

Evolution

Generated by Doxygen 1.8.14

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	myList< T >::Element Struct Reference	5
3.1.1	Constructor & Destructor Documentation	5
3.1.1.1	Element() [1/2]	5
3.1.1.2	Element() [2/2]	6
3.1.2	Member Data Documentation	6
3.1.2.1	value	6
3.1.2.2	wNext	6
3.2	Evolution Class Reference	6
3.2.1	Constructor & Destructor Documentation	8
3.2.1.1	Evolution()	8
3.2.2	Member Function Documentation	8
3.2.2.1	adjustmentFunction()	8
3.2.2.2	Evolve()	9
3.2.2.3	func()	9
3.2.2.4	generate_random_population_file()	9
3.2.2.5	RankDeathSelector()	9
3.2.2.6	RankReproductionSelector()	9

3.2.2.7	read_population_file()	10
3.2.2.8	RejectionDeathSelector()	10
3.2.2.9	RejectionReproductionSelector()	10
3.2.2.10	reproduce()	10
3.2.2.11	write_population_file()	10
3.2.3	Friends And Related Function Documentation	10
3.2.3.1	myList< int >	10
3.2.4	Member Data Documentation	11
3.2.4.1	chr_max	11
3.2.4.2	chr_min	11
3.2.4.3	chromosome_length	11
3.2.4.4	func_nr	11
3.2.4.5	iterations	11
3.2.4.6	max_gen_value	11
3.2.4.7	maxAdjustment	12
3.2.4.8	maxindex	12
3.2.4.9	min_gen_value	12
3.2.4.10	minAdjustment	12
3.2.4.11	pathIN	12
3.2.4.12	pathOUT	12
3.2.4.13	population	13
3.2.4.14	population_size	13
3.2.4.15	r	13
3.2.4.16	reproductionFactor	13
3.2.4.17	sumAdjustment	13
3.2.4.18	ToReproduction	13
3.2.4.19	w	14
3.3	myList< T > Class Template Reference	14
3.3.1	Constructor & Destructor Documentation	15
3.3.1.1	myList() [1/4]	15

3.3.1.2	myList() [2/4]	15
3.3.1.3	myList() [3/4]	15
3.3.1.4	myList() [4/4]	16
3.3.1.5	~myList()	16
3.3.2	Member Function Documentation	16
3.3.2.1	Add()	16
3.3.2.2	clean()	16
3.3.2.3	count()	17
3.3.2.4	CrossingOver()	17
3.3.2.5	get_r()	17
3.3.2.6	operator=()	17
3.3.2.7	operator[]()	18
3.3.2.8	Print()	18
3.3.2.9	Pull()	18
3.3.2.10	RemoveAt()	18
3.3.3	Friends And Related Function Documentation	18
3.3.3.1	Evolution	18
3.3.4	Member Data Documentation	19
3.3.4.1	adjustment	19
3.3.4.2	Count	19
3.3.4.3	wFirst	19
3.3.4.4	wLast	19
4	File Documentation	21
4.1	C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/Evolution.cpp File Reference	21
4.1.1	Function Documentation	21
4.1.1.1	get_random()	21
4.1.2	Variable Documentation	22
4.1.2.1	bl	22
4.1.2.2	bl2	22
4.1.2.3	bl2Range	22
4.1.2.4	blRange	22
4.2	C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/Evolution.h File Reference	22
4.3	C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/Main.cpp File Reference	23
4.3.1	Function Documentation	23
4.3.1.1	main()	23
4.4	C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/myList.h File Reference	23
4.5	C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/pch.cpp File Reference	23
4.6	C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/pch.h File Reference	23
Index		25

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

myList< T >::Element	5
Evolution	6
myList< T >	14

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/ Evolution.cpp	21
C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/ Evolution.h	22
C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/ Main.cpp	23
C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/ myList.h	23
C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/ pch.cpp	23
C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/ pch.h	23

Chapter 3

Class Documentation

3.1 myList< T >::Element Struct Reference

Public Member Functions

- [Element](#) ()
Pusty konstruktor.
- [Element](#) (T &Val)
Konstruktor z wartoscia elementu.

Public Attributes

- T [value](#)
Wartosc elementu.
- [Element](#) * [wNext](#) = nullptr
Wskaźnik na następny element.

