

Liste e Alberi Veicolo

1.0

Generato da Doxygen 1.9.1

1 Liste e Alberi per oggetti Veicolo	1
1.1 09/06/2021	1
1.2 Analisi del problema	1
1.3 Considerazioni progettuali	2
1.4 Classi utilizzate	2
2 Indice della gerarchia	3
2.1 Gerarchia delle classi	3
3 Indice dei tipi composti	5
3.1 Elenco dei tipi composti	5
4 Indice dei file	7
4.1 Elenco dei file	7
5 Documentazione delle classi	9
5.1 Riferimenti per la classe Auto	9
5.1.1 Documentazione dei costruttori e dei distruttori	10
5.1.1.1 Auto() [1/2]	10
5.1.1.2 Auto() [2/2]	10
5.1.2 Documentazione delle funzioni membro	10
5.1.2.1 getCavalli()	10
5.1.2.2 getMarca()	11
5.1.2.3 operator<()	11
5.1.2.4 operator==(())	11
5.1.2.5 operator>()	11
5.1.2.6 setCavalli()	11
5.1.2.7 setMarca()	11
5.1.2.8 toString()	11
5.1.3 Documentazione dei friend e delle funzioni collegate	12
5.1.3.1 operator<<	12
5.1.4 Documentazione dei membri dato	12
5.1.4.1 Cavalli	12
5.1.4.2 Marca	12
5.2 Riferimenti per la classe Barca	12
5.2.1 Documentazione dei costruttori e dei distruttori	13
5.2.1.1 Barca() [1/2]	13
5.2.1.2 Barca() [2/2]	14
5.2.2 Documentazione delle funzioni membro	14
5.2.2.1 getCavalli()	14
5.2.2.2 getMarca()	14
5.2.2.3 operator<()	14
5.2.2.4 operator==(())	14
5.2.2.5 operator>()	14

5.2.2.6 setCavalli()	14
5.2.2.7 setMarca()	15
5.2.2.8 toString()	15
5.2.3 Documentazione dei friend e delle funzioni collegate	15
5.2.3.1 operator<<	15
5.2.4 Documentazione dei membri dato	15
5.2.4.1 Cavalli	15
5.2.4.2 Marca	15
5.3 Riferimenti per la classe Lista	16
5.3.1 Documentazione dei costruttori e dei distruttori	16
5.3.1.1 Lista()	16
5.3.2 Documentazione delle funzioni membro	16
5.3.2.1 GetData()	16
5.3.2.2 Gethead()	16
5.3.2.3 Getnil()	16
5.3.2.4 Gettail()	17
5.3.2.5 Setdata()	17
5.3.2.6 Sethead()	17
5.3.2.7 Setnil()	17
5.3.2.8 Settail()	17
5.4 Riferimenti per la classe ListAuto	17
5.4.1 Documentazione dei costruttori e dei distruttori	18
5.4.1.1 ListAuto()	18
5.4.2 Documentazione delle funzioni membro	18
5.4.2.1 getData()	18
5.4.2.2 getHead()	18
5.4.2.3 getNil()	18
5.4.2.4 getTail()	19
5.4.2.5 insert()	19
5.4.2.6 isEmpty()	19
5.4.2.7 print()	19
5.4.2.8 setHead()	19
5.4.2.9 setNil()	19
5.4.2.10 setTail()	19
5.5 Riferimenti per la classe ListBarca	20
5.5.1 Documentazione dei costruttori e dei distruttori	20
5.5.1.1 ListBarca()	20
5.5.2 Documentazione delle funzioni membro	20
5.5.2.1 getData()	20
5.5.2.2 getHead()	20
5.5.2.3 getNil()	21
5.5.2.4 getTail()	21

5.5.2.5 insert()	21
5.5.2.6 isEmpty()	21
5.5.2.7 print()	21
5.5.2.8 setHead()	21
5.5.2.9 setNil()	21
5.5.2.10 setTail()	22
5.6 Riferimenti per la classe ListMoto	22
5.6.1 Documentazione dei costruttori e dei distruttori	22
5.6.1.1 ListMoto()	22
5.6.2 Documentazione delle funzioni membro	22
5.6.2.1 getData()	22
5.6.2.2 getHead()	23
5.6.2.3 getNil()	23
5.6.2.4 getTail()	23
5.6.2.5 insert()	23
5.6.2.6 isEmpty()	23
5.6.2.7 print()	23
5.6.2.8 setHead()	23
5.6.2.9 setNil()	24
5.6.2.10 setTail()	24
5.7 Riferimenti per la classe Moto	24
5.7.1 Documentazione dei costruttori e dei distruttori	25
5.7.1.1 Moto() [1/2]	25
5.7.1.2 Moto() [2/2]	25
5.7.2 Documentazione delle funzioni membro	25
5.7.2.1 getCavalli()	25
5.7.2.2 getMarca()	26
5.7.2.3 operator<()	26
5.7.2.4 operator==()	26
5.7.2.5 operator>()	26
5.7.2.6 setCavalli()	26
5.7.2.7 setMarca()	26
5.7.2.8 toString()	26
5.7.3 Documentazione dei friend e delle funzioni collegate	27
5.7.3.1 operator<<	27
5.7.4 Documentazione dei membri dato	27
5.7.4.1 Cavalli	27
5.7.4.2 Marca	27
5.8 Template per la classe Node< T >	27
5.8.1 Documentazione dei costruttori e dei distruttori	28
5.8.1.1 Node()	28
5.8.2 Documentazione delle funzioni membro	28

5.8.2.1 getKey()	28
5.8.2.2 getLeft()	28
5.8.2.3 getParent()	28
5.8.2.4 getRight()	28
5.8.2.5 setKey()	29
5.8.2.6 setLeft()	29
5.8.2.7 setParent()	29
5.8.2.8 setRight()	29
5.8.3 Documentazione dei friend e delle funzioni collegate	29
5.8.3.1 operator<<	29
5.9 Template per la classe Nodo< T >	30
5.9.1 Documentazione dei costruttori e dei distruttori	30
5.9.1.1 Nodo() [1/2]	30
5.9.1.2 Nodo() [2/2]	30
5.9.2 Documentazione delle funzioni membro	30
5.9.2.1 getData()	30
5.9.2.2 getNext()	31
5.9.2.3 getPrev()	31
5.9.2.4 print()	31
5.9.2.5 setData()	31
5.9.2.6 setNext()	31
5.9.2.7 setPrev()	31
5.10 Template per la classe Tree< T >	32
5.10.1 Documentazione dei costruttori e dei distruttori	32
5.10.1.1 Tree()	32
5.10.2 Documentazione delle funzioni membro	32
5.10.2.1 canc() [1/2]	32
5.10.2.2 canc() [2/2]	32
5.10.2.3 inOrder()	33
5.10.2.4 ins()	33
5.10.2.5 postOrder()	33
5.10.2.6 preOrder()	33
5.11 Riferimenti per la classe Veicolo	33
5.11.1 Documentazione dei costruttori e dei distruttori	34
5.11.1.1 Veicolo() [1/2]	34
5.11.1.2 Veicolo() [2/2]	34
5.11.2 Documentazione delle funzioni membro	34
5.11.2.1 getCavalli()	34
5.11.2.2 getMarca()	35
5.11.2.3 operator<()	35
5.11.2.4 operator==(())	35
5.11.2.5 operator>()	35

