

Nome..... Cognome..... Matricola.....

**È VIETATO l'uso di oggetti diversi dalla penna. Dove previsto scrivere la risposta CHIARAMENTE nell'apposito spazio.**

### Quesito 1

Si consideri il seguente modello concernente alcune componenti di una libreria grafica.

(A) Definire la seguente gerarchia di classi.

1. Definire una classe base astratta `Widget` i cui oggetti rappresentano un generico componente (un cosiddetto widget) di una Gui. Ogni `Widget` è caratterizzato da larghezza e altezza in pixels, e dall'essere visibile o meno.
  - `Widget` è astratta in quanto prevede il metodo virtuale puro `void setStandardSize()` che deve garantire il seguente contratto: `w->setStandardSize()` imposta la dimensione larghezza×altezza definita come standard per il widget `*w`.
  - `Widget` rende disponibile almeno un opportuno costruttore per impostare le caratteristiche dei widget.
2. Definire una classe `AbstractButton` derivata da `Widget` i cui oggetti rappresentano un generico componente pulsante. Ogni oggetto `AbstractButton` è caratterizzato dalla stringa che etichetta il pulsante.
  - `AbstractButton` rende disponibile almeno un opportuno costruttore per impostare le caratteristiche dei pulsanti.
3. Definire una classe `PushButton` derivata da `AbstractButton` i cui oggetti rappresentano un pulsante clickabile.
  - `PushButton` implementa il metodo virtuale puro `setStandardSize()` come segue: per ogni puntatore `p` a `PushButton`, `p->setStandardSize()` imposta la dimensione standard 80×20 per il pulsante clickabile `*p`.
  - `PushButton` rende disponibile almeno un opportuno costruttore per impostare le caratteristiche dei pulsanti clickabili.
4. Definire una classe `CheckBox` derivata da `AbstractButton` i cui oggetti rappresentano un pulsante checkabile. Ogni oggetto `CheckBox` è caratterizzato dall'essere nello stato "checked" o "unchecked"; inoltre, tutti gli oggetti `CheckBox` sono sempre visibili.
  - `CheckBox` implementa il metodo virtuale puro `setStandardSize()` come segue: per ogni puntatore `p` a `CheckBox`, `p->setStandardSize()` imposta la dimensione standard 5×5 per il pulsante checkabile `*p`.
  - `CheckBox` rende disponibile almeno un opportuno costruttore per impostare le caratteristiche dei pulsanti checkabili.

(B) Definire una classe `Gui` i cui oggetti rappresentano le componenti di una Gui. Un oggetto `Gui` è caratterizzato da:

- un `std::vector<NoButtons>` di oggetti di tipo `const Widget*` che contiene tutte i widget di una Gui che **non sono** un pulsante.
- una `std::list<Buttons>` di oggetti di tipo `const AbstractButton*` che contiene tutte i widget di una Gui che **sono** un pulsante.

La classe `Gui` rende disponibili i seguenti metodi:

1. Un metodo `void insert(Widget*)` con il seguente comportamento: in una invocazione `g.insert(p)`, se `p` è nullo allora viene sollevata l'eccezione "NoInsert" di tipo `string`; altrimenti, viene inserito il widget `*p` nella Gui `g`.
2. Un metodo `void insert(unsigned int, PushButton&)` con il seguente comportamento: in una invocazione `g.insert(pos, pb)`, se `pos` è un indice valido della lista `Buttons` della Gui `g` allora inserisce il puntatore `pb` nella posizione `pos` della lista `Buttons`; se invece `pos` non è un indice valido allora viene sollevata l'eccezione "NoInsert" di tipo `string`. Si ricorda che, come per tutti i contenitori, gli indici validi per una lista vanno da 0 al numero di elementi contenuti nella lista: quindi, l'indice 0 significa inserimento in testa, mentre un indice uguale al numero di elementi contenuti significa inserimento in coda.
3. Un metodo `vector<AbstractButton*> removeUnchecked()` con il seguente comportamento: una invocazione `g.removeUnchecked()` rimuove dalla Gui `g` tutti i pulsanti checkabili che sono nello stato "unchecked", e ritorna tutti i pulsanti checkabili rimossi in un `vector` di `AbstractButton*`.
4. Un metodo `void setStandardPushButton()` con il seguente comportamento: una invocazione `g.setStandardPushButton()` imposta alla dimensione standard tutti i pulsanti clickabili contenuti nella Gui `g` aventi etichetta diversa dalla stringa vuota.