

# Tecniche di Progettazione: Design Patterns

GoF: Builder, CoR, Flyweight

# Ex 1. Builder Advent Calendars

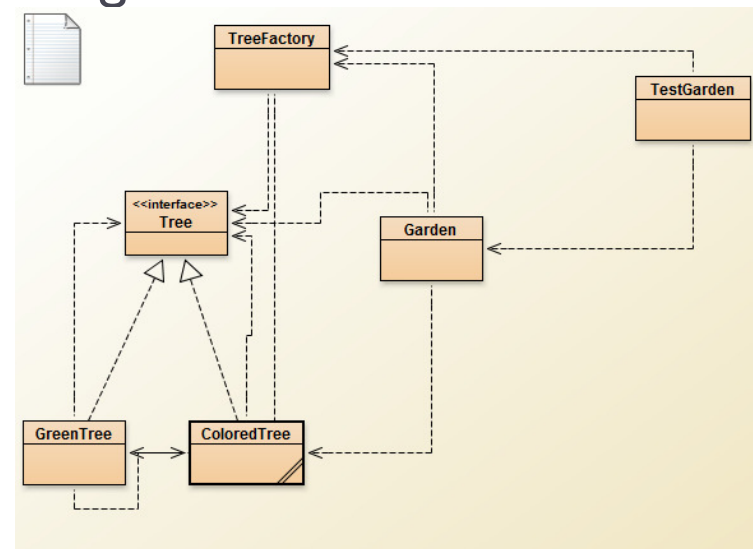
---

- ▶ Using pattern Builder, write the code to construct Advent Calendars;
- ▶ (Different representation may mean that one builder uses a queue and the other one an ArrayList)
  - ▶ Either one filled with chocolate, offering method
    - ▶ Chocolate dequeue() (if you do not use queue see.g. open(int i))
  - ▶ Or one filled with hipster pieces (all the same but one) for the crib, offering method
    - ▶ CribPiece get(int i)
- ▶ Builders offer two methods: build() and buildSpecial()
- ▶ The Director has knowledge of how any calendar has to be built: 23 normal pieces, a special one



## Ex 2. Flyweight

- ▶ You have to fill a garden (10x10 matrix) with trees.
- ▶ To add a tree, the garden uses a Factory.
- ▶ A tree offers a
  - ▶ `display()` method printing its color.
  - ▶ `Tree setColor(String color)` to change the color
- ▶ Default trees are green.  
(Flyweight)
- ▶ When a green tree is colored of a different color, a new object is instantiated.



## Ex3. CoR

---

### **Example: Trusty Bank**

To further explore the Chain of Responsibility pattern, we'll model the action of a bank that is careful about allowing withdrawals of large amounts. Trusty Bank's business rules for handling withdrawals are:

- Clerks can handle withdrawals of up to \$1,000.
- Supervisors can handle withdrawals of up to \$4,000.
- The bank manager can handle withdrawals of up to \$9,000.