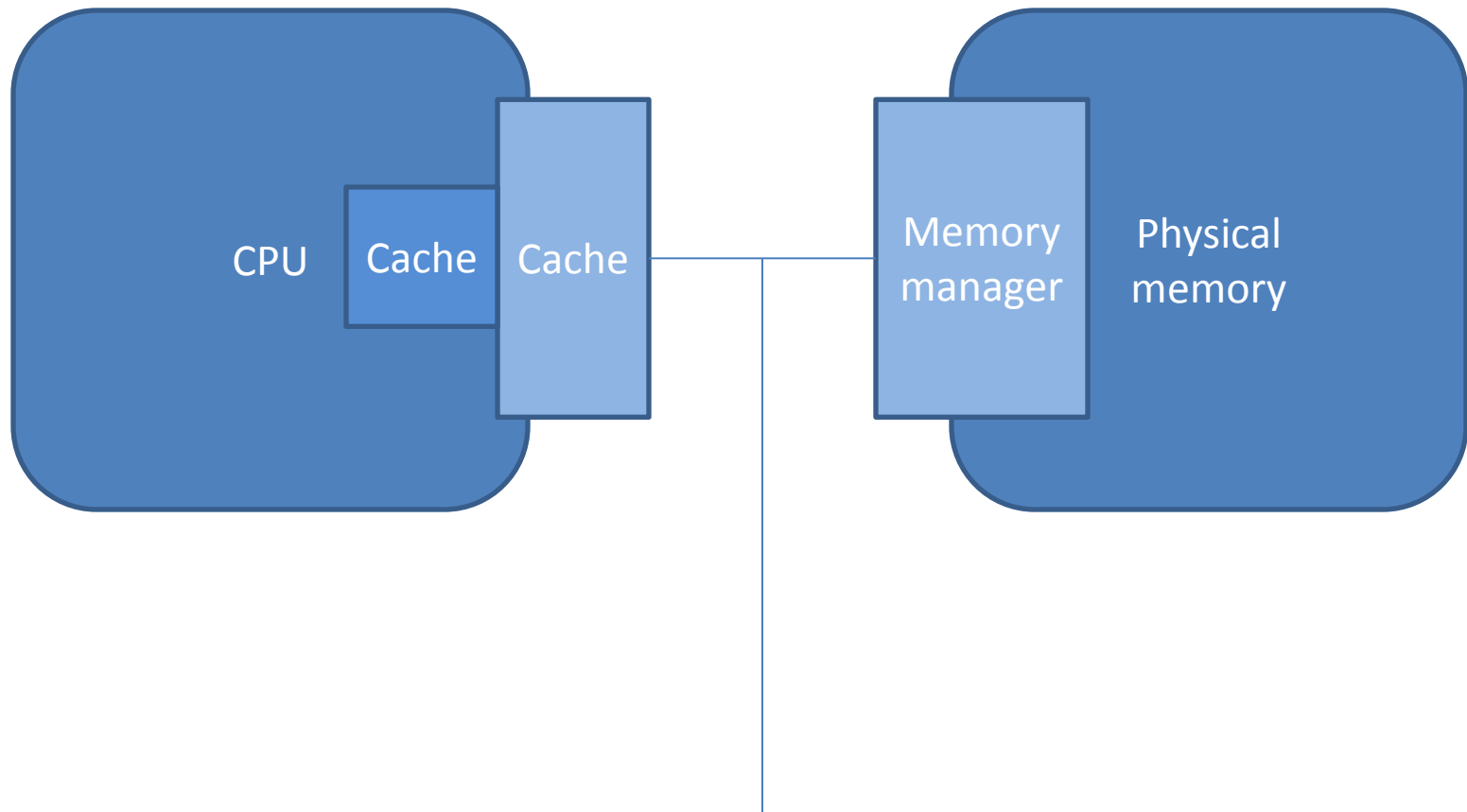
- Ett fält med n bitar, kan representera 2^n olika värden.

n	2^n	Exempel
1	$2^1=2$	0,1
8	$2^8=256$	-128 – 127, 0 – 255
16	$2^{16}=65\,536$	0 – 65 535
32	$2^{32}=4\,294\,967\,296$	4 Gbyte