5.11.2.6 setCavalli()	35
5.11.2.7 setMarca()	35
5.11.2.8 toString()	35
5.11.3 Documentazione dei friend e delle funzioni collegate	36
5.11.3.1 operator<<	36
5.11.4 Documentazione dei membri dato	36
5.11.4.1 Cavalli	36
5.11.4.2 Marca	36
6 Documentazione dei file	37
6.1 Riferimenti per il file Considerazioni.md	37
6.2 Riferimenti per il file include/Auto.h	37
6.3 Riferimenti per il file include/Barca.h	38
6.4 Riferimenti per il file include/Lista.h	39
6.5 Riferimenti per il file include/ListAuto.h	39
6.6 Riferimenti per il file include/ListBarca.h	40
6.7 Riferimenti per il file include/ListMoto.h	41
6.8 Riferimenti per il file include/Moto.h	42
6.9 Riferimenti per il file include/Node.h	44
6.10 Riferimenti per il file include/Nodo.h	44
6.11 Riferimenti per il file include/Tree.h	45
6.12 Riferimenti per il file include/Veicolo.h	46
6.13 Riferimenti per il file main.cpp	47
6.13.1 Documentazione delle funzioni	47
6.13.1.1 canc()	47
6.13.1.2 cancelA()	48
6.13.1.3 cancelB()	48
6.13.1.4 cancelM()	48
6.13.1.5 loadA()	48
6.13.1.6 loadB()	48
6.13.1.7 loadM()	48
6.13.1.8 main()	49
6.13.1.9 print()	49
6.13.1.10 travasaA()	49
6.13.1.11 travasaB()	49
6.13.1.12 travasaM()	49
Indice analitico	51

Capitolo 1

Liste e Alberi per oggetti Veicolo

1.1 09/06/2021

1.2 Analisi del problema

Il testo richiede la progettazione e l'implementazione di una o più opportune strutture dati astratte (ADT) che consentano di:

- inserire elementi nella strutture dati appropriate al loro tipo di veicolo;
- cercare elementi nella struttura dati;
- cancellare elementi nella struttura dati sotto una cilindrata minima.
- stampare gli elementi seguendo una vista PostOrder
- stampare gli elementi seguendo una vista PreOrder
- stampare gli elementi seguendo una vista InOrder
- cercare il massimo nella struttura dati
- cercare il minimo nella struttura dati
- cercare il successore di un elemento
- cercare il predecessore di un elemento

Tra i requisiti, si nota che:

- la ricerca e la cancellazione avvengono in base a dei valori passati come parametri alle rispettive funzioni;
- i dati da inserire sono forniti in file di testo;
- la stampa della struttura dati [Tree](#) deve essere effettuata seguendo una vista tra: inorder, postorder, preorder.

1.3 Considerazioni progettuali

1. La struttura di un Binary Search **Tree** suggerisce l'utilizzo di una struttura dati che sia "percorribile" in entrambi i versi;
2. Il fatto che la struttura dati venga popolata a partire da un file di testo suggerisce che è opportuno utilizzare un'implementazione dinamica, che consenta quindi di variare il numero di elementi nella struttura stessa;
3. L'inserimento viene effettuato seguendo le proprietà dell'ordinamento parziale della struttura, poiché i nodi vanno inseriti a sinistra se la cilindrata della key del nodo da inserire è più piccola della cilindrata della key del nodo in cui siamo arrivati o a destra se più grande, cioè ogni nodo avrà a sinistra i figli con la cilindrata più piccola e a destra i figli con la cilindrata più grande;
4. Gli elementi da inserire sono dei veicoli di tipo diverso perciò opto a creare una classe che verrà ereditata, contenente i membri marca e cilindrata, dalle classi figlie auto, moto e barca;
5. Il fatto che l'inserimento e il travaso delle varie strutture dati sono indipendenti tra di loro ho deciso di utilizzare la programmazione multi-thread per alleggerire la complessità computazionale e svolgere queste operazioni in parallelo così da velocizzare l'esecuzione del programma.

Si decide, pertanto, di utilizzare una **Lista** per ogni veicolo adattata all'esigenze della struttura dati, con un puntatore alla root; e dove gli elementi che la compongono hanno 3 puntatori per i BST che conterranno i veicoli travasati e una lista per ogni veicolo per contenere i dati iniziali.

1.4 Classi utilizzate

- **Node**: classe template che rappresenta un nodo del BST(Tree) con figlio sinistro(left) , figlio destro(right) e genitore(parent).
 - Qui vengono implementate le funzioni base di: creazione del figlio, settare figlio destro , figlio sinistro e parent, e restituire i puntatori a left, right e parent oltre all'overload del cout(operator<<) per poter effettuare la stampa dell'oggetto di tipo Node<T>.
- **Tree**: classe template che rappresenta una lista adattata alla struttura dati di un Binary Search **Tree**, con un puntatore alla root dell'albero.
 - Qui vengono implementate le funzioni richieste: le varie viste per la stampa, le funzioni di ricerca e le funzioni di inserimento e cancellazione.
- **Auto, Moto, Barca**: classe che definisce l'oggetto come oggetto di tipo veicolo.
 - Qui mi occupo di inizializzare l'oggetto e di gestirne la stampa.
- **ListAuto, ListMoto, ListBarca**: classi che rappresentano le liste dei veicoli. -Qui gestisco l'inserimento dei valori e la stampa della lista.
- **Veicolo**: classe che si occupa di definire i membri delle classi che la deriveranno.
- **Nodo**: classe che rappresenta un nodo della lista dei veicoli.

Capitolo 2

Indice della gerarchia

2.1 Gerarchia delle classi

Questo elenco di ereditarietà è ordinato approssimativamente, ma non completamente, in ordine alfabetico:

Lista	16
ListAuto	17
ListBarca	20
ListMoto	22
Node< T >	27
Nodo< T >	30
Nodo< Auto >	30
Nodo< Barca >	30
Nodo< Moto >	30
Tree< T >	32
Veicolo	33
Auto	9
Barca	12
Moto	24

Capitolo 3

Indice dei tipi composti

3.1 Elenco dei tipi composti

Queste sono le classi, le struct, le union e le interfacce con una loro breve descrizione:

Auto	9
Barca	12
Lista	16
ListAuto	17
ListBarca	20
ListMoto	22
Moto	24
Node< T >	27
Nodo< T >	30
Tree< T >	32
Veicolo	33

Capitolo 4

Indice dei file

4.1 Elenco dei file

Questo è un elenco di tutti i file con una loro breve descrizione:

main.cpp	47
include/Auto.h	37
include/Barca.h	38
include/Lista.h	39
include/ListAuto.h	39
include/ListBarca.h	40
include/ListMoto.h	41
include/Moto.h	42
include/Node.h	44
include/Nodo.h	44
include/Tree.h	45
include/Veicolo.h	46

Capitolo 5

Documentazione delle classi

5.1 Riferimenti per la classe Auto

```
#include <Auto.h>
```

Diagramma delle classi per Auto

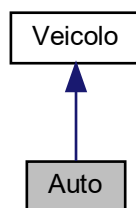
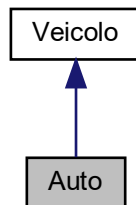


Diagramma di collaborazione per Auto:



Membri pubblici

- [Auto](#) (string [Marca](#), int [Cavalli](#))
- [Auto](#) ()
- string [toString](#) () const
- string [getMarca](#) ()
- void [setMarca](#) (string val)
- int [getCavalli](#) ()
- void [setCavalli](#) (int val)
- bool [operator<](#) (const [Veicolo](#) &v)
- bool [operator>](#) (const [Veicolo](#) &v)
- bool [operator==](#) (const [Veicolo](#) &v)

Attributi protetti

- string [Marca](#)
- int [Cavalli](#)

Friend

- ostream & [operator<<](#) (ostream &out, [Auto](#) &a)

5.1.1 Documentazione dei costruttori e dei distruttori

5.1.1.1 [Auto\(\)](#) [1/2]

```
Auto::Auto (
    string Marca,
    int Cavalli ) [inline]
```

