Implementazione di strutture dati dinamiche in Java

Variabili e tipi di dato

- Variabile: unità di memoria identificata da un nome simbolico
- Ogni variabile ha:
  - un nome simbolico
  - un tipo, che identifica il tipo di informazione che può contenere
  - un contenuto
  - alcuni attributi
- Per quanto riguarda i tipi di dato, Java mette a disposizioni:
  - i tipi di dati primitivi (es. int, float, double, long, char, boolean) anche detti tipi elementari
  - i tipi riferimento (o *reference*) anche detti tipi oggetti

Tipi di dato primitivi

- In una variabile di tipo primitivo, la locazione di memoria identificata dal nome della variabile contiene il valore della variabile.

```
int a;

a = 3;
```

Oggetti

In una variabile di tipo oggetto, la locazione di memoria identificata dal nome della variabile contiene il riferimento (*reference*) ad un'altra locazione di memoria dove è contenuto l'oggetto

```
Integer b;

b = new Integer(3);
```

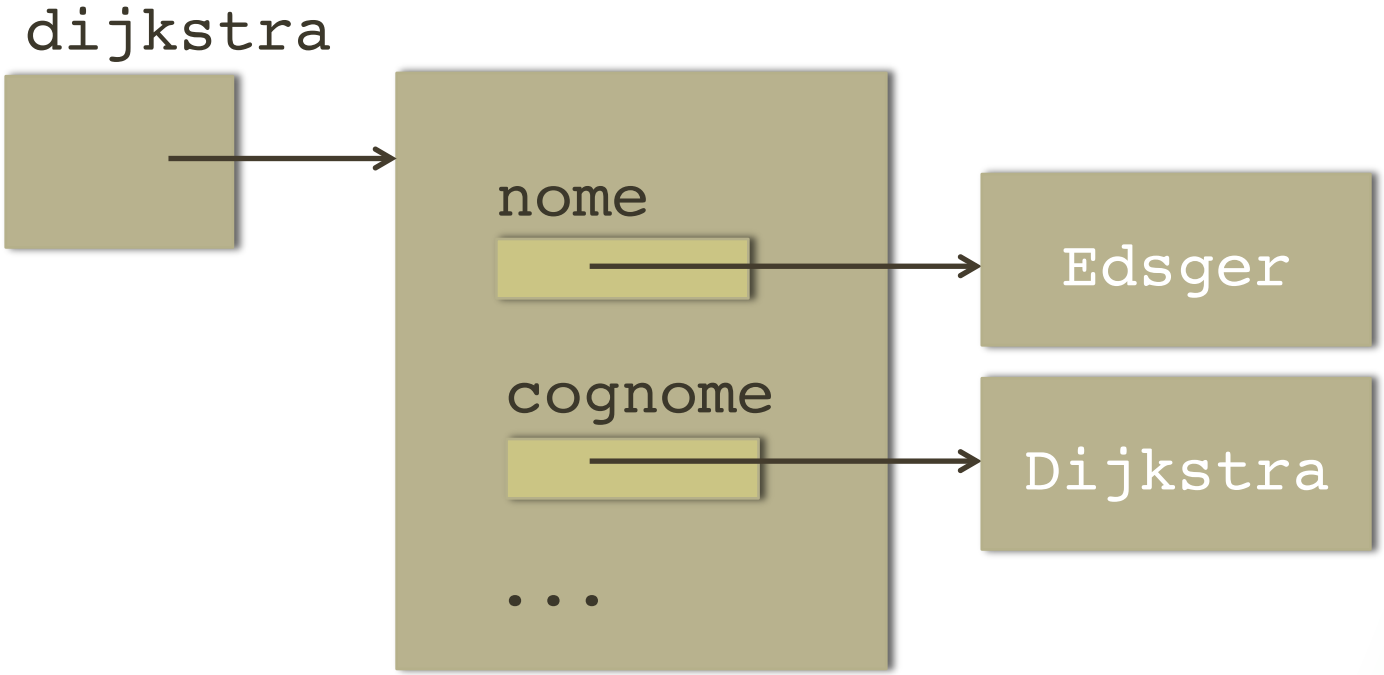
Definizione di classi

- In Java esistono classi predefinite (es. Integer) ma è anche possibile definire nuove classi.
- Una classe può contenere variabili di tipo diverso.

```
public class Utente{

    String nome, cognome;
    String nickName;
    Date   dataDiNascita;

    …
}
```