Datorns delar



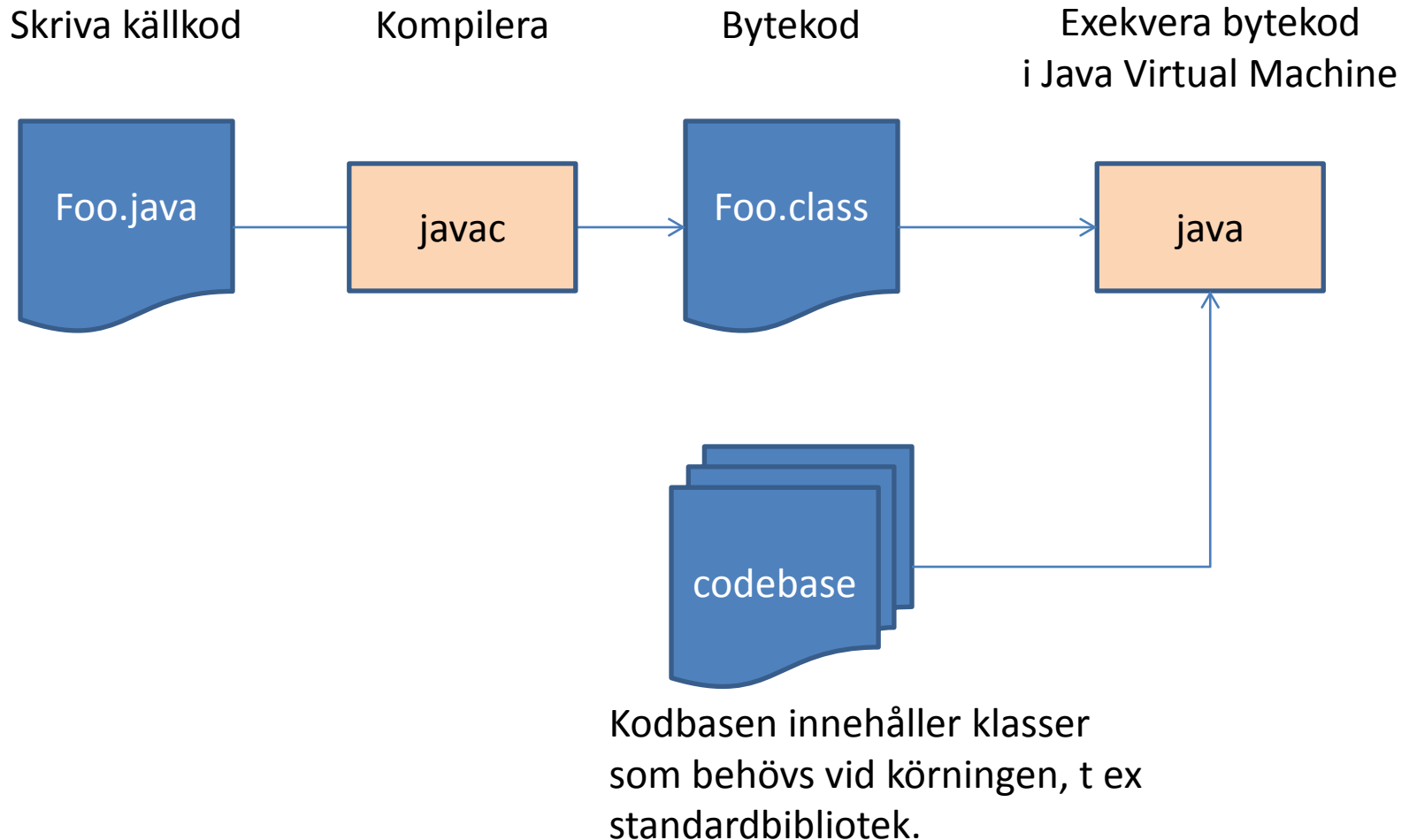
CPU och minne



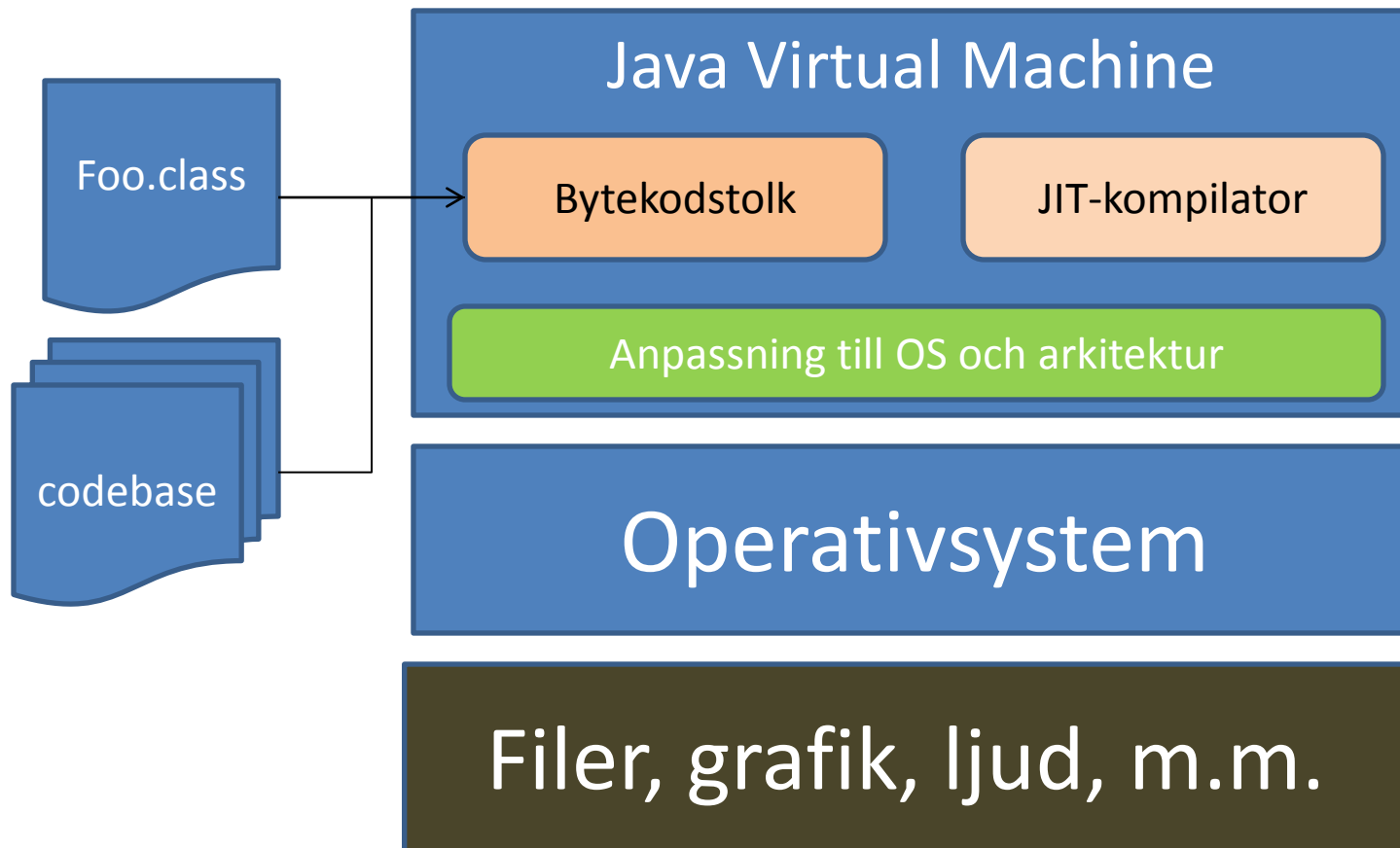
Programmeringsspråket Java

- Skapat 1990-1995 vid Sun Microsystems
- Version 1.0, 1996
- Version 1.1, 1997
- Version 1.2, 1998
- Version 1.3, 2000
- Version 1.4, 2002
- Version 5, 2004
- Version 6, 2006
- Oracle köper Sun Microsystems, 2009
- **Version 7, Juli 2011**
- Version 8, sommaren 2013
- Version 9, ?

