

# Relazione Progetto C++

Nome: Alaa Eddine  
Cognome: Ghanimi  
Matricola: 856573  
E-mail: a.ghanimi@campus.unimib.it

## Scelte progettuali

L'implementazione della classe generica Set si basa su una linked list semplice. Ogni nodo contiene il valore generico da memorizzare e un puntatore al nodo successivo. Dato che un set non può contenere duplicati, la classe oltre ad essere templata sul tipo T, lo è anche su un tipo Eql, che è il funtore predicato usato dalla classe per effettuare il confronto tra due tipi T.

## Tipi di dati

La classe Set si appoggia su una struttura interna "nodo" per costruire la lista. La struttura nodo contiene un valore generico e un puntatore al nodo successivo. Inoltre, essendo solo di supporto, la gestione dei puntatori e della memoria è stata lasciata alla classe Set.

## Implementazione

La classe Set contiene i seguenti attributi: un puntatore head di tipo nodo che punta al primo elemento della lista, una variabile size che contiene il numero degli elementi inseriti e un funtore equals di tipo Eql usato per effettuare l'uguaglianza tra due tipi T.

La classe è dotata dei seguenti metodi:

- **Costruttore di default:** costruisce un set vuoto inizializzando il numero degli elementi a zero e il puntatore di testa al suo valore di default (puntatore nullo)
- **Copy constructor:** costruisce un set a partire da un altro set scorrendo quest'ultimo e aggiungendo i suoi dati al primo set. Visto che questa funzione richiama la funzione add all'interno di un ciclo, è stato necessario usare un blocco try catch nel caso venisse lanciata un'eccezione causata da una cattiva allocazione dei nodi. Se dovesse essere lanciata quest'eccezione, la classe viene riportata in uno stato coerente svuotando la lista di nodi e rilanciando l'eccezione al chiamante.
- **Operator=:** assegna al set corrente i valori del set passato come parametro. Quest'operazione viene fatta utilizzando un set temporaneo costruito

richiamando il copy constructor e scambiando i suoi attributi con quelli della classe corrente this (il set temporaneo a fine scopo viene distrutto).

- **Distruttore**: distrugge il set scorrendo la lista e cancellando dallo heap i nodi e impostando a zero il numero dei nodi.
- **Costruttore templatato**: costruisce un set di tipo T a partire da due iteratori di tipo Q. Visto che la funzione add viene richiamata all'interno di un ciclo, è stato necessario usare un blocco try catch nel caso venisse lanciata un'eccezione causata da una cattiva allocazione dei nodi. Se dovesse essere lanciata l'eccezione, la classe viene riportata in uno stato coerente svuotando la lista di nodi e rilanciando l'eccezione al chiamante. Inoltre, essendo il tipo degli iteratori diverso da quello della classe, è stato utilizzato un cast statico dal tipo Q al tipo T.
- **Add**: aggiunge il nodo passato come parametro alla coda della lista. La funzione add si appoggia alla funzione contains per verificare che il valore contenuto nel nodo non sia presente e questo viene fatto utilizzando il funtore equals.
- **Remove**: rimuove dalla lista il nodo passato come parametro. Esso si appoggia alla funzione contains per verificare che il valore contenuto nel nodo sia presente per poi rimuoverlo.
- **Clear**: svuota la lista eliminando dalla memoria (heap) tutti i nodi della lista.
- **Size**: ritorna la dimensione della lista, cioè il numero degli elementi salvati.
- **isEmpty**: ritorna un valore booleano che indica se il set è vuoto o meno.
- **Contains**: verifica se il nodo passato come parametro è presente nella lista utilizzando il funtore equals per il confronto.
- **Operator[]**: riceve in input un indice e ritorna il riferimento del valore che si trova in quella posizione. Se l'indice è fuori dal range viene lanciata un'eccezione con un messaggio in cui si avverte che non è stato possibile leggere il dato.
- **Operator==**: effettua il confronto tra il set corrente e il set passato come parametro e ritorna true se entrambi sono vuoti o se contengono gli stessi valori, ritorna false se il set corrente ha una dimensione diversa dal set passato come parametro.
- **Operator<<**: riceve in input lo stream di output e un set e spedisce quest'ultimo sullo stream iterando sui suoi elementi.
- **Begin**: ritorna un iteratore costante che punta al primo nodo della lista
- **End**: ritorna un iteratore costante che punta al nodo successivo all'ultimo nodo della lista, quindi un nodo nullo.
- **Filter\_out**: riceve in input un set e un predicato e ritorna un nuovo set contenente i valori che soddisfano il funtore predicato. Come per il copy constructor e il costruttore templatato anche qui va usato il blocco try catch per lo stesso motivo.

- **Operator+:** riceve in input due set e ritorna un nuovo set che contiene gli elementi di entrambi i set (unione insiemistica). Anche qui se vi è una cattiva allocazione della memoria utilizzo un blocco try catch per catturare l'eccezione e svuotare il nuovo set (per evitare leak di memoria) e rilanciare l'eccezione al chiamante.
- **Operator-:** riceve in input due set e ritorna un nuovo set che contiene gli elementi comuni a entrambi i set (intersezione insiemistica). Valgono le stesse considerazioni descritte nell'operator+ per quanto riguarda il blocco try catch.

Per il tipo di struttura dati che implementa la classe Set, quest'ultima dispone di un iteratore costante di tipo forward con i relativi metodi.

## Test Main

Il file main contiene:

- una struttura point che implementa un punto
- una classe Concessionaria che utilizza un Set di tipo Auto
- una serie di funtori predicato applicati sui tipi int, string, point e set

I test eseguiti sono:

- test di tutte le funzioni della classe su un set di tipo point
- test di tutte le funzioni della classe su un set di tipo string
- test di tutte le funzioni della classe su un set di tipo int
- test di tutti i metodi della classe const\_iterator
- test dei metodi fondamentali su set di tipo point
- test dell'operator[] su un set di tipo point
- test dell'operator== quadre su un set di tipo point
- test della funzione clear su un set generico
- test su un set costante di tipo point
- test della funzione filter\_out su set di tipo string, point, int e set di tipo point