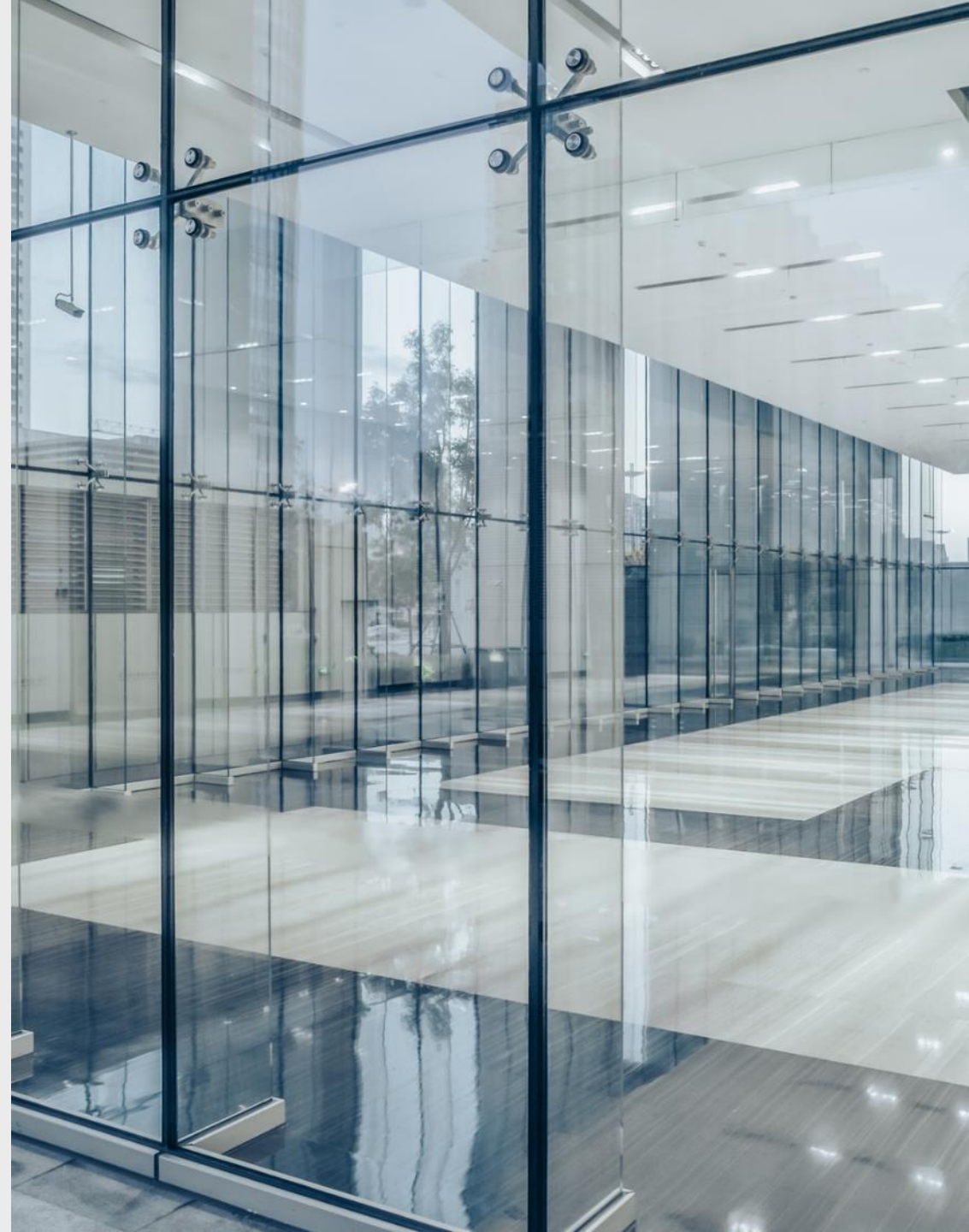


Objektorienterad programmering i Java

Michael Granbäck
michael.granback@xlent.se



Programmering

Ett programspråk har en specifik syntax

Syntax är regler som säger:

- Vilka nyckelord som används
- Hur dessa ord och symboler sätts ihop till korrekta satser

Ett program består av ett antal satser

Programmering sker i princip genom att:

- 1. Skriver kod som vanlig text
- 2. Sparar koden till en fil
- 3. Kompilerar programmet
- 4. Exekvera programmet

Editiera/spara --> kompilera --> exekvera/kontrollera



Enkelt Javaprogram – HelloApplication.java

- vad består det av?

```
/*
 * HelloApplication.java
 * Author: Michael Granbäck
 * Created: 2022-08-25
 */
package mypackage; // Package declaration
/*
 * A program that prints "Hello world!" to the screen
 */
public class HelloApplication { // Class definition
    public static void main(String[] args) { // Main method

        System.out.println("Hello world!"); // Print
    }
}
```





Vad är objektorienterad programmering?