Definizione di classe

```
Utente dijkstra =
   new Utente(“Edsger”,”Dijkstra”,…);
```



Esercizio

Si supponga di volere definire una classe che rappresenti un utente di una piattaforma sociale nella quale:

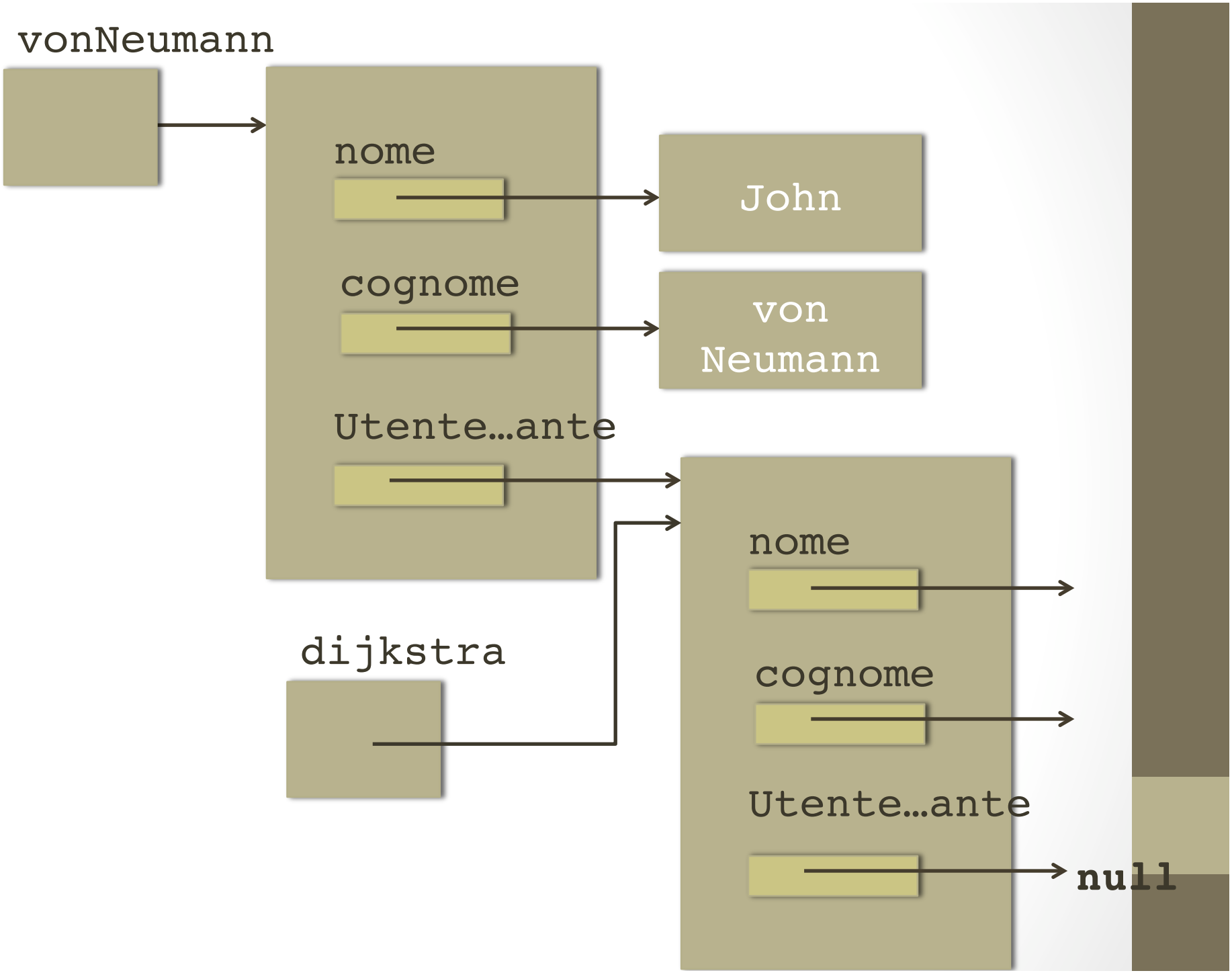
- ogni utente può entrare solo se presentato da un altro utente