3.1.1 Constructor & Destructor Documentation

3.1.1.1 [Element\(\)](#) [1/2]

```
template<typename T >  
myList< T >::Element::Element ( )
```

Pusty konstruktor.

Domyslny konstruktor.

=====

Konstruktory

Struct [Element](#)

3.1.1.2 Element() [2/2]

```
template<typename T>
myList< T >::Element::Element (
    T & Val )
```

Konstruktor z wartoscia elementu.

Z wartoscia elementu przekazana przez referencje.

3.1.2 Member Data Documentation

3.1.2.1 value

```
template<typename T>
T myList< T >::Element::value
```

Wartosc elementu.

3.1.2.2 wNext

```
template<typename T>
Element* myList< T >::Element::wNext = nullptr
```

Wskaznik na nastepny element.

The documentation for this struct was generated from the following file:

- C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/[myList.h](#)

3.2 Evolution Class Reference

```
#include <Evolution.h>
```

Public Member Functions

- [Evolution](#) (unsigned int Iterations=200, std::string path_in="First Population.txt", std::string path_out="Final Population.txt", double reproductionCount_factor=0.5, double death_factor=0.8, double reproduction_factor=0.8)
konstruktor z przekazanymi parametrami. lub domyslnymi.
- void [Evolve](#) ()
Start programu.
- void [generate_random_population_file](#) ()
Generuj plik txt z losowymi osobnikami wg powyzzszych parametrow.
- void [adjustmentFunction](#) ()
Funkcja dopasowania. ocenia osobnikow.
- double [func](#) (int x)
- bool [RejectionReproductionSelector](#) ()
- void [RankReproductionSelector](#) ()
losowanie osobnikow przez ranking.
- void [reproduce](#) ()
rozmnazanie.
- bool [RejectionDeathSelector](#) ()
- void [RankDeathSelector](#) ()
losowanie osobnikow przez ranking.
- void [read_population_file](#) ()
Wczytaj populacje z pliku txt do listy population.
- void [write_population_file](#) ()
Zapisz populacje do pliku wyjsciowego.

Public Attributes

- double [reproductionFactor](#)
- unsigned int [iterations](#)
- std::string [pathIN](#)
plik wejscowy
- std::string [pathOUT](#)
plik wyjsciowy
- double [w](#)
wspolczynnik wymierania (0;1)
- double [r](#)
wspolczynnik rozmnazania (0;1)
- [myList](#)< [myList](#)< int > > [population](#)
- [myList](#)< int > [ToReproduction](#)
Indeksy osobnikow do rozmnazania (z listy population).
- double [maxAdjustment](#)
najwyzsza wartosc funkcji dostosowania sposrod osobnikow.
- double [minAdjustment](#)
najnizsza wartosc funkcji dostosowania sposrod osobnikow.
- double [sumAdjustment](#)
suma
- unsigned int [maxindex](#) = 0
indeks elementu o max wartosci funkcji.

Static Public Attributes

- static const unsigned int `population_size` = 1600
- static const unsigned int `chromosome_length` = 3
- static const unsigned int `chr_min` = 2
- static const unsigned int `chr_max` = 10
- static const unsigned int `min_gen_value` = 0
Min wartosc genu.
- static const unsigned int `max_gen_value` = 10000
Max wartosc genu.
- static const unsigned int `func_nr` = 2
Wybor funkcji dopasowania.

Friends

- class `myList< int >`

3.2.1 Constructor & Destructor Documentation

3.2.1.1 Evolution()

```
Evolution::Evolution (
    unsigned int Iterations = 200,
    std::string path_in = "First Population.txt",
    std::string path_out = "Final Population.txt",
    double reproductionCount_factor = 0.5,
    double death_factor = 0.8,
    double reproduction_Factor = 0.8 )
```

konstruktor z przekazanymi parametrami. lub domyslnymi.

Konstruktor przyjmuje iteracje, sciezki pliku wejscowego i wyjscowego, % osobnikow ktorzy przystapia do rozmnazania, wspolczynnik wymierania i wsp rozmnazania z zakresu 0-1

3.2.2 Member Function Documentation

3.2.2.1 adjustmentFunction()

```
void Evolution::adjustmentFunction ( )
```

Funkcja dopasowania. ocenia osobnikow.

3.2.2.2 Evolve()

```
void Evolution::Evolve ( )
```

Start programu.

start programu.