5.1.1.2 [Auto\(\)](#) [2/2]

```
Auto::Auto ( ) [inline]
```

5.1.2 Documentazione delle funzioni membro

5.1.2.1 [getCavalli\(\)](#)

```
int Veicolo::getCavalli ( ) [inline], [inherited]
```

5.1.2.2 getMarca()

```
string Veicolo::getMarca ( ) [inline], [inherited]
```

5.1.2.3 operator<()

```
bool Veicolo::operator< (
    const Veicolo & v ) [inline], [inherited]
```

5.1.2.4 operator==()

```
bool Veicolo::operator== (
    const Veicolo & v ) [inline], [inherited]
```

5.1.2.5 operator>()

```
bool Veicolo::operator> (
    const Veicolo & v ) [inline], [inherited]
```

5.1.2.6 setCavalli()

```
void Veicolo::setCavalli (
    int val ) [inline], [inherited]
```

5.1.2.7 setMarca()

```
void Veicolo::setMarca (
    string val ) [inline], [inherited]
```

5.1.2.8 toString()

```
string Auto::toString ( ) const [inline], [virtual]
```

Funzione di stampa da usare nell overload dell operatore <<

Implementa [Veicolo](#).

5.1.3 Documentazione dei friend e delle funzioni collegate

5.1.3.1 operator<<

```
ostream& operator<< (
    ostream & out,
    Auto & a ) [friend]
```

5.1.4 Documentazione dei membri dato

5.1.4.1 Cavalli

```
int Veicolo::Cavalli [protected], [inherited]
```

5.1.4.2 Marca

```
string Veicolo::Marca [protected], [inherited]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- include/[Auto.h](#)

5.2 Riferimenti per la classe Barca

```
#include <Barca.h>
```

Diagramma delle classi per Barca

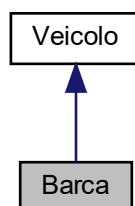
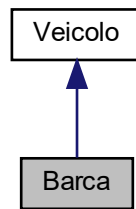


Diagramma di collaborazione per Barca:



Membri pubblici

- `Barca` (string `Marca`, int `Cavalli`)
- `Barca` ()
- string `toString` () const
- string `getMarca` ()
- void `setMarca` (string val)
- int `getCavalli` ()
- void `setCavalli` (int val)
- bool `operator<` (const `Veicolo` &v)
- bool `operator>` (const `Veicolo` &v)
- bool `operator==` (const `Veicolo` &v)

Attributi protetti

- string `Marca`
- int `Cavalli`

Friend

- ostream & `operator<<` (ostream &out, `Barca` &b)

5.2.1 Documentazione dei costruttori e dei distruttori

5.2.1.1 `Barca()` [1/2]

```
Barca::Barca (  
    string Marca,  
    int Cavalli ) [inline]
```

5.2.1.2 Barca() [2/2]

```
Barca::Barca ( ) [inline]
```

5.2 Documentazione delle funzioni membro

5.2.2.1 getCavalli()

```
int Veicolo::getCavalli ( ) [inline], [inherited]
```

5.2.2.2 getMarca()

```
string Veicolo::getMarca ( ) [inline], [inherited]
```

5.2.2.3 operator<()

```
bool Veicolo::operator< (
    const Veicolo & v ) [inline], [inherited]
```

5.2.2.4 operator==()

```
bool Veicolo::operator==(
    const Veicolo & v ) [inline], [inherited]
```

5.2.2.5 operator>()

```
bool Veicolo::operator> (
    const Veicolo & v ) [inline], [inherited]
```

5.2.2.6 setCavalli()

```
void Veicolo::setCavalli (
    int val ) [inline], [inherited]
```

5.2.2.7 setMarca()

```
void Veicolo::setMarca (
    string val ) [inline], [inherited]
```

5.2.2.8 toString()

```
string Barca::toString ( ) const [inline], [virtual]
```

Funzione di stampa da usare nell overload dell operatore <<

Implementa [Veicolo](#).

5.2.3 Documentazione dei friend e delle funzioni collegate

5.2.3.1 operator<<

```
ostream& operator<< (
    ostream & out,
    Barca & b ) [friend]
```

5.2.4 Documentazione dei membri dato

5.2.4.1 Cavalli

```
int Veicolo::Cavalli [protected], [inherited]
```

5.2.4.2 Marca

```
string Veicolo::Marca [protected], [inherited]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- include/[Barca.h](#)

5.3 Riferimenti per la classe Lista

```
#include <Lista.h>
```

Membri pubblici

- [Lista](#) ()
- [T](#) * [Getdata](#) ()
- void [Setdata](#) ([T](#) *val)
- [Node](#)< [T](#) > * [Gettail](#) ()
- void [Settail](#) ([Node](#)< [T](#) > *val)
- [Node](#)< [T](#) > * [Gethead](#) ()
- void [Sethead](#) ([Node](#)< [T](#) > *val)
- [Node](#)< [T](#) > * [Getnil](#) ()
- void [Setnil](#) ([Node](#)< [T](#) > *val)

5.3.1 Documentazione dei costruttori e dei distruttori

5.3.1.1 Lista()

```
Lista::Lista ( ) [inline]
```

5.3.2 Documentazione delle funzioni membro

5.3.2.1 Getdata()

```
T* Lista::Getdata ( ) [inline]
```

5.3.2.2 Gethead()

```
Node<T>* Lista::Gethead ( ) [inline]
```

5.3.2.3 Getnil()

```
Node<T>* Lista::Getnil ( ) [inline]
```


5.3.2.4 Gettail()

```
Node<T>* Lista::Gettail ( ) [inline]
```

5.3.2.5 Setdata()

```
void Lista::Setdata (
    T * val ) [inline]
```

5.3.2.6 Sethead()

```
void Lista::Sethead (
    Node< T > * val ) [inline]
```

5.3.2.7 Setnil()

```
void Lista::Setnil (
    Node< T > * val ) [inline]
```

5.3.2.8 Settail()

```
void Lista::Settail (
    Node< T > * val ) [inline]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- include/[Lista.h](#)

5.4 Riferimenti per la classe ListAuto

```
#include <ListAuto.h>
```

Membri pubblici

- `ListAuto ()`
- `Nodo< Auto > * getHead () const`
- `void setHead (Nodo< Auto > *val)`
- `Nodo< Auto > * getTail () const`
- `void setTail (Nodo< Auto > *val)`
- `Nodo< Auto > * getNil () const`
- `void setNil (Nodo< Auto > *val)`
- `bool isEmpty ()`
- `void insert (Auto value)`
- `void print ()`
- `Auto getData (Nodo< Auto > *tmp)`

5.4.1 Documentazione dei costruttori e dei distruttori

5.4.1.1 ListAuto()

```
ListAuto::ListAuto ( ) [inline]
```

5.4.2 Documentazione delle funzioni membro

5.4.2.1 getData()

```
Auto ListAuto::getData (
    Nodo< Auto > * tmp ) [inline]
```

5.4.2.2 getHead()

```
Nodo<Auto>* ListAuto::getHead ( ) const [inline]
```

5.4.2.3 getNil()

```
Nodo<Auto>* ListAuto::getNil ( ) const [inline]
```

5.4.2.4 getTail()

```
Nodo<Auto>* ListAuto::getTail ( ) const [inline]
```

5.4.2.5 insert()

```
void ListAuto::insert (
    Auto value ) [inline]
```

Funzione per l'inserimento

5.4.2.6 isEmpty()

```
bool ListAuto::isEmpty ( ) [inline]
```

5.4.2.7 print()

```
void ListAuto::print ( ) [inline]
```

5.4.2.8 setHead()