4

9/4/2024

- ▶ Klass – Programmet byggs upp av delar, klasser, efter de begrepp som hanteras. Varje klass är en abstraktion av ett begrepp.
- ▶ Inkapsling - Ett objekts exakta beteende och funktion är dold för omvärlden. Vi kan specificera hur klasser eller objekt får interagera med varandra.
- ▶ Arv – En klass kan ärva egenskaper från en annan klass. Detta innebär att man kan skapa nya klasser genom att utgå från befintliga.
- ▶ Polymorfism – En metod kan ha olika beteenden beroende på vilken klass den anropas från.

Klasser och objekt

5

9/4/2024

- ▶ En klass definierar egenskaper hos en viss typ av objekt
- ▶ Ett objekt tillhör (definieras av) en klass
- ▶ Ett objekt är en instans av en klass
- ▶ Klassen Car och två olika instanser av klassen:

Klassen Car:	Objekt 1	Objekt 2	Objekt 3
Ägare:	Anna	Erik	Nils
Regnr:	RAA71G	UDD31E	GAB438
Nypris:	433150.99kr	149800.00kr	210100.90kr

Klassen bestämmer egenskaper hos ett eller flera bilobjekt

Varje objekt är en konkret bil

KUNSKAPER

- ▶ Arv för programmering av system
- ▶ Objektorienterad programmering med Java
- ▶ Programmeringsspråket Java

FÄRDIGHETER

- ▶ Skapa och använda klasser och objekt i programkod
- ▶ Använda sig av arv för programmering av system
- ▶ Implementera objektorientering med Java
- ▶ Använda attribut och inkapsling i objektorientering

KOMPETENSER

- ▶ Implementera enklare applikationer med objektorienterad programmering
- ▶ Designa en objektmodell från en enkel kravspecifikation

PRINCIPER FÖR BETYGSSÄTTNING

9

Den studerandes prestation betygsätts efter genomförd kurs med betygen Icke Godkänt (IG), Godkänt (G) eller Väl Godkänt (VG).

Kursbetyget baseras på en sammanvägning av samtliga kontroller.

Om den studerande genomför alla examinerande moment/uppgifter/muntlig redovisning men inte uppnår kursens mål erhålls betyget "IG".

Om den studerande inte genomför alla examinerande moment/uppgifter/tentamen enligt anordnarens anvisningar efter avslutad kurs anges "betyg kan ej sättas".

Icke Godkänt (IG) Den studerande har genomfört kursen utan att nå alla kursens mål på minst en G-nivå.

Godkänt (G) Den studerande har genomfört kursen och nått alla kursens mål.

Väl Godkänt (VG) Den studerande har genomfört kursen och nått alla kursens mål. Förutom detta kan den studerande självständigt genomföra implementationen av grundläggande applikationer med hjälp av objektorienterad programmering. Dessutom har den studerande förmågan att självständigt designa en objektmodell baserad på en enkel kravspecifikation, samt analysera och motivera sina val och genomförande.

KUNSKAPSKONTROLL

❖ En inlämningsuppgift för hela kursen
Deadline 8 oktober, klockan 17:00

❖ Muntlig redovisning
Äger rum någonstans mellan 10-11 oktober

Se till att ha tillgång till:

- Webbkamera
- Mikrofon

Varför?

Ett krav för att jag ska kunna bedöma era kunskaper.

Dessutom blir det mycket trevligare övrig tid om vi kan se varandra och diskussioner flyter på bättre och mer effektivt än vid chattande.



