

# Simulatore “Noleggio Smart Mobility”

## Requisiti funzionali (minimi)

1. *L'app deve permettere di:*
  - Registrare un parco mezzi (almeno 6 oggetti di tipi diversi).
  - Elencare solo i mezzi disponibili.
  - Prenotare un mezzo per una durata in minuti.
  - Chiudere una corsa e calcolare il costo totale (polimorfico).
2. *Il costo dipende dal tipo del mezzo (tariffe diverse e/o sovrapprezzi).*
3. *Se il mezzo è elettrico, la batteria scende durante la corsa; se scende sotto una soglia, applicare un fee di “ricarica emergenza”.*
4. *Se il mezzo è pieghevole, applicare uno sconto su tratte brevi.*
5. *Validazioni essenziali (es. non prenotare mezzi già in uso, durata > 0, batteria 0–100, ecc.).*

## Dominio (vincoli di design)

### 1) Classe astratta

- *abstract class Veicolo*
  - *Campi privati + proprietà (ID, Modello, TariffaBaseAlMinuto, IsDisponibile, ecc.).*
  - *Costruttori coerenti.*
  - *Metodi:*
    - *public abstract decimal CalcolaCosto(int minuti); (polimorfismo obbligatorio)*
    - *public virtual void AvviaCorsa() / public virtual void TerminaCorsa() con comportamento base (modifica disponibilità).*
  - *ToString() sovrascritto per stampa leggibile.*

### 2) Sottoclassi concrete (almeno 3)

- *Esempi: Monopattino, Bici, Scooter, Auto, E-Bike.*
- *Ognuna ridefinisce CalcolaCosto con logiche diverse (sovrapprezzi/sconti).*

### 3) Interfacce (almeno 2, usate da più classi)

- *interface IElettrico*
  - *Proprietà: int Batteria { get; }*
  - *Metodi: void Ricarica(int percento);*
  - *Contratto: durante TerminaCorsa() un mezzo elettrico riduce la batteria in base ai minuti.*
- *interface IPieghevole*
  - *Proprietà: bool ÈPiegato { get; }*
  - *Metodi: void Piega(); void Apri();*
  - *Contratto: eventuale sconto su tratte brevi quando piegato.*
- *(Facoltativa) interface IPrenotabile { bool Prenota(); bool Rilascia(); }*

*Nota: Le interfacce devono essere indipendenti dalla gerarchia (le implementano solo i tipi pertinenti).*

## Incapsulamento (obbligatorio)

- *Tutti i dati sensibili privati.*
- *Proprietà con validazione (es. TariffaBaseAlMinuto > 0, Batteria tra 0 e 100).*
- *Metodi pubblici chiari e coesi; niente setter pubblici non necessari.*

## Polimorfismo (obbligatorio)

- *Usare Veicolo come tipo di riferimento in collezioni/metodi.*
- *Esempio: List<Veicolo> parco; e operare senza conoscere il tipo concreto.*
- *Il calcolo del costo avviene chiamando CalcolaCosto(minuti) su Veicolo.*

## Comportamenti suggeriti (esempi da adattare)

- *Monopattino (elettrico): costo = base/min + sovrapprezzo se batteria < 15%.*
- *Bici (pieghevole): costo = base/min – sconto sui primi 10 minuti se piegata a inizio corsa.*
- *Scooter (elettrico): costo = base/min + fee fisso per avvio; consumo batteria maggiore.*