```
void ListAuto::setHead (
    Nodo< Auto > * val ) [inline]
```

5.4.2.9 setNil()

```
void ListAuto::setNil (
    Nodo< Auto > * val ) [inline]
```

5.4.2.10 setTail()

```
void ListAuto::setTail (
    Nodo< Auto > * val ) [inline]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- include/[ListAuto.h](#)

5.5 Riferimenti per la classe ListBarca

```
#include <ListBarca.h>
```

Membri pubblici

- [ListBarca](#) ()
- [Nodo](#)< [Barca](#) > * [getHead](#) () const
- void [setHead](#) ([Nodo](#)< [Barca](#) > *val)
- [Nodo](#)< [Barca](#) > * [getTail](#) () const
- void [setTail](#) ([Nodo](#)< [Barca](#) > *val)
- [Nodo](#)< [Barca](#) > * [getNil](#) () const
- void [setNil](#) ([Nodo](#)< [Barca](#) > *val)
- bool [isEmpty](#) ()
- void [insert](#) ([Barca](#) value)
- void [print](#) ()
- [Barca](#) [getData](#) ([Nodo](#)< [Barca](#) > *tmp)

5.5.1 Documentazione dei costruttori e dei distruttori

5.5.1.1 ListBarca()

```
ListBarca::ListBarca ( ) [inline]
```

5.5.2 Documentazione delle funzioni membro

5.5.2.1 getData()

```
Barca ListBarca::getData (
    Nodo< Barca > * tmp ) [inline]
```

5.5.2.2 getHead()

```
Nodo<Barca>* ListBarca::getHead ( ) const [inline]
```

5.5.2.3 getNil()

```
Nodo<Barca>* ListBarca::getNil ( ) const [inline]
```

5.5.2.4 getTail()

```
Nodo<Barca>* ListBarca::getTail ( ) const [inline]
```

5.5.2.5 insert()

```
void ListBarca::insert (
    Barca value ) [inline]
```

Funzione per l'inserimento

5.5.2.6 isEmpty()

```
bool ListBarca::isEmpty ( ) [inline]
```

5.5.2.7 print()

```
void ListBarca::print ( ) [inline]
```

5.5.2.8 setHead()

```
void ListBarca::setHead (
    Nodo< Barca > * val ) [inline]
```

5.5.2.9 setNil()

```
void ListBarca::setNil (
    Nodo< Barca > * val ) [inline]
```

5.5.2.10 setTail()

```
void ListBarca::setTail (
    Nodo< Barca > * val ) [inline]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- include/[ListBarca.h](#)

5.6 Riferimenti per la classe ListMoto

```
#include <ListMoto.h>
```

Membri pubblici

- [ListMoto](#) ()
- [Nodo< Moto > * getHead](#) () const
- void [setHead](#) ([Nodo< Moto > *val](#))
- [Nodo< Moto > * getTail](#) () const
- void [setTail](#) ([Nodo< Moto > *val](#))
- [Nodo< Moto > * getNil](#) () const
- void [setNil](#) ([Nodo< Moto > *val](#))
- bool [isEmpty](#) ()
- void [insert](#) ([Moto](#) value)
- void [print](#) ()
- [Moto](#) [getData](#) ([Nodo< Moto > *tmp](#))

5.6.1 Documentazione dei costruttori e dei distruttori

5.6.1.1 ListMoto()

```
ListMoto::ListMoto ( ) [inline]
```

5.6.2 Documentazione delle funzioni membro

5.6.2.1 getData()

```
Moto ListMoto::getData (
    Nodo< Moto > \* tmp ) [inline]
```

5.6.2.2 getHead()

```
Nodo<Moto>* ListMoto::getHead ( ) const [inline]
```

5.6.2.3 getNil()

```
Nodo<Moto>* ListMoto::getNil ( ) const [inline]
```

5.6.2.4 getTail()

```
Nodo<Moto>* ListMoto::getTail ( ) const [inline]
```

5.6.2.5 insert()

```
void ListMoto::insert (
    Moto value ) [inline]
```

Funzione per l'inserimento

5.6.2.6 isEmpty()

```
bool ListMoto::isEmpty ( ) [inline]
```

5.6.2.7 print()

```
void ListMoto::print ( ) [inline]
```

5.6.2.8 setHead()

```
void ListMoto::setHead (
    Nodo< Moto > * val ) [inline]
```

5.6.2.9 setNil()

```
void ListMoto::setNil (
    Nodo< Moto > * val ) [inline]
```

5.6.2.10 setTail()

```
void ListMoto::setTail (
    Nodo< Moto > * val ) [inline]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- include/[ListMoto.h](#)

5.7 Riferimenti per la classe Moto

```
#include <Moto.h>
```

Diagramma delle classi per Moto

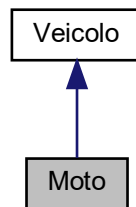
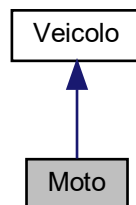


Diagramma di collaborazione per Moto:



Membri pubblici

- `Moto` (string `Marca`, int `Cavalli`)
- `Moto` ()
- string `toString` () const
- string `getMarca` ()
- void `setMarca` (string val)
- int `getCavalli` ()
- void `setCavalli` (int val)
- bool `operator<` (const `Veicolo` &v)
- bool `operator>` (const `Veicolo` &v)
- bool `operator==` (const `Veicolo` &v)

Attributi protetti

- string `Marca`
- int `Cavalli`

Friend

- ostream & `operator<<` (ostream &out, `Moto` &m)

5.7.1 Documentazione dei costruttori e dei distruttori

5.7.1.1 `Moto()` [1/2]

```
Moto::Moto (
    string Marca,
    int Cavalli ) [inline]
```

5.7.1.2 `Moto()` [2/2]

```
Moto::Moto ( ) [inline]
```

5.7.2 Documentazione delle funzioni membro

5.7.2.1 `getCavalli()`

```
int Veicolo::getCavalli ( ) [inline], [inherited]
```

5.7.2.2 getMarca()

```
string Veicolo::getMarca ( ) [inline], [inherited]
```

5.7.2.3 operator<()

```
bool Veicolo::operator< (
    const Veicolo & v ) [inline], [inherited]
```

5.7.2.4 operator==()

```
bool Veicolo::operator== (
    const Veicolo & v ) [inline], [inherited]
```

5.7.2.5 operator>()

```
bool Veicolo::operator> (
    const Veicolo & v ) [inline], [inherited]
```

5.7.2.6 setCavalli()

```
void Veicolo::setCavalli (
    int val ) [inline], [inherited]
```

5.7.2.7 setMarca()

```
void Veicolo::setMarca (
    string val ) [inline], [inherited]
```

5.7.2.8 toString()

```
string Moto::toString ( ) const [inline], [virtual]
```

Funzione di stampa da usare nell overload dell operatore <<

Implementa [Veicolo](#).

