

ALBERI

Angelo Di Iorio

Università di Bologna

Esercizio 1

- Implementare una classe Java per memorizzare e leggere informazioni relative a una dinastia (esempio nella prossima slide)
- Per ogni persona:
 - Nome
 - Anni di vita

Dinastia dei Medici

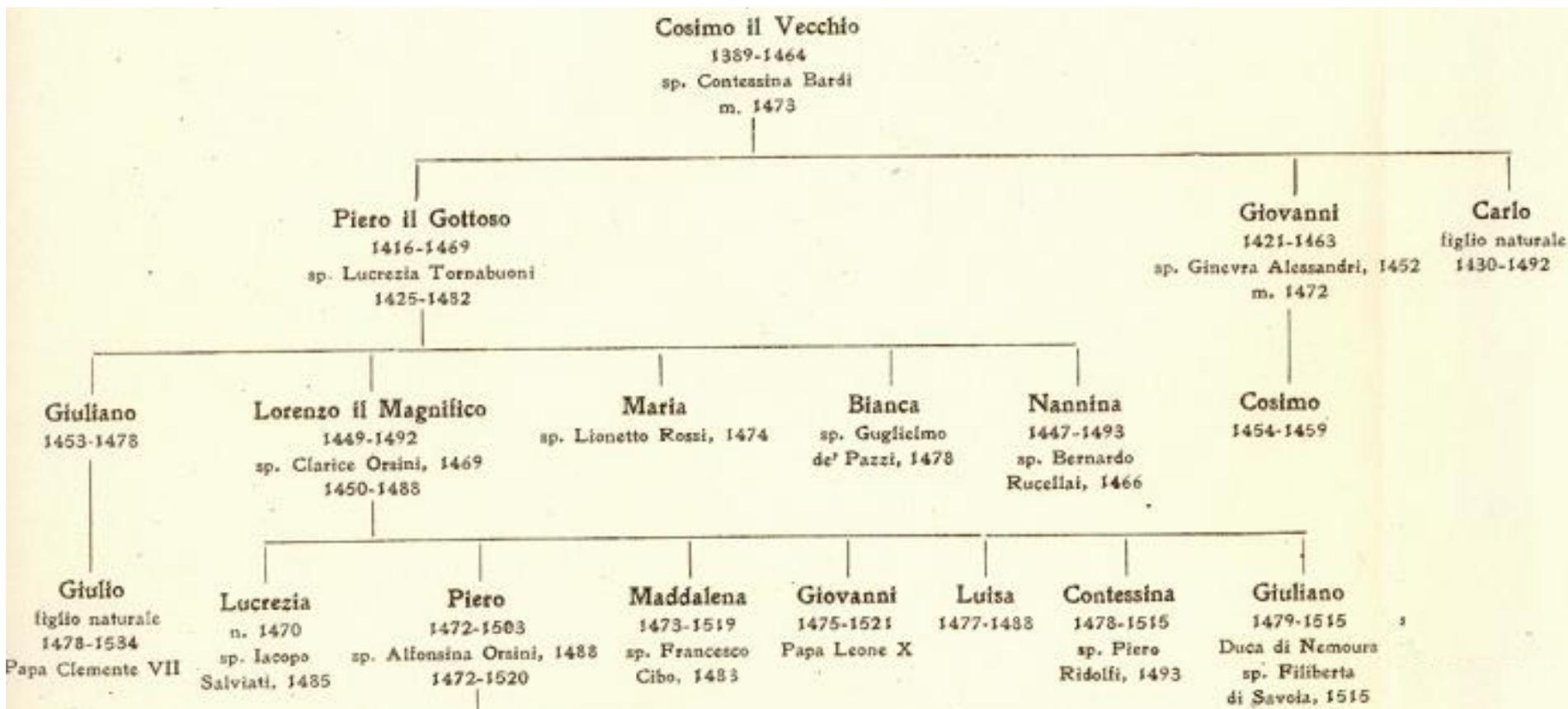


Immagine completa: <https://www.firenze-online.com/img/artisti/foto-famigliademedici3.jpg>

Esercizio 1

- La classe `DinastyTree` rappresenta un nodo (sottoalbero) ed espone i metodi per:
 - inizializzare il nodo (radice) con nome e anni
 - accedere alle informazioni relative al nodo (nome, anni, figli, padre)
 - aggiungere un nodo come ultimo figlio di questo nodo
 - aggiungere un insieme di figli (passati con un vettore)

- Realizzazione basata su lista dei figli

Esercizio 2

- Implementare una classe `DinastyTreeVisits` per visitare l'albero.
- Implementare un metodo `printTree(DinastyTree t)` che stampa su terminale i dati memorizzati nell'albero
 - visualizzazione indentata semplice, vedi prossima slide
- Dati di esempio su:
<http://diiorio.nws.cs.unibo.it/asd1718/pdf/DinastyTreeTest.java>

Esercizio 2

```
Cosimo il Vecchio(75)
  Piero il Gottoso(53)
    Giuliano(25)
      Giulio(56)
    Lorenzo il Magnifico(43)
      Lucrezia(34)
      Piero(31)
      Maddalena(46)
      Giovanni II(46)
      Luisa(46)
      Contessina(37)
      Giuliano II(36)
    Maria(66)
    Bianca(66)
    Nannina(46)
  Giovanni(42)
  Carlo(62)
```

Esercizio 3

- Aggiungere alla classe `DinastyTreeVisits` un metodo (ricorsivo) per cercare un discendente per nome

```
DinastyTree cercaDiscendentePerNome(DinastyTree tree,  
String nome);
```

- Il metodo ritorna il primo discendente (in previsita) con lo stesso nome
- Domande:
 - Quanti figli ha avuto Lorenzo il Magnifico?
 - Quanti anni ha vissuto Piero il Gottoso?
 - Cosimo il Vecchio ha avuto un discendente chiamato Guido?
 - ...

Esercizio 4

- Aggiungere un metodo per cercare il primo discendente che ha vissuto almeno X anni

```
public DynastyTree  
cercaPrimoDiscendenteEtaMinima(DynastyTree tree,  
Integer anni);
```

- Analizzare nell'ordine: tutti i figli, poi tutti i nipoti, poi tutti i pronipoti, etc.

Esercizio 5

- Dato un albero ordinato i cui nodi contengono valori interi, si vogliono contare tutti i nodi (anche quelli intermedi) tali per cui il percorso radice-nodo ha somma complessiva dei valori uguale a k .

Implementare in Java:

- una classe `TreeInt` che rappresenta l'albero (per semplicità: costruttore e metodo per aggiungere un figlio)
- una classe `TreeIntVisit` che implementa un metodo per contare il numero di nodi che soddisfano la proprietà vista sopra:

```
public Integer nodeSumK(TreeInt t, Integer k,  
Integer somma);
```