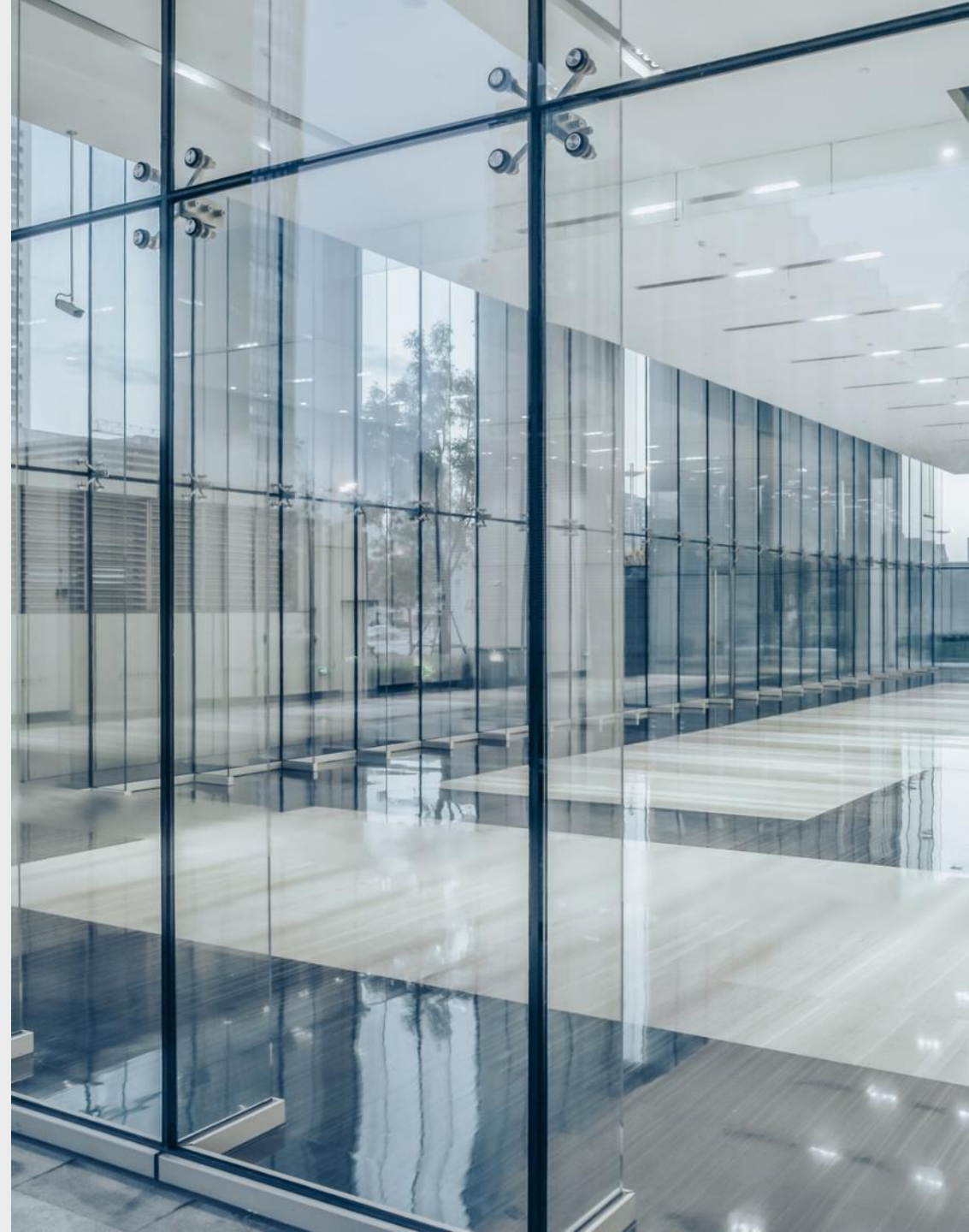
<https://www.jetbrains.com/idea/download/>



INTELLIJ IDEA

Integrated Development Environment

- Installera IntelliJ
- Kom igång med exempelkod
- Översikt
- JDK-version



Agenda

- ▶ Datatyper
- ▶ Variabler, konstanter
- ▶ Array
- ▶ Input / Output
- ▶ Operatorer
- ▶ Selektioner
- ▶ Iterationer
- ▶ Statiska metoder

I mån av tid kör vi en gruppövning, som inte har specifikt med Java att göra, för att få lite omväxling. Övningen relaterar dock till systemutveckling



Variabler

Primitiva Datatyper

m.m.