5.7.3 Documentazione dei friend e delle funzioni collegate

5.7.3.1 operator<<

```
ostream& operator<< (
    ostream & out,
    Moto & m ) [friend]
```

5.7.4 Documentazione dei membri dato

5.7.4.1 Cavalli

```
int Veicolo::Cavalli [protected], [inherited]
```

5.7.4.2 Marca

```
string Veicolo::Marca [protected], [inherited]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- include/Moto.h

5.8 Template per la classe Node< T >

```
#include <Node.h>
```

Membri pubblici

- Node (T key)
- void setLeft (Node< T > *left)
- void setRight (Node< T > *right)
- void setParent (Node< T > *parent)
- void setKey (T key)
- Node< T > * getLeft ()
- Node< T > * getRight ()
- Node< T > * getParent ()
- T getKey ()

Friend

- ostream & `operator<<` (ostream &out, `Node`< T > &n)

5.8.1 Documentazione dei costruttori e dei distruttori

5.8.1.1 Node()

```
template<class T >
Node< T >::Node (
    T key ) [inline]
```

Costruttore che riceve in ingresso una key di tipo generico

5.8.2 Documentazione delle funzioni membro

5.8.2.1 getKey()

```
template<class T >
T Node< T >::getKey ( ) [inline]
```

Metodo che ritorna la key

5.8.2.2 getLeft()

```
template<class T >
Node<T>* Node< T >::getLeft ( ) [inline]
```

Metodo che ritorna il puntatore al figlio sinistro

5.8.2.3 getParent()

```
template<class T >
Node<T>* Node< T >::getParent ( ) [inline]
```

Metodo che ritorna il puntatore al parent

5.8.2.4 getRight()

```
template<class T >
Node<T>* Node< T >::getRight ( ) [inline]
```

Metodo che ritorna il puntatore al figlio destro

5.8.2.5 setKey()

```
template<class T >
void Node< T >::setKey (
    T key ) [inline]
```

Metodo per impostare la key

5.8.2.6 setLeft()

```
template<class T >
void Node< T >::setLeft (
    Node< T > * left ) [inline]
```

Metodo per impostare il figlio sinistro

5.8.2.7 setParent()

```
template<class T >
void Node< T >::setParent (
    Node< T > * parent ) [inline]
```

Metodo per impostare il parent

5.8.2.8 setRight()

```
template<class T >
void Node< T >::setRight (
    Node< T > * right ) [inline]
```

Metodo per impostare il figlio dietro

5.8.3 Documentazione dei friend e delle funzioni collegate

5.8.3.1 operator<<

```
template<class T >
ostream& operator<< (
    ostream & out,
    Node< T > & n ) [friend]
```

Overriding dell'operatore << del cout per stampare un oggetto di tipo Node<T> stampando la key del nodo

La documentazione per questa classe è stata generata a partire dal seguente file:

- include/Node.h

5.9 Template per la classe `Nodo< T >`

```
#include <Nodo.h>
```

Membri pubblici

- `Nodo` (T data)
- `Nodo` ()
- T `getData` ()
- void `setData` (T val)
- `Nodo< T > * getNext` ()
- void `setNext` (`Nodo< T > *val`)
- `Nodo< T > * getPrev` ()
- void `setPrev` (`Nodo< T > *val`)
- void `print` ()

5.9.1 Documentazione dei costruttori e dei distruttori

5.9.1.1 `Nodo()` [1/2]

```
template<class T >  
Nodo< T >::Nodo (   
    T data ) [inline]
```

5.9.1.2 `Nodo()` [2/2]

```
template<class T >  
Nodo< T >::Nodo ( ) [inline]
```

5.9.2 Documentazione delle funzioni membro

5.9.2.1 `getData()`

```
template<class T >  
T Nodo< T >::getData ( ) [inline]
```

5.9.2.2 getNext()

```
template<class T >
Nodo<T>* Nodo< T >::getNext ( ) [inline]
```

5.9.2.3 getPrev()

```
template<class T >
Nodo<T>* Nodo< T >::getPrev ( ) [inline]
```

5.9.2.4 print()

```
template<class T >
void Nodo< T >::print ( ) [inline]
```

5.9.2.5 setData()

```
template<class T >
void Nodo< T >::setData (
    T val ) [inline]
```

5.9.2.6 setNext()

```
template<class T >
void Nodo< T >::setNext (
    Nodo< T > * val ) [inline]
```

5.9.2.7 setPrev()

```
template<class T >
void Nodo< T >::setPrev (
    Nodo< T > * val ) [inline]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- include/[Nodo.h](#)

5.10 Template per la classe Tree< T >

```
#include <Tree.h>
```

Membri pubblici

- `Tree ()`
- `void ins (T key)`
- `void postOrder ()`
- `void preOrder ()`
- `void inOrder ()`
- `void canc (T key)`
- `void canc (int key)`

5.10.1 Documentazione dei costruttori e dei distruttori

5.10.1.1 Tree()

```
template<class T >  
Tree< T >::Tree ( ) [inline]
```

Costruttore di default

5.10.2 Documentazione delle funzioni membro

5.10.2.1 canc() [1/2]

```
template<class T >  
void Tree< T >::canc (   
    int key ) [inline]
```

5.10.2.2 canc() [2/2]

```
template<class T >  
void Tree< T >::canc (   
    T key ) [inline]
```


5.10.2.3 inOrder()

```
template<class T >
void Tree< T >::inOrder ( ) [inline]
```

5.10.2.4 ins()

```
template<class T >
void Tree< T >::ins (
    T key ) [inline]
```

Funzione per l'inserimento di un nodo tramite passaggio di una chiave di tipo generico T se l'albero è vuoto allora il nodo creato diventerà root, nel caso in cui l'albero non sia vuoto, allora si farà confronto per vedere se il nuovo nodo deve essere inserito nel sottoalbero destro o nel sottoalbero sinistro

5.10.2.5 postOrder()

```
template<class T >
void Tree< T >::postOrder ( ) [inline]
```

5.10.2.6 preOrder()

```
template<class T >
void Tree< T >::preOrder ( ) [inline]
```

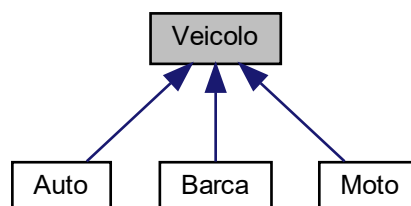
La documentazione per questa classe è stata generata a partire dal seguente file:

- include/Tree.h

5.11 Riferimenti per la classe Veicolo

```
#include <Veicolo.h>
```

Diagramma delle classi per Veicolo



Membri pubblici

- `Veicolo` (string `Marca`, int `Cavalli`)
- `Veicolo` ()
- string `getMarca` ()
- void `setMarca` (string val)
- int `getCavalli` ()
- void `setCavalli` (int val)
- bool `operator<` (const `Veicolo` &v)
- bool `operator>` (const `Veicolo` &v)
- bool `operator==` (const `Veicolo` &v)
- virtual std::string `toString` () const =0

Attributi protetti

- string `Marca`
- int `Cavalli`

Friend

- std::ostream & `operator<<` (std::ostream &out, const `Veicolo` &v)

5.11.1 Documentazione dei costruttori e dei distruttori

5.11.1.1 `Veicolo()` [1/2]

```
Veicolo::Veicolo (
    string Marca,
    int Cavalli ) [inline]
```

5.11.1.2 `Veicolo()` [2/2]

```
Veicolo::Veicolo ( ) [inline]
```

5.11.2 Documentazione delle funzioni membro

5.11.2.1 `getCavalli()`

```
int Veicolo::getCavalli ( ) [inline]
```

5.11.2.2 getMarca()

```
string Veicolo::getMarca ( ) [inline]
```

5.11.2.3 operator<()

```
bool Veicolo::operator< (
    const Veicolo & v ) [inline]
```

5.11.2.4 operator==()

```
bool Veicolo::operator== (
    const Veicolo & v ) [inline]
```

5.11.2.5 operator>()

```
bool Veicolo::operator> (
    const Veicolo & v ) [inline]
```

5.11.2.6 setCavalli()

```
void Veicolo::setCavalli (
    int val ) [inline]
```

5.11.2.7 setMarca()

```
void Veicolo::setMarca (
    string val ) [inline]
```

5.11.2.8 toString()

```
virtual std::string Veicolo::toString ( ) const [pure virtual]
```

Implementato in [Moto](#), [Barca](#), e [Auto](#).