Metody

3.2.2.3 func()

```
double Evolution::func (
    int x )
```

rownania funkcji dopasowania przyjmuje wartosc genu. zwraca wartosc funkcji dopasowania.

Funkcja dopasowania. Obliczana jest wartosc funkcji dla kazdego genu, nastepnie wyciagana jest srednia i zwracany jest jej szescian.

3.2.2.4 generate_random_population_file()

```
void Evolution::generate_random_population_file ( )
```

Generuj plik txt z losowymi osobnikami wg powyzzszych parametrow.

Generuje plik txt z losowa populacja.

class [Evolution](#)

3.2.2.5 RankDeathSelector()

```
void Evolution::RankDeathSelector ( )
```

losowanie osobnikow przez ranking.

3.2.2.6 RankReproductionSelector()

```
void Evolution::RankReproductionSelector ( )
```

losowanie osobnikow przez ranking.

3.2.2.7 read_population_file()

```
void Evolution::read_population_file ( )
```

Wczytaj populacje z pliku txt do listy population.

Wczytaj populacje z pliku.

3.2.2.8 RejectionDeathSelector()

```
bool Evolution::RejectionDeathSelector ( )
```

losowanie osobnikow do eliminacji. algorytm typu Monte Carlo method - Rejection sampling. zwraca false gdy nie udalo sie wyznaczyc osobnika.

3.2.2.9 RejectionReproductionSelector()

```
bool Evolution::RejectionReproductionSelector ( )
```

losowanie osobnikow ktorzy przystapia do rozmnazania. algorytm typu Monte Carlo method - Rejection sampling. zwraca false gdy nie udalo sie wyznaczyc osobnika.

3.2.2.10 reproduce()

```
void Evolution::reproduce ( )
```

rozmnazanie.

Rozmnazanie.

3.2.2.11 write_population_file()

```
void Evolution::write_population_file ( )
```

Zapisz populacje do pliku wyjsciowego.

3.2.3 Friends And Related Function Documentation

3.2.3.1 myList< int >

```
friend class myList< int > [friend]
```


3.2.4 Member Data Documentation

3.2.4.1 chr_max

```
const unsigned int Evolution::chr_max = 10 [static]
```

3.2.4.2 chr_min

```
const unsigned int Evolution::chr_min = 2 [static]
```

3.2.4.3 chromosome_length

```
const unsigned int Evolution::chromosome_length = 3 [static]
```

Długość chromosomu / osobnika. jeśli 0, długość wybierana losowo pomiędzy chr_min a chr_max

3.2.4.4 func_nr

```
const unsigned int Evolution::func_nr = 2 [static]
```

Wybór funkcji dopasowania.

3.2.4.5 iterations

```
unsigned int Evolution::iterations
```

Liczba pokoleń. Dla dobrze dobranych parametrów powinno wystarczyć 100 - 200 pokoleń.

3.2.4.6 max_gen_value

```
const unsigned int Evolution::max_gen_value = 10000 [static]
```

Max wartość genu.

3.2.4.7 maxAdjustment

```
double Evolution::maxAdjustment
```

najwyższa wartosc funkcji dostosowania sposrod osobnikow.

3.2.4.8 maxindex

```
unsigned int Evolution::maxindex = 0
```

indeks elementu o max wartosci funkcji.

3.2.4.9 min_gen_value

```
const unsigned int Evolution::min_gen_value = 0 [static]
```

Min wartosc genu.

3.2.4.10 minAdjustment

```
double Evolution::minAdjustment
```

najniższa wartosc funkcji dostosowania sposrod osobnikow.

3.2.4.11 pathIN

```
std::string Evolution::pathIN
```

plik wejscowy

3.2.4.12 pathOUT

```
std::string Evolution::pathOUT
```

plik wyjsciowy

3.2.4.13 population

```
myList<myList<int> > Evolution::population
```

Populacja.

3.2.4.14 population_size

```
const unsigned int Evolution::population_size = 1600 [static]
```

[://en.wikipedia.org/wiki/Rejection_sampling](https://en.wikipedia.org/wiki/Rejection_sampling)

Parametry

Liczebność populacji.

3.2.4.15 r

```
double Evolution::r
```

współczynnik rozmnażania (0;1)

3.2.4.16 reproductionFactor

```
double Evolution::reproductionFactor
```

1 -> max x = 50 2 -> max x = 475

ile % populacji przystępuje do rozmnażania.