Datatyp	Tilldela ett värde	Värde	bytes
byte	byte tinyNumber = 127;	-128 till 127	1
short	short smallNumber = 32767;	-32,768 till 32,767	2
int	int normalNumber = 2_147_483_647;	-2,147,483,648 till 2,147,483,647	4
long	long hugeNumber = 9_223_372_036_854_775_807 L;	-9,223,372,036,854,775,808 till 9,223,372,036,854,775,807	8
float	float floatNum = 2.1234567f;	7 decimalers precision	4
double	double doubleNum = 2.123456789101112d;	15 decimalers precision	8
char	char myFirstChar = 'A'; char mySecondChar = 65;	Ett enda Unicode-tecken	2
boolean	boolean needCoffee = true;	true eller false	1
String	String name = "Lars";	Unicode tom	-

Heltalstyper - Heltalsoperationer

Aritmetiska operatorer

+ (Addition)

- (Subtraktion)

* (Multiplikation)

/ (Division)

% (Modulus)

Demo:

Heltalsdivision

Modulus $A \% B = A - A / B * B$

Teckentypen char

```
public class HelloApplication { // Class definition
    public static void main(String[] args) { // Main method
        String name = "Michael Granbäck";
        char dot = '.';
        char first = name.charAt(0); // Get first character
        char second = name.charAt(8); // Get second character
        System.out.println("My initials are " + first + dot + second + dot);
    }
}
```

Utskrift: My initials are M.G.

Den logiska typen boolean

En variabel av datatypen boolean kan endast ha värdet true eller false

```
boolean a = true;
```

```
boolean b = 5 < 7;           // b = true
```

```
boolean c = 5 + 2 <= 6 * 1   // c = false
```

```
boolean d = 12 == 3 * 4      // d = true
```

Jämförelseoperator == har uttrycken lika värde?

Heltalsoperatorer

Increment:

`n++;` \Leftrightarrow `n = n + 1;`

Decrement:

`n--;` \Leftrightarrow `n = n - 1;`

Add. Assignment:

`n += 2;` \Leftrightarrow `n = n + 2;`

Subtr. Assignment:

`n -= 2;` \Leftrightarrow `n = n - 2;`

Fungerar lika för `*`, `/`, `%`

Identifierare: bokstäver siffror _ \$

```
int userAge = 40;  
String userName = "James Gosling";  
double userSalary = 41277.51;  
char myFavoriteChar = 'A';  
char myFavoriteCharAsAscii = 65;  
boolean gateOpen = false;
```

Namngivning variabler

lowerCamelCase

Namngivning konstant

```
final String PERSONAL_NUMBER = "550519-1996";  
UPPERCASE_WITH_UNDERSCORE_AS_SEPARATOR
```



Variabler, konstanter

```
7 int userAge;  
8 System.out.println(userAge);  
9  
10  
11  
12  
13
```

Variable 'userAge' might not have been initialized

Initialize variable 'userAge' Alt+Skift+Enter More actions... Alt+Enter

int userAge

DummyProject

Variabeln är deklarerad men har ej tilldelats ett värde, den har inte initierats

```
7 int userAge;  
8 userAge = 40;  
9 System.out.println(userAge);  
10 userAge = 41;  
11 System.out.println(userAge);  
12  
13  
14
```

Variabeln deklareras, ej initierad
Därefter tilldelas den ett värde och har nu initierats

På rad 10 tilldelas userAge värdet 41 och värdet för variabeln ändras

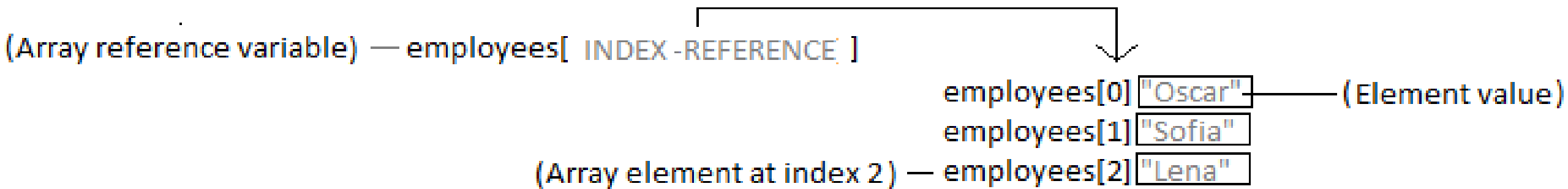
Initierad variabel

ARRAY

En array i Java är en samling element av samma datatyp som lagras i minnet på ett sekventiellt sätt. Varje element i en array kan nås genom att använda dess index. Index är ett heltal som representerar positionen för elementet i arrayen. Indexet för det första elementet i en array är alltid 0.




```
String employee1 = "Oscar";  
String employee2 = "Sofia";  
String employee3 = "Lena";
```