5.11.3 Documentazione dei friend e delle funzioni collegate

5.11.3.1 `operator<<`

```
std::ostream& operator<< (
    std::ostream & out,
    const Veicolo & v ) [friend]
```

5.11.4 Documentazione dei membri dato

5.11.4.1 Cavalli

```
int Veicolo::Cavalli [protected]
```

5.11.4.2 Marca

```
string Veicolo::Marca [protected]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- include/[Veicolo.h](#)

Capitolo 6

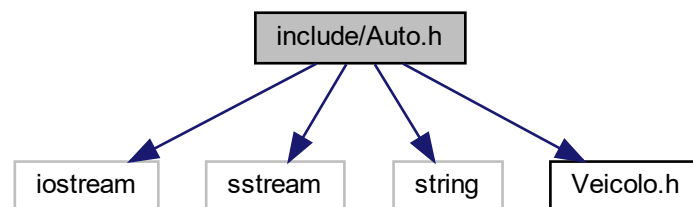
Documentazione dei file

6.1 Riferimenti per il file Considerazioni.md

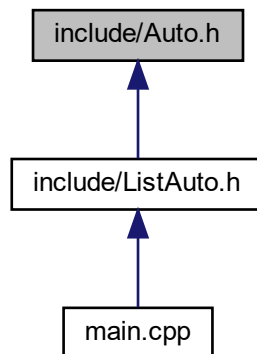
6.2 Riferimenti per il file include/Auto.h

```
#include <iostream>  
#include <sstream>  
#include <string>  
#include <Veicolo.h>
```

Grafo delle dipendenze di inclusione per Auto.h:



Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:



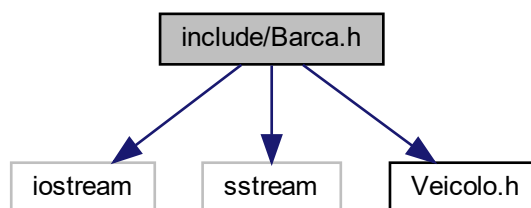
Composti

- class [Auto](#)

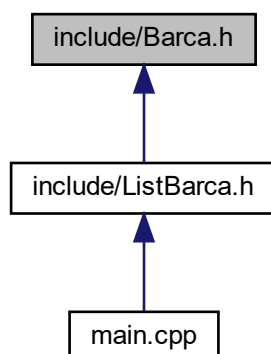
6.3 Riferimenti per il file `include/Barca.h`

```
#include <iostream>
#include <sstream>
#include <Veicolo.h>
```

Grafo delle dipendenze di inclusione per `Barca.h`:



Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:



Composti

- class [Barca](#)

6.4 Riferimenti per il file include/Lista.h

Composti

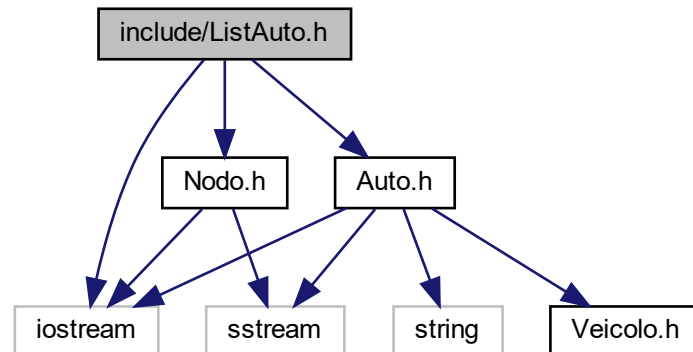
- class [Lista](#)

6.5 Riferimenti per il file include/ListAuto.h

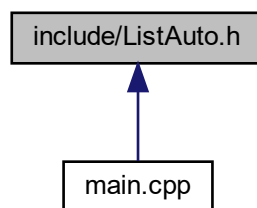
```
#include <iostream>
#include "Nodo.h"
```

```
#include "Auto.h"
```

Grafo delle dipendenze di inclusione per ListAuto.h:



Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:



Composti

- class [ListAuto](#)

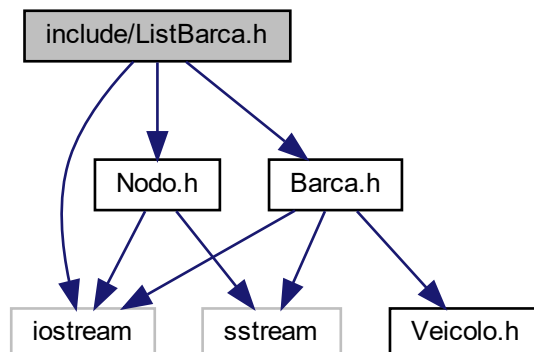
6.6 Riferimenti per il file include/ListBarca.h

```
#include <iostream>
#include "Nodo.h"
```

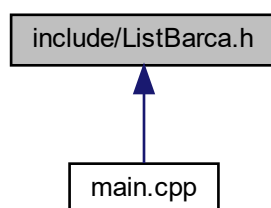


```
#include "Barca.h"
```

Grafo delle dipendenze di inclusione per ListBarca.h:



Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:



Composti

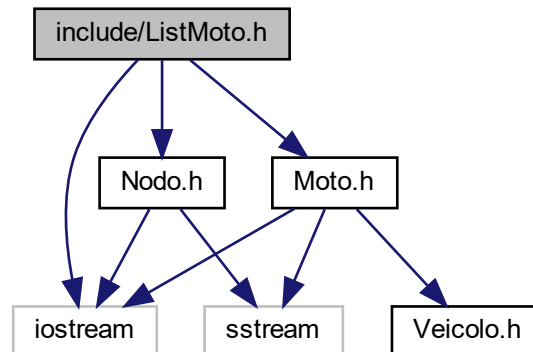
- class [ListBarca](#)

6.7 Riferimenti per il file include/ListMoto.h

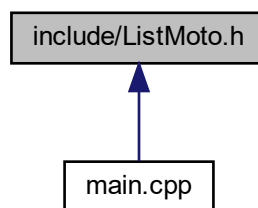
```
#include <iostream>
#include "Nodo.h"
```

```
#include "Moto.h"
```

Grafo delle dipendenze di inclusione per ListMoto.h:



Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:



Composti

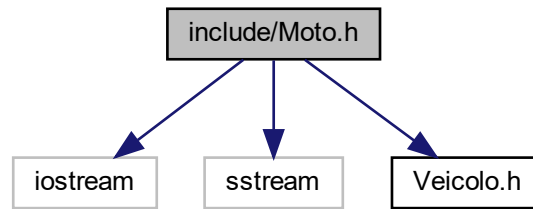
- class [ListMoto](#)

6.8 Riferimenti per il file include/Moto.h

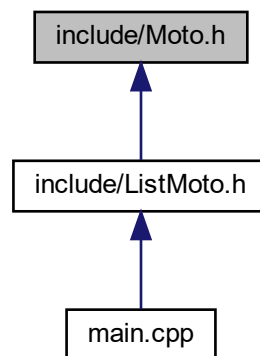
```
#include <iostream>
#include <sstream>
```