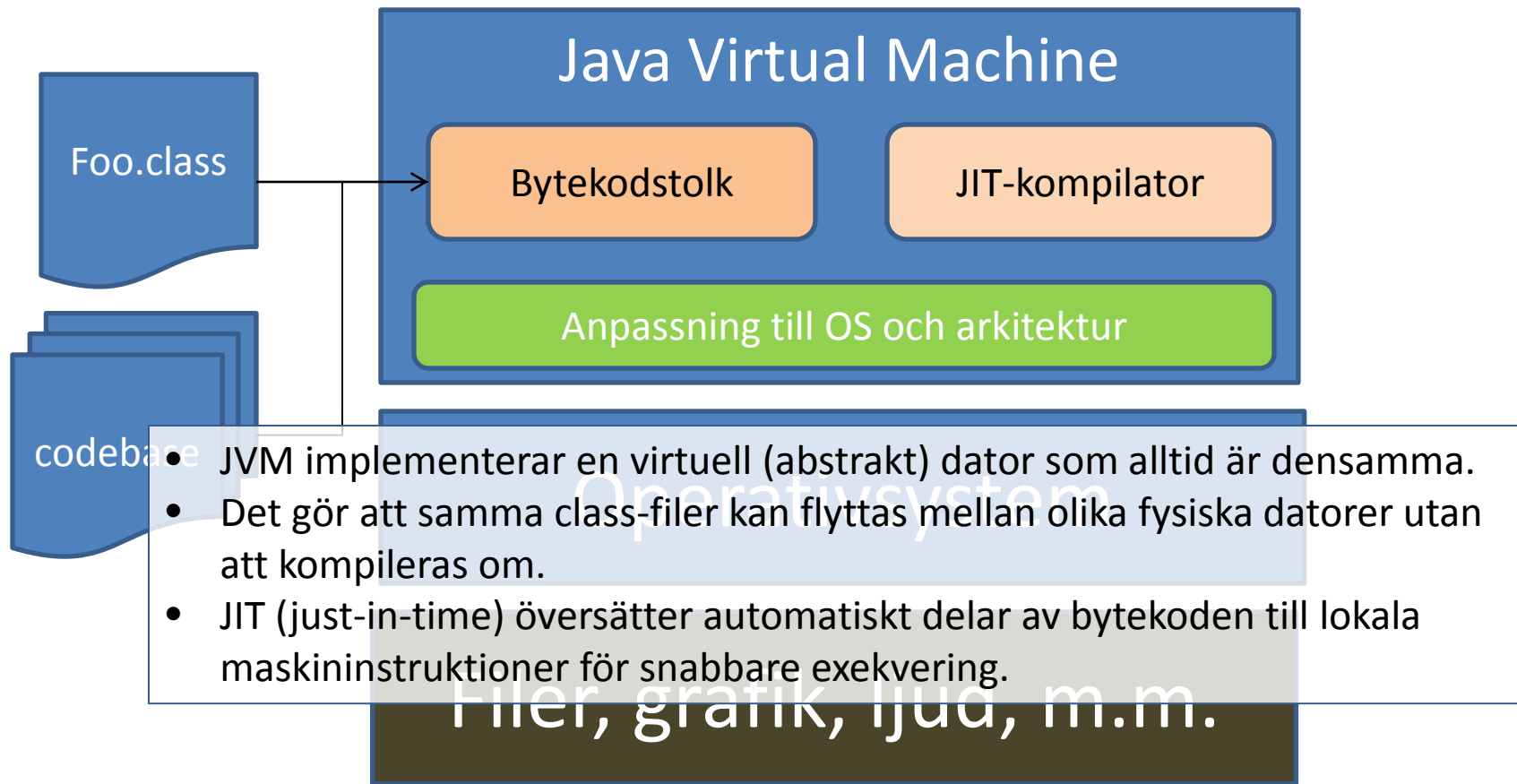
Java programutveckling och exekvering



Java vs operativsystem och dator



Java vs operativsystem och dator

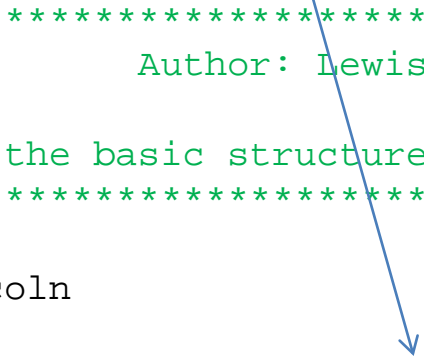


Ett enkelt Javaprogram

```
//*****  
//  Lincoln.java          Author: Lewis/Loftus  
//  
//  Demonstrates the basic structure of a Java application.  
//*****  
  
public class Lincoln  
{  
    //-----  
    //  Prints a presidential quote.  
    //-----  
    public static void main (String[] args)  
    {  
        System.out.println ("A quote by Abraham Lincoln:");  
  
        System.out.println ("Whatever you are, be a good one.");  
    }  
}
```