3.2.4.17 sumAdjustment

```
double Evolution::sumAdjustment
```

suma

3.2.4.18 ToReproduction

```
myList<int> Evolution::ToReproduction
```

Indeksy osobników do rozmnażania (z listy population).

3.2.4.19 w

```
double Evolution::w
```

współczynnik wymierania (0;1)

The documentation for this class was generated from the following files:

- C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/[Evolution.h](#)
- C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/[Evolution.cpp](#)

3.3 myList< T > Class Template Reference

```
#include <myList.h>
```

Classes

- struct [Element](#)

Public Member Functions

- T & [operator\[\]](#) (unsigned int index)
przeciazony operator []
- [myList](#)< T > & [operator=](#) ([myList](#) &m1)
przeciazony operator =
- [myList](#) ()
Domyslny konstruktor.
- [myList](#) ([myList](#) &original)
Konstruktor kopiujacy.
- [myList](#) (T &value)
Konstruktor listy z pierwszym elementem.
- [myList](#) (T *arr, int n)
Konstruktor listy z tablicy.
- [~myList](#) ()
Destruktor listy.
- int [count](#) ()
Getter wartosci int Count.
- void [Add](#) (T &val)
Dodaj do listy przez referencje.
- T * [Pull](#) (int index)
Wyciagnij wskaznik na element o indexie.
- void [RemoveAt](#) (int index)
Usun element o indexie.
- void [clean](#) ()
Usun wszystkie elementy z listy.
- void [Print](#) ()
Wypisz w konsoli cala liste.
- void [CrossingOver](#) ([myList](#) &m1)
rekombinacja osobnikow (na potrzeby klasy [Evolution](#))
- int [get_r](#) (int min, int max)
losowy int z przedzialu min ; max.

Private Attributes

- `Element * wFirst = nullptr`
Wskaźnik pierwszego elementu listy.
- `Element * wLast = nullptr`
Wskaźnik ostatniego elementu listy.
- `unsigned int Count = 0`
Ilość elementów w liście.
- `double adjustment`
przechowuje wartość przystosowania 'osobnika'.

Friends

- class `Evolution`

3.3.1 Constructor & Destructor Documentation

3.3.1.1 myList() [1/4]

```
template<typename T >
myList< T >::myList ( )
```

Domyslny konstruktor.

class `myList`

3.3.1.2 myList() [2/4]

```
template<typename T >
myList< T >::myList (
    myList< T > & original )
```

Konstruktor kopiujący.

Kopiujący konstruktor.

3.3.1.3 myList() [3/4]

```
template<typename T>
myList< T >::myList (
    T & value )
```

Konstruktor listy z pierwszym elementem.

Z pierwszym elementem listy, przekazany przez referencję.

3.3.1.4 `myList()` [4/4]

```
template<typename T>
myList< T >::myList (
    T * arr,
    int n )
```

Konstruktor listy z tablicy.

Z tablicy przekazanej przez wskaznik.

3.3.1.5 `~myList()`

```
template<typename T >
myList< T >::~~myList ( )
```

Destruktor listy.

`myList`

=====

Destruktory i przeciazone operatory

3.3.2 Member Function Documentation

3.3.2.1 `Add()`

```
template<typename T>
void myList< T >::Add (
    T & val )
```

Dodaj do listy przez referencje.

Dodaj przez referencje.

3.3.2.2 `clean()`

```
template<typename T >
void myList< T >::clean ( )
```

Usun wszystkie elementy z listy.

Wyczysc liste.

3.3.2.3 count()

```
template<typename T >
int myList< T >::count ( )
```

Getter wartosci int Count.

Getter warosci int Count.

=====

Metody**myList****3.3.2.4 CrossingOver()**

```
template<typename T >
void myList< T >::CrossingOver (
    myList< T > & m1 )
```

rekombinacja osobnikow (na potrzeby klasy [Evolution](#))

Wymiana fragmentow 2 list. na potrzeby klasy [Evolution](#).

3.3.2.5 get_r()

```
template<typename T >
int myList< T >::get_r (
    int min,
    int max )
```

losowy int z przedzialu min ; max.

Losowy int z przedzialu min, max.