```
#include <Veicolo.h>
```

Grafo delle dipendenze di inclusione per Moto.h:



Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:

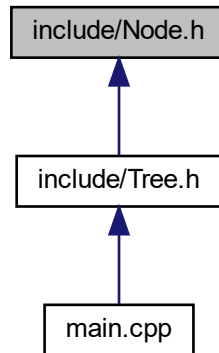


Composti

- class [Moto](#)

6.9 Riferimenti per il file include/Node.h

Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:



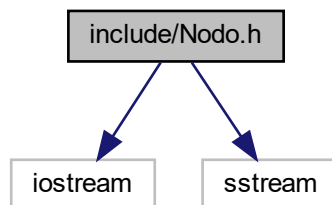
Composti

- class `Node< T >`

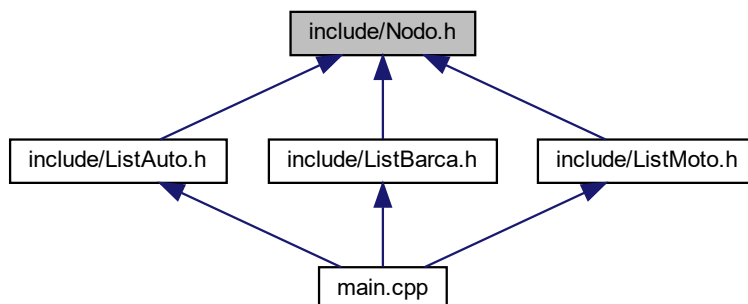
6.10 Riferimenti per il file include/Nodo.h

```
#include <iostream>
#include <sstream>
```

Grafo delle dipendenze di inclusione per Nodo.h:



Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:



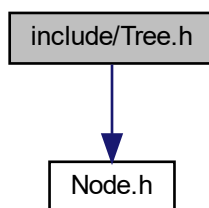
Composti

- class `Nodo< T >`

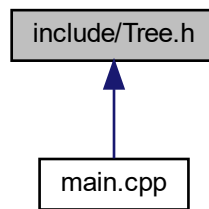
6.11 Riferimenti per il file include/Tree.h

```
#include "Node.h"
```

Grafo delle dipendenze di inclusione per Tree.h:



Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:

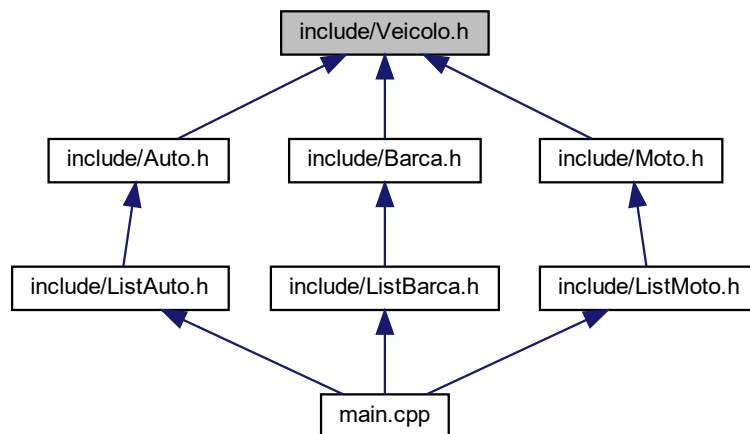


Composti

- class [Tree< T >](#)

6.12 Riferimenti per il file include/Veicolo.h

Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:



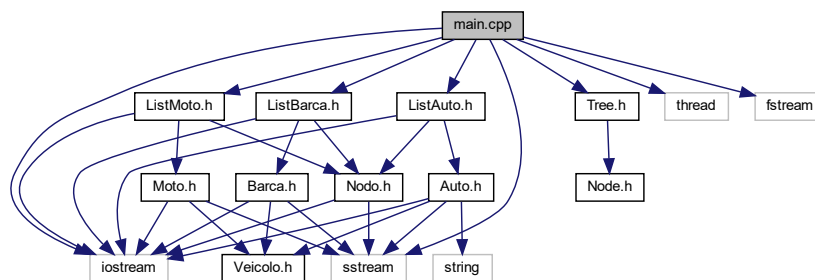
Composti

- class [Veicolo](#)

6.13 Riferimenti per il file main.cpp

```
#include <iostream>
#include "ListAuto.h"
#include "ListMoto.h"
#include "ListBarca.h"
#include "Tree.h"
#include <sstream>
#include <thread>
#include <fstream>
```

Grafo delle dipendenze di inclusione per main.cpp:



Funzioni

- void `loadA` (`ListAuto` &aList, string fileNameA)
- void `loadM` (`ListMoto` &mList, string fileNameM)
- void `loadB` (`ListBarca` &bList, string fileNameB)
- void `travasaA` (`ListAuto` &aList, `Tree`< `Auto` > &aTree)
- void `travasaM` (`ListMoto` &mList, `Tree`< `Moto` > &mTree)
- void `travasaB` (`ListBarca` &bList, `Tree`< `Barca` > &bTree)
- void `cancA` (int key, `Tree`< `Auto` > &aTree)
- void `cancM` (int key, `Tree`< `Moto` > &mTree)
- void `cancB` (int key, `Tree`< `Barca` > &bTree)
- void `print` (`ListAuto` &aList, `ListMoto` &mList, `ListBarca` &bList)
- void `canc` (`Tree`< `Auto` > &aTree, `Tree`< `Moto` > &mTree, `Tree`< `Barca` > &bTree)
- int `main` ()

6.13.1 Documentazione delle funzioni

6.13.1.1 `canc()`

```
void canc (
    Tree< Auto > & aTree,
    Tree< Moto > & mTree,
    Tree< Barca > & bTree )
```

Funzione che cancella tutte le auto, moto e barche o una di una categoria sotto una cilindrata data

6.13.1.2 `cancA()`

```
void cancA (
    int key,
    Tree< Auto > & aTree )
```

Funzione di cancellazione di un auto

6.13.1.3 `cancB()`

```
void cancB (
    int key,
    Tree< Barca > & bTree )
```

Funzione di cancellazione di una barca

6.13.1.4 `cancM()`

```
void cancM (
    int key,
    Tree< Moto > & mTree )
```

Funzione di cancellazione di una moto

6.13.1.5 `loadA()`

```
void loadA (
    ListAuto & aList,
    string fileNameA )
```

Funzione che si occupa di acquisire i dati all'interno della lista delle auto

6.13.1.6 `loadB()`

```
void loadB (
    ListBarca & bList,
    string fileNameB )
```

Funzione che si occupa di acquisire i dati all'interno della lista delle barche

6.13.1.7 `loadM()`

```
void loadM (
    ListMoto & mList,
    string fileNameM )
```

Funzione che si occupa di acquisire i dati all'interno della lista delle moto

6.13.1.8 main()

```
int main ( )
```

Creo il tread per l'inserimento delle auto nella lista di appartenenza

Creo il tread per l'inserimento delle moto nella lista di appartenenza

Creo il tread per l'inserimento delle barche nella lista di appartenenza

Creo il tread per l'inserimento delle auto nell' albero di appartenenza

Creo il tread per l'inserimento delle moto nell' albero di appartenenza

Creo il tread per l'inserimento delle barche nell' albero di appartenenza

Funzione che gestisce la possibilit  effettuare cancellazioni negli alberi

6.13.1.9 print()

```
void print (
    ListAuto & aList,
    ListMoto & mList,
    ListBarca & bList )
```

Funzione per la stampa delle liste

6.13.1.10 travasaA()

```
void travasaA (
    ListAuto & aList,
    Tree< Auto > & aTree )
```

Funzione che si occupa di travasare i dati all'interno del BST [Auto](#) dalla lista delle auto

6.13.1.11 travasaB()

```
void travasaB (
    ListBarca & bList,
    Tree< Barca > & bTree )
```

Funzione che si occupa di travasare i dati all'interno del BST Barche dalla lista delle barche

6.13.1.12 travasaM()