Ett enkelt Javaprogram

Kommentarer



```
//*****  
//  Lincoln.java          Author: Lewis/Loftus  
//  
//  Demonstrates the basic structure of a Java application.  
//*****  
  
public class Lincoln  
{  
    //-----  
    //  Prints a presidential quote.  
    //-----  
    public static void main (String[] args)  
    {  
        System.out.println ("A quote by Abraham Lincoln:");  
  
        System.out.println ("Whatever you are, be a good one.");  
    }  
}
```

Ett enkelt Javaprogram

Klassen och källkodsfilen har samma namn!

```
//*****  
//  Lincoln.java          Author: Lewis/Loftus  
//  
//  Demonstrates the basic structure of a Java application.  
//*****  
  
public class Lincoln  
{  
    //-----  
    //  Prints a presidential quote.  
    //-----  
    public static void main (String[] args)  
    {  
        System.out.println ("A quote by Abraham Lincoln:");  
  
        System.out.println ("Whatever you are, be a good one.");  
    }  
}
```

Ett enkelt Javaprogram

Reserverade ord (kan inte användas till annat)

```
//*****
//  Lincoln.java          Author: Lewis/Loftus
//
//  Demonstrates the basic structure of a Java application.
//*****

public class Lincoln
{
    //-----
    //  Prints a presidential quote.
    //-----
    public static void main (String[] args)
    {
        System.out.println ("A quote by Abraham Lincoln:");

        System.out.println ("Whatever you are, be a good one.");
    }
}
```

Ett enkelt Javaprogram

Identifierare valda av programmeraren

```
//*****
//  Lincoln.java      Author: Lewis/Loftus
//
//  Demonstrates the basic structure of a Java application.
//*****

public class Lincoln
{
    //-----
    //  Prints a presidential quote.
    //-----
    public static void main (String[] args)
    {
        System.out.println ("A quote by Abraham Lincoln:");

        System.out.println ("Whatever you are, be a good one.");
    }
}
```

The diagram illustrates the identification of key components in a Java program. Three elements are highlighted with boxes: 'Lincoln' (class name), 'main' (method name), and 'args' (parameter name). Arrows point from the text 'Identifierare valda av programmeraren' to these three elements, indicating that these are the identifiers chosen by the programmer.

Ett enkelt Javaprogram

```
//*****  
//  Lincoln.java          Author: Lewis/Loftus  
//  
//  Demonstrates the basic structure of a Java application.  
//*****  
  
public class Lincoln  
{  
    //-----  
    //  Prints a presidential quote.  
    //-----  
    public static void main (String[] args)  
    {  
        System.out.println ("A quote by Abraham Lincoln:");  
        System.out.println ("Whatever you are, be a good one.");  
    }  
}
```

Klass, variabel och metod ur standardbiblioteket

Ett enkelt Javaprogram

```
//*****  
//  Lincoln.java          Author: Lewis/Loftus  
//  
//  Demonstrates the basic structure of a Java application.  
//*****  
  
public class Lincoln  
{  
    //-----  
    //  Prints a presidential quote.  
    //  
    public static void main (String[] args)  
    {  
        System.out.println ("A quote by Abraham Lincoln:");  
        System.out.println ("Whatever you are, be a good one.");  
    }  
}
```

Programmet startar alltid i metoden main.

Kommentarer

```
/*  
    Kommentarstext på flera rader  
*/
```

```
/*  
 * Kommentarstext på flera rader  
*/
```

```
/* Kommentar mitt i en rad */
```

Dessa är syntaktiskt identiska

```
// Kommentar till slutet av raden
```

```
/**
```

Lägg märke till dubbelasterisken!

```
 * JavaDoc-kommentar  
 */
```

JavaDoc är ett verktyg för att generera dokumentation direkt ur källkoden. Kommentaren står alltid omedelbart före det som kommenteras.

Kommentarer

- Kommentarer i källkod är precis lika viktiga som koden.
- Kommentera
 - syfte, avsikt,
 - funktion,
 - mening,
 - användning.

Kommentarer och indentering är för människor

- Kommentarer gör koden mer förståelig för människor
- Indentering (formatering) av koden gör den mer läslig för människor
- Kompilatorn bryr sig inte om kommentarer och tomrum.

```
public class Lincoln{public static void main(String[] args){  
System.out.println("A quote by Abraham Lincoln:"); System.  
out.println("Whatever you are, be a good one.");}}
```

Kommentarer och indentering

- Skriv källkod som om den vore ett exempel
- Lita inte på det egna minnet
- När koden ändras, ändra kommentarerna med!