3.3.2.6 operator=()

```
template<typename T >
myList< T > & myList< T >::operator= (
    myList< T > & m1 )
```

przeciazony operator =

operator =

3.3.2.7 operator[]()

```
template<typename T >
T & myList< T >::operator[] (
    unsigned int index )
```

przeciazony operator []

operator []

3.3.2.8 Print()

```
template<typename T >
void myList< T >::Print ( )
```

Wypisz w konsoli cala liste.

Wypisz wszystkie wartosci listy w konsoli.

3.3.2.9 Pull()

```
template<typename T >
T * myList< T >::Pull (
    int index )
```

Wyciagnij wskaznik na element o indexie.

Zwraca wskaznik na wartosc o indexie.

3.3.2.10 RemoveAt()

```
template<typename T >
void myList< T >::RemoveAt (
    int index )
```

Usun element o indexie.

3.3.3 Friends And Related Function Documentation

3.3.3.1 Evolution

```
template<typename T>
friend class Evolution [friend]
```


3.3.4 Member Data Documentation

3.3.4.1 adjustment

```
template<typename T>
double myList< T >::adjustment [private]
```

przechowuje wartosc przystosowania 'osobnika'.

3.3.4.2 Count

```
template<typename T>
unsigned int myList< T >::Count = 0 [private]
```

Ilosc elementow w liscie.

3.3.4.3 wFirst

```
template<typename T>
Element* myList< T >::wFirst = nullptr [private]
```

Wskaznik pierwszego elementu listy.

3.3.4.4 wLast

```
template<typename T>
Element* myList< T >::wLast = nullptr [private]
```

Wskaznik ostatniego elementu listy.

The documentation for this class was generated from the following file:

- C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/[myList.h](#)

Chapter 4

File Documentation

4.1 C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/Evolution.cpp File Reference

```
#include "pch.h"
#include "Evolution.h"
#include <random>
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
```

Functions

- int `get_random` (int min, int max)
Losowy int z przedzialu min, max.

Variables

- int `bl` = 0
- int `blRange` = 5000
- int `bl2` = 0
- int `bl2Range` = 10000

4.1.1 Function Documentation

4.1.1.1 `get_random()`

```
int get_random (
    int min,
    int max )
```

Losowy int z przedzialu min, max.

=====

Metody

4.1.2 Variable Documentation

4.1.2.1 bl

```
int bl = 0
```

Losowanie osobnikow do reprodukcji. algorytm typu Monte Carlo method, Rejection sampling. zwraca false, jesli nie udalo sie wylosowac osobnika przez okreslona ilosc iteracji.

4.1.2.2 bl2

```
int bl2 = 0
```

Losowanie osobnikow do eliminacji. to samo co w RejectionReproductionSelector() tylko odwrotnie.

4.1.2.3 bl2Range

```
int bl2Range = 10000
```

4.1.2.4 blRange

```
int blRange = 5000
```

4.2 C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/Evolution.h File Reference

```
#include "pch.h"  
#include "myList.h"  
#include <string>
```

Classes

- class [Evolution](#)

4.3 C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/Main.cpp File Reference

```
#include "pch.h"
#include "Evolution.h"
#include <iostream>
#include <vld.h>
```

Functions

- int [main](#) (int argc, char **argv)

4.3.1 Function Documentation

4.3.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

4.4 C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/myList.h File Reference

```
#include "pch.h"
#include "Evolution.h"
#include <iostream>
#include <string>
#include <random>
```

Classes

- class [myList< T >](#)
- struct [myList< T >::Element](#)

