

## Requisiti

Implementare una soluzione software che consenta a vari tipi di veicoli di parcheggiare in un parcheggio coperto.

- Quando un veicolo arriva al parcheggio coperto e se c'è spazio disponibile, dovrebbe apparire sullo schermo il messaggio “Benvenuto, entri pure”.
- Quando un veicolo arriva all'ingresso del parcheggio coperto e questo è pieno, dovrebbe apparire il messaggio “Spiacenti, non ci sono più posti disponibili”.
- Deve accettare più tipi di veicoli: auto, furgoni, motociclette.
- Ogni tipo avrà dimensioni diverse (ad esempio, 2 motociclette possono parcheggiare in 1 posto auto, un furgone occupa 1,5 posti auto).
- Deve avere 3 piani, ciascuno con una capacità diversa.
- Un furgone può parcheggiare solo al piano terra.

## Suggerimenti

- Dovrai fare alcune ipotesi mentre lavori a questa sfida, quindi ti preghiamo di includerle chiaramente, indicando come deve essere eseguito il programma e qualsiasi altro commento che ritieni utile per i revisori del codice nel file README.md incluso in questo compito.
- Come per qualsiasi cosa nel software, è necessario trovare un equilibrio tra estensibilità/manutenibilità e tempo impiegato (costo).
- Va bene se volete discostarvi dai requisiti non funzionali indicati di seguito, ma tenete presente il tempo che volete dedicare a questo. Se vi discostate, aggiungete i motivi di tali deviazioni nel file README.md incluso in questo compito.

## Criteri di valutazione

- Correttezza: implementazione priva di bug secondo i requisiti;
- Qualità del codice: quanto è pulito, leggibile, ha una denominazione corretta e quanto è organizzato il codice; ha le giuste astrazioni?
- Scalabilità del codice: quanto sarebbe facile aggiungere ulteriori funzionalità senza riscritture
- Manutenibilità: quanto sarebbe facile per un nuovo programmatore imparare il codice e comprendere i concetti e le regole del dominio
- Test: quanto le tue unità di test aggiungono affidabilità all'implementazione. Aspettative
- La soluzione deve essere scritta in OOP PHP. Non devono essere necessari database o altre risorse esterne (ad esempio, utilizzare la memoria o mock)
- Il programma deve essere in grado di funzionare e verificare se i requisiti sono stati implementati
- Mantenerlo semplice (ad esempio, senza framework) considerando i limiti di tempo
- Non è richiesta alcuna interfaccia web, l'app può essere eseguita tramite riga di comando