Kommentarer

A

```
noItem = scan.nextInt(); // Läs en int  
scan.nextLine();         // Läs raden
```

B

```
noItems = scan.nextInt(); // Läs önskat antal  
scan.nextLine();         // Hoppa över resten av raden
```

Vilket exempel är bäst? Varför?

Kommentarer och identifierare

```
public int bestest (int a, int b) {return (a>b)?a:b;}
```

```
/**  
 * Returns the higher of two integer values.  
 * @param a The first value  
 * @param b The second value  
 * @return The greater of the two values.  
 */  
public int max (int a, int b) {  
    return (a > b) ? a : b;  
}
```

Kommentarer och identifierare

```
public int bestest (int a, int b) {return (a>b)?a:b;}
```



Luddigt valt namn

```
/**
 * Returns the higher of two integer values.
 * @param a The first value
 * @param b The second value
 * @return The greater of the two values.
 */
public int max (int a, int b) {
    return (a > b) ? a : b;
}
```

Identifierare

- Identifierare används för att ge namn åt klasser, variabler och metoder
- Reserverade ord kan inte användas som identifierare
- Stor och liten bokstav är OLIKA i Java

P1, p1, p_1

getelement, GetElement,
getElement, GET_ELEMENT

Dessa är alla olika
och unika

Identifierare

- Halva jobbet med att programmera är att hitta på bra namn på identifierare
- Stavningskonventioner underlättar att känna igen vad identifieraren används till

`TT_WORD, COLOR_BLACK`

Konstanter, stor bokstav, underscore

`Lincoln, System`

Klasser, stor begynnelsebokstav

`main, i, nextInteger()`

Variabler, metoder, liten begynnelsebokstav

Traditionella identifierare

- i, j, k – index i arrayer, strängar, samlingar (heltal)
- n – antal (heltal)
- a, b, u, v, w – storheter, faktorer, tal
- e – element (i mängd)
- p – punkt i koordinatsystem
- s, t - strängar
- x, y, z - koordinater

Java reserved words

abstract	continue	for	new	switch
assert***	default	goto*	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum****	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp**	volatile
const*	float	native	super	while

* not used
** added in 1.2
*** added in 1.4
**** added in 5.0

Java reserved keywords – primitive data types

abstract	continue	for	new	switch
assert***	default	goto*	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum****	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp**	volatile
const*	float	native	super	while

* not used
** added in 1.2
*** added in 1.4
**** added in 5.0

Java reserverade ord – ofta använda

abstract	continue	for	new	switch
assert***	default	goto*	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum****	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp**	volatile
const*	float	native	super	while

* not used
** added in 1.2
*** added in 1.4
**** added in 5.0

Java reserverade ord – ibland använda

abstract	continue	for	new	switch
assert ^{***}	default	goto [*]	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum ^{****}	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp ^{**}	volatile
const [*]	float	native	super	while

* not used
** added in 1.2
*** added in 1.4
**** added in 5.0

Java reserverade ord – sällan använda

abstract	continue	for	new	switch
assert***	default	goto*	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum****	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp**	volatile
const*	float	native	super	while

* not used
** added in 1.2
*** added in 1.4
**** added in 5.0

Objektorienterad programmering

- Bryt ner uppgiften i mindre delar
- Låt delarna samverka på väldefinierade sätt
- Välj eller konstruera lämpliga LEGO-bitar
- Sätt samman bitarna till en lösning
- Välj eller definiera lämpliga klasser
- Skapa instanser ur klasserna och anropa metoder



Viktiga objektorienterade begrepp

- Klass – definierar data och metoder
- Metod – namngiven enhet med programkod
(kallas i andra språk *funktion*, *procedur*)
- Objekt – en instans av en klass, tar plats
- Metodanrop – kör koden i en viss metod

Ett enkelt objektorienterat Javaprogram

```
/** *****  
//  Lincoln.java          Author: Lewis/Loftus  
//  Demonstrates the basic structure of a Java application.  
/** *****  
public class Lincoln  
{  
    public void printQuote() {  
        System.out.println ("A quote by Abraham Lincoln:");  
  
        System.out.println ("Whatever you are, be a good one.");  
    }  
    //-----  
    //  Prints a presidential quote.  
    //-----  
    public static void main (String[] args)  
    {  
        Lincoln abe = new Lincoln();  
        abe.printQuote();  
    }  
}
```