```
void travasaM (
    ListMoto & mList,
    Tree< Moto > & mTree )
```

Funzione che si occupa di travasare i dati all'interno del BST [Moto](#) dalla lista delle moto

Indice analitico

Auto, [9](#)

- Auto, [10](#)
- Cavalli, [12](#)
- getCavalli, [10](#)
- getMarca, [10](#)
- Marca, [12](#)
- operator<, [11](#)
- operator<<, [12](#)
- operator>, [11](#)
- operator==, [11](#)
- setCavalli, [11](#)
- setMarca, [11](#)
- toString, [11](#)

Barca, [12](#)

- Barca, [13](#)
- Cavalli, [15](#)
- getCavalli, [14](#)
- getMarca, [14](#)
- Marca, [15](#)
- operator<, [14](#)
- operator<<, [15](#)
- operator>, [14](#)
- operator==, [14](#)
- setCavalli, [14](#)
- setMarca, [14](#)
- toString, [15](#)

canc

- main.cpp, [47](#)
- Tree< T >, [32](#)

cancA

- main.cpp, [47](#)

cancB

- main.cpp, [48](#)

cancM

- main.cpp, [48](#)

Cavalli

- Auto, [12](#)
- Barca, [15](#)
- Moto, [27](#)
- Veicolo, [36](#)

Considerazioni.md, [37](#)

getCavalli

- Auto, [10](#)
- Barca, [14](#)
- Moto, [25](#)
- Veicolo, [34](#)

Getdata

Lista, [16](#)

getData

- ListAuto, [18](#)
- ListBarca, [20](#)
- ListMoto, [22](#)
- Nodo< T >, [30](#)

Gethead

- Lista, [16](#)

getHead

- ListAuto, [18](#)
- ListBarca, [20](#)
- ListMoto, [22](#)

getKey

- Nodo< T >, [28](#)

getLeft

- Nodo< T >, [28](#)

getMarca

- Auto, [10](#)
- Barca, [14](#)
- Moto, [25](#)
- Veicolo, [34](#)

getNext

- Nodo< T >, [30](#)

Getnil

- Lista, [16](#)

getNil

- ListAuto, [18](#)
- ListBarca, [20](#)
- ListMoto, [23](#)

getParent

- Nodo< T >, [28](#)

getPrev

- Nodo< T >, [31](#)

getRight

- Nodo< T >, [28](#)

Gettail

- Lista, [16](#)

getTail

- ListAuto, [18](#)
- ListBarca, [21](#)
- ListMoto, [23](#)

- include/Auto.h, [37](#)

- include/Barca.h, [38](#)

- include/Lista.h, [39](#)

- include/ListAuto.h, [39](#)

- include/ListBarca.h, [40](#)

- include/ListMoto.h, [41](#)

- include/Moto.h, [42](#)

- include/Node.h, [44](#)

- include/Nodo.h, 44
- include/Tree.h, 45
- include/Veicolo.h, 46
- inOrder
 - Tree< T >, 32
- ins
 - Tree< T >, 33
- insert
 - ListAuto, 19
 - ListBarca, 21
 - ListMoto, 23
- isEmpty
 - ListAuto, 19
 - ListBarca, 21
 - ListMoto, 23
- Lista, 16
 - Getdata, 16
 - Gethead, 16
 - Getnil, 16
 - Gettail, 16
 - Lista, 16
 - Setdata, 17
 - Sethead, 17
 - Setnil, 17
 - Settail, 17
- ListAuto, 17
 - getData, 18
 - getHead, 18
 - getNil, 18
 - getTail, 18
 - insert, 19
 - isEmpty, 19
 - ListAuto, 18
 - print, 19
 - setHead, 19
 - setNil, 19
 - setTail, 19
- ListBarca, 20
 - getData, 20
 - getHead, 20
 - getNil, 20
 - getTail, 21
 - insert, 21
 - isEmpty, 21
 - ListBarca, 20
 - print, 21
 - setHead, 21
 - setNil, 21
 - setTail, 21
- ListMoto, 22
 - getData, 22
 - getHead, 22
 - getNil, 23
 - getTail, 23
 - insert, 23
 - isEmpty, 23
 - ListMoto, 22
 - print, 23
 - setHead, 23
 - setNil, 23
 - setTail, 24
- loadA
 - main.cpp, 48
- loadB
 - main.cpp, 48
- loadM
 - main.cpp, 48
- main
 - main.cpp, 48
- main.cpp, 47
 - canc, 47
 - cancA, 47
 - cancB, 48
 - cancM, 48
 - loadA, 48
 - loadB, 48
 - loadM, 48
 - main, 48
 - print, 49
 - travasaA, 49
 - travasaB, 49
 - travasaM, 49
- Marca
 - Auto, 12
 - Barca, 15
 - Moto, 27
 - Veicolo, 36
- Moto, 24
 - Cavalli, 27
 - getCavalli, 25
 - getMarca, 25
 - Marca, 27
 - Moto, 25
 - operator<, 26
 - operator<<, 27
 - operator>, 26
 - operator==, 26
 - setCavalli, 26
 - setMarca, 26
 - toString, 26
- Node
 - Node< T >, 28
- Node< T >, 27
 - getKey, 28
 - getLeft, 28
 - getParent, 28
 - getRight, 28
 - Node, 28
 - operator<<, 29
 - setKey, 28
 - setLeft, 29
 - setParent, 29
 - setRight, 29
- Nodo
 - Nodo< T >, 30

- Nodo< T >, 30
 - getData, 30
 - getNext, 30
 - getPrev, 31
 - Nodo, 30
 - print, 31
 - setData, 31
 - setNext, 31
 - setPrev, 31
- operator<
 - Auto, 11
 - Barca, 14
 - Moto, 26
 - Veicolo, 35
- operator<<
 - Auto, 12
 - Barca, 15
 - Moto, 27
 - Node< T >, 29
 - Veicolo, 36
- operator>
 - Auto, 11
 - Barca, 14
 - Moto, 26
 - Veicolo, 35
- operator==
 - Auto, 11
 - Barca, 14
 - Moto, 26
 - Veicolo, 35
- postOrder
 - Tree< T >, 33
- preOrder
 - Tree< T >, 33
- print
 - ListAuto, 19
 - ListBarca, 21
 - ListMoto, 23
 - main.cpp, 49
 - Nodo< T >, 31
- setCavalli
 - Auto, 11
 - Barca, 14
 - Moto, 26
 - Veicolo, 35
- Setdata
 - Lista, 17
- setData
 - Nodo< T >, 31
- Sethead
 - Lista, 17
- setHead
 - ListAuto, 19
 - ListBarca, 21
 - ListMoto, 23
- setKey
 - Node< T >, 28
- setLeft
 - Node< T >, 29
- setMarca
 - Auto, 11
 - Barca, 14
 - Moto, 26
 - Veicolo, 35
- setNext
 - Nodo< T >, 31
- Setnil
 - Lista, 17
- setNil
 - ListAuto, 19
 - ListBarca, 21
 - ListMoto, 23
- setParent
 - Node< T >, 29
- setPrev
 - Nodo< T >, 31
- setRight
 - Node< T >, 29
- Settail
 - Lista, 17
- setTail
 - ListAuto, 19
 - ListBarca, 21
 - ListMoto, 24
- toString
 - Auto, 11
 - Barca, 15
 - Moto, 26
 - Veicolo, 35
- travasaA
 - main.cpp, 49
- travasaB
 - main.cpp, 49
- travasaM
 - main.cpp, 49
- Tree
 - Tree< T >, 32
- Tree< T >, 32
 - canc, 32
 - inOrder, 32
 - ins, 33
 - postOrder, 33
 - preOrder, 33
 - Tree, 32
- Veicolo, 33
 - Cavalli, 36
 - getCavalli, 34
 - getMarca, 34
 - Marca, 36
 - operator<, 35
 - operator<<, 36
 - operator>, 35
 - operator==, 35

setCavalli, [35](#)
setMarca, [35](#)
toString, [35](#)
Veicolo, [34](#)