Array – indexed variabel

Skapa en array

```
String[] employees = new String[]{"Oscar", "Sofia", "Lena"};
```

```
String[] upcomingEmployees = new String[3];  
upcomingEmployees[0] = "Oscar";  
upcomingEmployees[1] = "Sofia";  
upcomingEmployees[2] = "Lena";
```

```
System.out.println(employees[0]);  
System.out.println(upcomingEmployees[0]);
```

(En indexerad variabel kan användas på samma sätt som en vanlig variabel)

OUTPUT

```
7 String userName = "James Gosling";  
8 int userAge = 40;  
9 double userSalary = 41277.51;  
10 System.out.println("Användarnamn: " + userName);  
11 System.out.print("Ålder: " + userAge);  
12 System.out.println("Lön: " + userSalary);
```

```
"C:\Program Files\Amazon Corretto\jdk11.0.3_7\bin\java.exe"
```

```
Användarnamn: James Gosling
```

```
Ålder: 40Lön: 41277.51
```

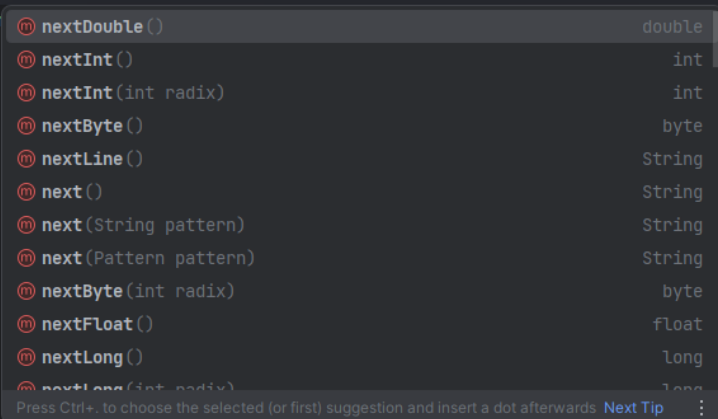
```
Process finished with exit code 0
```

```
System.out.print();  
System.out.println();
```

Använda live template
(fördefinierade kodmallar)
sout [ENTER]

INPUT

```
9 Scanner scanner = new Scanner(System.in);
10 String userName = scanner.nextLine();
11 int userAge = scanner.nextInt();
12 double userSalary = scanner.nextDouble();
13 System.out.println("Användarens namn: " + userName);
14 System.out.print("Ålder: ");
15 System.out.println("Lön: ");
16
17
18
19
20
21
22
23
24
25
26
```



`Scanner scanner = new Scanner(System.in);`

Scanner kan användas för att läsa indata från filer och andra källor, inte bara från användaren.

Utvecklaren vet vilken data som förväntas
Användaren har ingen aning

INPUT

```
Scanner scanner = new Scanner(System.in);
```

```
9 Scanner scanner = new Scanner(System.in);
10 System.out.print("Ange ett användarnamn: ");
11 String userName = scanner.nextLine();
12 System.out.println("Ange ålder: ");
13 int userAge = scanner.nextInt();
14 System.out.print("Ange lön: ");
15 double userSalary = scanner.nextDouble();
16 System.out.println("Användarnamn: " + userName);
17 System.out.print("Ålder: " + userAge);
18 System.out.println("Lön: " + userSalary);
```

```
"C:\Program Files\Amazon Corretto\jdk11.0.3_7\bin\java.exe"
Ange ett användarnamn: James Gosling
Ange ålder:
40
Ange lön: 41277,51
Användarnamn: James Gosling
Ålder: 40Lön: 41277.51

Process finished with exit code 0
```

OUTPUT

System.out.print() Utan radbrytning
System.out.println() Med radbrytning