Ett enkelt objektorienterat Javaprogram

Klass

```
//*****  
// Lincoln.java           Author: Lewis/Loftus  
// Demonstrates the basic structure of a Java application.  
//*****  
public class Lincoln  
{  
    public void printQuote() {  
        System.out.println ("A quote by Abraham Lincoln:");  
  
        System.out.println ("Whatever you are, be a good one.");  
    }  
    //-----  
    // Prints a presidential quote.  
    //-----  
    public static void main (String[] args)  
    {  
        Lincoln abe = new Lincoln();  
        abe.printQuote();  
    }  
}
```

Ett enkelt objektorienterat Javaprogram

Metod

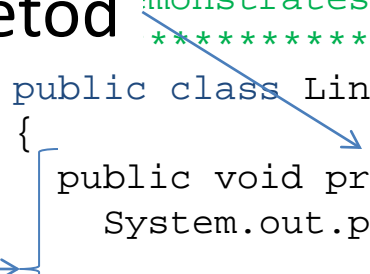
```

//*****
//  Lincoln.java          Author: Lewis/Loftus
//  Demonstrates the basic structure of a Java application.
//*****

public class Lincoln
{
    public void printQuote() {
        System.out.println ("A quote by Abraham Lincoln:");

        System.out.println ("Whatever you are, be a good one.");
    }
    //-----
    //  Prints a presidential quote.
    //-----
    public static void main (String[] args)
    {
        Lincoln abe = new Lincoln();
        abe.printQuote();
    }
}

```



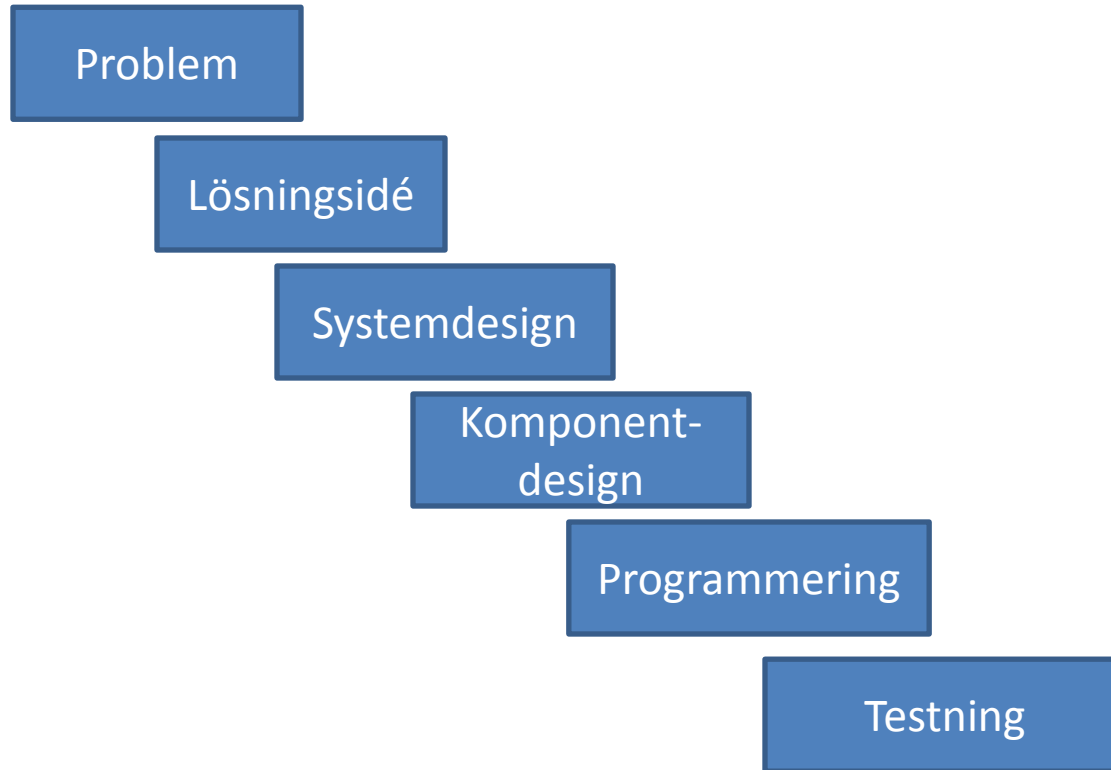
Ett enkelt objektorienterat Javaprogram

```
/** *****  
//  Lincoln.java      Author: Lewis/Loftus  
//  Demonstrates the basic structure of a Java application.  
/** *****  
public class Lincoln  
{  
    public void printQuote() {  
        System.out.println ("A quote by Abraham Lincoln:");  
  
        System.out.println ("Whatever you are, be a good one.");  
    }  
    //-----  
    //  Print Skapar ny instans av klassen -----  
    //-----  
    public static void main (String[] args)  
    {  
        Lincoln abe = new Lincoln();  
        abe.printQuote();  
    }  
}
```