4.5 C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/pch.cpp File Reference

```
#include "pch.h"
```

4.6 C:/Users/korne/Desktop/projekt nowa wersja backup/myList/myList/pch.h File Reference

Index

- ~myList
 - myList, [16](#)
- Add
 - myList, [16](#)
- adjustment
 - myList, [19](#)
- adjustmentFunction
 - Evolution, [8](#)
- bl
 - Evolution.cpp, [22](#)
- bl2
 - Evolution.cpp, [22](#)
- bl2Range
 - Evolution.cpp, [22](#)
- blRange
 - Evolution.cpp, [22](#)
- C:/Users/korne/Desktop/projekt nowa wersja backup/my↔
List/myList/Evolution.cpp, [21](#)
- C:/Users/korne/Desktop/projekt nowa wersja backup/my↔
List/myList/Evolution.h, [22](#)
- C:/Users/korne/Desktop/projekt nowa wersja backup/my↔
List/myList/Main.cpp, [23](#)
- C:/Users/korne/Desktop/projekt nowa wersja backup/my↔
List/myList/myList.h, [23](#)
- C:/Users/korne/Desktop/projekt nowa wersja backup/my↔
List/myList/pch.cpp, [23](#)
- C:/Users/korne/Desktop/projekt nowa wersja backup/my↔
List/myList/pch.h, [23](#)
- chr_max
 - Evolution, [11](#)
- chr_min
 - Evolution, [11](#)
- chromosome_length
 - Evolution, [11](#)
- clean
 - myList, [16](#)
- Count
 - myList, [19](#)
- count
 - myList, [16](#)
- CrossingOver
 - myList, [17](#)
- Element
 - myList::Element, [5, 6](#)
- Evolution, [6](#)
 - adjustmentFunction, [8](#)
- chr_max, [11](#)
- chr_min, [11](#)
- chromosome_length, [11](#)
- Evolution, [8](#)
- Evolve, [8](#)
- func, [9](#)
- func_nr, [11](#)
- generate_random_population_file, [9](#)
- iterations, [11](#)
- max_gen_value, [11](#)
- maxAdjustment, [11](#)
- maxindex, [12](#)
- min_gen_value, [12](#)
- minAdjustment, [12](#)
- myList, [18](#)
- myList< int >, [10](#)
- pathIN, [12](#)
- pathOUT, [12](#)
- population, [12](#)
- population_size, [13](#)
- r, [13](#)
- RankDeathSelector, [9](#)
- RankReproductionSelector, [9](#)
- read_population_file, [9](#)
- RejectionDeathSelector, [10](#)
- RejectionReproductionSelector, [10](#)
- reproduce, [10](#)
- reproductionFactor, [13](#)
- sumAdjustment, [13](#)
- ToReproduction, [13](#)
- w, [13](#)
- write_population_file, [10](#)
- Evolution.cpp
 - bl, [22](#)
 - bl2, [22](#)
 - bl2Range, [22](#)
 - blRange, [22](#)
 - get_random, [21](#)
- Evolve
 - Evolution, [8](#)
- func
 - Evolution, [9](#)
- func_nr
 - Evolution, [11](#)
- generate_random_population_file
 - Evolution, [9](#)
- get_r
 - myList, [17](#)

- get_random
 - Evolution.cpp, 21
- iterations
 - Evolution, 11
- main
 - Main.cpp, 23
- Main.cpp
 - main, 23
- max_gen_value
 - Evolution, 11
- maxAdjustment
 - Evolution, 11
- maxindex
 - Evolution, 12
- min_gen_value
 - Evolution, 12
- minAdjustment
 - Evolution, 12
- myList
 - ~myList, 16
 - Add, 16
 - adjustment, 19
 - clean, 16
 - Count, 19
 - count, 16
 - CrossingOver, 17
 - Evolution, 18
 - get_r, 17
 - myList, 15
 - operator=, 17
 - operator[], 17
 - Print, 18
 - Pull, 18
 - RemoveAt, 18
 - wFirst, 19
 - wLast, 19
- myList< int >
 - Evolution, 10
- myList< T >, 14
- myList< T >::Element, 5
- myList::Element
 - Element, 5, 6
 - value, 6
 - wNext, 6
- operator=
 - myList, 17
- operator[]
 - myList, 17
- pathIN
 - Evolution, 12
- pathOUT
 - Evolution, 12
- population
 - Evolution, 12
- population_size
 - Evolution, 13
- Print
 - myList, 18
- Pull
 - myList, 18
- r
 - Evolution, 13
- RankDeathSelector
 - Evolution, 9
- RankReproductionSelector
 - Evolution, 9
- read_population_file
 - Evolution, 9
- RejectionDeathSelector
 - Evolution, 10
- RejectionReproductionSelector
 - Evolution, 10
- RemoveAt
 - myList, 18
- reproduce
 - Evolution, 10
- reproductionFactor
 - Evolution, 13
- sumAdjustment
 - Evolution, 13
- ToReproduction
 - Evolution, 13
- value
 - myList::Element, 6
- w
 - Evolution, 13
- wFirst
 - myList, 19
- wLast
 - myList, 19
- wNext
 - myList::Element, 6
- write_population_file
 - Evolution, 10