

# La Banca

Thread  
synchronized  
wait/notify



# Descrizione

Realizzare una classe **Banca** che dispone di 10 conti correnti con un saldo iniziale di 5000 €

La banca fornisce 2 metodi di business:

- ▶ `void bonifico(int ccOrdinante,  
int ccBeneficiario,int importo)`
  - ▶ Sposta l'importo specificato dal conto ordinante al beneficiario (se c'è disponibilità)
  - ▶ NOTA: i soldi si spostano solo all'interno della stessa banca!
- ▶ `int getPatrimonio()`
  - ▶ Calcola il totale dei saldi di tutti i conti

La classe Banca sarà condivisa dai Thread descritti di seguito

# Gestori dei conti

Realizzare inoltre una classe **ThreadConto** che rappresenta il gestore di un conto.

Sarà definita da un numero e un'istanza di Banca.

- ▶ Il ThreadConto ordinerà - presso la sua banca - bonifici a ripetizione sugli altri conti → loop infinito
- ▶ L'importo versato sarà 500€.
- ▶ I conti beneficiari saranno decisi randomicamente

# Gestore patrimonio

Realizzare inoltre una classe **ThreadPatrimonio** che rappresenta il gestore del patrimonio della banca e che quindi disporrà di un'istanza di Banca.

- ▶ Il ThreadPatrimonio recupera il totale del patrimonio e lo stampa - all'infinito - ogni 500 millisecondi.
- ▶ Avendo 10 conti con 5000 € e considerato che i bonifici sono sempre interni, il totale mostrato dovrà sempre essere 50.000€

# Uso di `synchronized`

- ▶ Fornire una prima implementazione senza **`synchronized`** che metta in luce le problematiche di incongruenza dovuta alla mancata sincronizzazione dei thread.
- ▶ Fornire una seconda implementazione che renda la classe Banca **threadsafe**
- ▶ Usare eventualmente **`wait`** e **`notify`** per assicurarsi che il bonifico sia eseguito solo dal threadconto che ha realmente disponibilità a farlo.