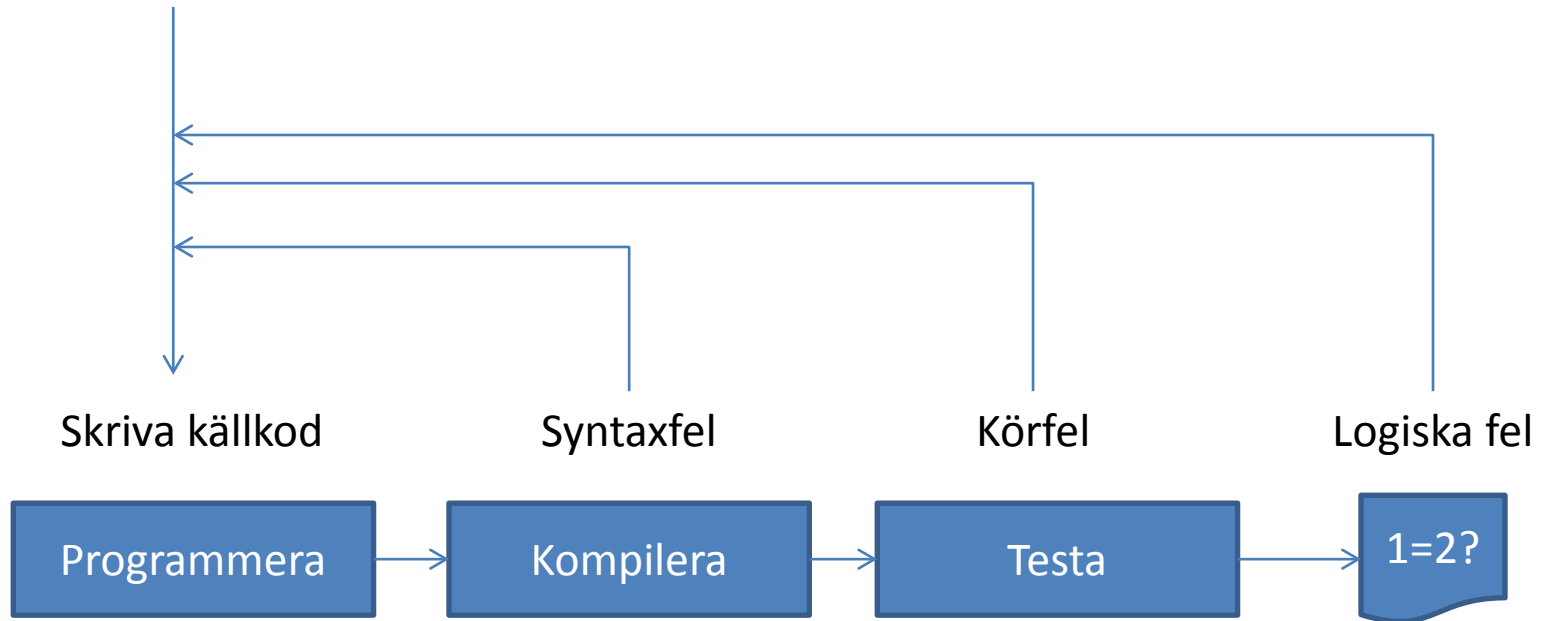
Ett enkelt objektorienterat Javaprogram

```
/** *****  
//  Lincoln.java          Author: Lewis/Loftus  
//  Demonstrates the basic structure of a Java application.  
/** *****  
public class Lincoln  
{  
→ public void printQuote() {  
    System.out.println ("A quote by Abraham Lincoln:");  
  
    System.out.println ("Whatever you are, be a good one.");  
}  
//-----  
//  Prints a presidential quote.  
//-----  
public static void main (String[] args)  
{  
    Lincoln abe = new Lincoln();  
    abe.printQuote(); ← Metodanrop  
}  
}
```

Programmering är 'bara' en del...



Programmering är 'bara' en del...



F1 – INGA FLER BILDER