Cosa si deve aggiungere alla classe utente precedentemente definita?

```
public class Utente{

    String nome, cognome;
    String nickName;
    Date   dataDiNascita;
    Utente utentePresentante;

    …
}
```

Esercizio

```
Utente dijkstra=
     new Utente(“Edsger”,”Dijkstra”,…);
Utente vonNeumann =
     new Utente(“John”,”von Neumann”,…,
          dijkstra);
```


vonNeumann
nome
John
cognome
von
Neumann
Utente…ante
nome
dijkstra
cognome
Utente…ante
null

Esercizio

- Si supponga che nel social network un utente possa presentare uno e un solo utente. Modificare la classe precedentemente definita in modo tale da includere quest informazione.

Esercizio

- Si supponga di volere definire una classe che rappresenti un social network.

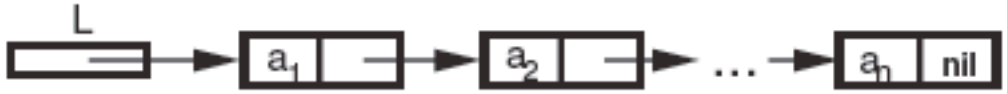
```
public class SocialPlatform{

    ...
    Utente fondatore;
    Utente ultimoEntrato;
    ...
}
```

Esercizio

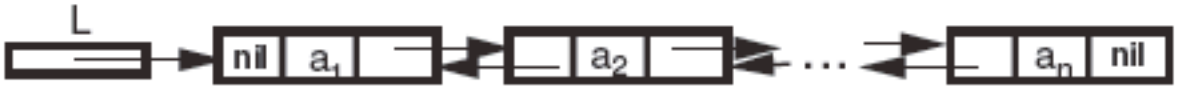
- Implementare un metodo della classe `SocialPlatform` che permetta l'inserimento di un nuovo utente.
- Inserire nel social network implementato dalla classe `SocialPlatform` i seguenti utenti:
  - Alan Turing
  - Niklaus Wirth
  - Ada Lovelace
  - Tim Berners-Lee

  Implementare un metodo della classe `SocialPlatform` che stampi tutti gli utenti che attualmente presenti nella piattaforma

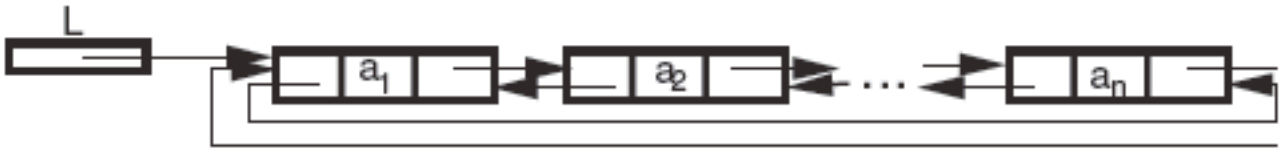
Come implementare una lista?



monodirezionale



bidirezionale



bidirezionale circolare

monodirezionale con sentinella

Come implementare una lista

```
public class Elem{
    Object data;
    Elem next;
    ...
}

public class Lista{
    Elem head;
    ...
}
```

Esercizio

Implementare la classe Lista implementando i seguenti metodi:

LIST

```
LIST pred          % Predecessore
LIST succ          % Successore
ITEM value         % Elemento

LIST List()
    LIST t ← new LIST
    t.pred ← t
    t.succ ← t
    return t

boolean isEmpty()
    return pred = succ = this

POS head()
    return succ

POS tail()
    return pred

POS next(POS p)
    return p.succ

POS prev(POS p)
    return p.pred

boolean finished(POS p)
    return (p = this)

ITEM read(POS p)
    return p.value

write(POS p, ITEM v)
    p.value ← v

POS insert(POS p, ITEM v)
    LIST t ← List()
    t.value ← v
    t.pred ← p.pred
    p.pred.succ ← t
    t.succ ← p
    p.pred ← t
    return t;

POS remove(POS p)
    p.pred.succ ← p.succ
    p.succ.pred ← p.pred
    LIST t ← p.succ
```