

Geny

Generated by Doxygen 1.9.6

1 Todo List	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 miasto Struct Reference	7
4.1.1 Detailed Description	7
4.1.2 Member Data Documentation	7
4.1.2.1 numer	7
4.1.2.2 x	8
4.1.2.3 y	8
4.2 populacja Struct Reference	8
4.2.1 Detailed Description	8
4.2.2 Member Data Documentation	8
4.2.2.1 nr_pokolenia	8
4.2.2.2 trasy	9
4.3 trasa Struct Reference	9
4.3.1 Detailed Description	9
4.3.2 Member Data Documentation	9
4.3.2.1 dlugosc	9
4.3.2.2 miasta	9
5 File Documentation	11
5.1 Geny/Geny/funkcje.cpp File Reference	11
5.1.1 Function Documentation	12
5.1.1.1 dlugosc_trasy()	12
5.1.1.2 error()	12
5.1.1.3 generowanie_populacji()	13
5.1.1.4 generowanie_trasy()	13
5.1.1.5 generuj_punkty_prostokat()	13
5.1.1.6 krzyzowanie_pmx()	14
5.1.1.7 losowa()	14
5.1.1.8 losowa2()	14
5.1.1.9 mutacja()	15
5.1.1.10 odczyt_miast()	15
5.1.1.11 odleglosc_miast()	16
5.1.1.12 parametry_wejsciowe()	16
5.1.1.13 selekcja_turniejowa()	16
5.1.1.14 wypisz_miasta()	17
5.1.1.15 wypisz_populacje()	17

5.1.1.16 wypisz_trase()	18
5.1.1.17 zapisz_do_pliku()	18
5.2 Geny/Geny/funkcje.h File Reference	18
5.2.1 Function Documentation	19
5.2.1.1 dlugosc_trasy()	19
5.2.1.2 error()	20
5.2.1.3 generowanie_populacji()	20
5.2.1.4 generowanie_trasy()	20
5.2.1.5 generuj_punkty_prostokat()	21
5.2.1.6 krzyzowanie_pmx()	21
5.2.1.7 losowa()	22
5.2.1.8 losowa2()	22
5.2.1.9 mutacja()	22
5.2.1.10 odczyt_miast()	22
5.2.1.11 odleglosc_miast()	23
5.2.1.12 parametry_wejscowe()	23
5.2.1.13 selekcja_turniejowa()	24
5.2.1.14 wypisz_miasta()	24
5.2.1.15 wypisz_populacje()	24
5.2.1.16 wypisz_trase()	25
5.2.1.17 zapisz_do_pliku()	25
5.3 funkcje.h	25
5.4 Geny/Geny/main.cpp File Reference	26
5.4.1 Function Documentation	26
5.4.1.1 main()	26
5.5 Geny/Geny/struct.h File Reference	27
5.5.1 Enumeration Type Documentation	27
5.5.1.1 class_error	27
5.6 struct.h	27
Index	29

Chapter 1

Todo List

Member main (p. 26) (int ile_param, char *param[])

main -r wejscie.txt działa

make: *** [makefile:34: release] Naruszenie ochrony pamięci (zrzut pamięci) działa

Member selekcja_turniejowa (p. 24) (populacja &popula)

wiecej z jednego shuffle

Chapter 2

Class Index

2.1 Class List

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

miasto		
	Struktura miasta	7
populacja		
	Fstruktura populacji	8
trasa		
	Struktura trasy	9

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Geny/Geny/ funkcje.cpp	11
Geny/Geny/ funkcje.h	18
Geny/Geny/ main.cpp	26
Geny/Geny/ struct.h	27

Chapter 4

Class Documentation

4.1 miasto Struct Reference

Struktura miasta.

```
#include <struct.h>
```

Public Attributes

- int **numer**
- double **x**
- double **y**

4.1.1 Detailed Description

Struktura miasta.

Parameters

<i>numer</i>	numer miasta
<i>x</i>	wspolrzedne x
<i>y</i>	wspolrzedne y

4.1.2 Member Data Documentation

4.1.2.1 numer

```
int miasto::numer
```

4.1.2.2 x

```
double miasto::x
```

4.1.2.3 y

```
double miasto::y
```

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

- Geny/Geny/ **struct.h**

4.2 populacja Struct Reference

Fstruktura populacji.

```
#include <struct.h>
```

Public Attributes

- int **nr_pokolenia**
- std::vector< **trasa** > **trasy**

4.2.1 Detailed Description

Fstruktura populacji.

Parameters

<i>nr_pokolenia</i>	numer pokolenia
<i>trasy</i>	wektor tras / osobnikow

4.2.2 Member Data Documentation

4.2.2.1 nr_pokolenia

```
int populacja::nr_pokolenia
```

4.2.2.2 trasy

```
std::vector< trasa> populacja::trasy
```

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

- Geny/Geny/ **struct.h**

4.3 trasa Struct Reference

Struktura trasy.

```
#include <struct.h>
```

Public Attributes

- std::vector< **miasto** > **miasta**
- double **dlugosc**

4.3.1 Detailed Description

Struktura trasy.

Parameters

<i>miasta</i>	wektor miast bedacych trasa
<i>dlugosc</i>	dlugosc trasy

4.3.2 Member Data Documentation

4.3.2.1 dlugosc

```
double trasa::dlugosc
```

4.3.2.2 miasta

```
std::vector< miasto> trasa::miasta
```

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

- Geny/Geny/ **struct.h**

Chapter 5

File Documentation

5.1 Geny/Geny/funkcje.cpp File Reference

```
#include <iostream>
#include <string>
#include <fstream>
#include <sstream>
#include <algorithm>
#include <vector>
#include <cmath>
#include <random>
#include <unordered_map>
#include <chrono>
#include <numeric>
#include "funkcje.h"
#include "struct.h"
```

Functions

- **class_error parametry_wejscowe** (const int ile_param, char *param[], std::string &plik_wejscowy, std::string &plik_wyjsciowy, int &liczba_pokolen, int &liczba_osobnikow, int &generowanie_miast)
Funkcja czytuje i sprawdza poprawnosc paramentrow wejscowych.
- void **error** ()
Funkcja wyswietla blad zlego wprowadzenia paramentorw wejscowych.
- std::pair< **class_error**, std::vector< **miasto** > > **odczyt_miast** (const std::string &plik_wejscowy)
Funkcja czytuje miasta podane w pliku wejscowym.
- void **wypisz_miasta** (const std::vector< **miasto** > &miasta)
Funkcja wypisuje zczytane miasta.
- **trasa generowanie_trasy** (std::vector< **miasto** > miasta)
Funkcja generuje losowa trase.
- void **dlugosc_trasy** (**trasa** &tras)
Funkcja oblicza i aktualizuje dlugosc trasy.
- double **odleglosc_miast** (const **miasto** &miasto1, const **miasto** &miasto2)
Funkcja oblicza odleglosc od siebie podanych dwoch miast.
- void **wypisz_trase** (const **trasa** &droga1)
Funkcja wypisuje podana trase.

- **populacja generowanie_populacji** (const int &liczba_osobnikow, const std::vector< **miasto** > miasta)
Funkcja generuje losowa populacje.
- void **wypisz_populacje** (const **populacja** &populacja1)
Funkcja wypisuje podana populacje.
- int **losowa** (const int pocz, const int **koniec**)
- void **krzyzowanie_pmx** (**populacja** &popula, std::vector< **miasto** > miasta)
Funkcja krzyzujaca dwie losowe trasy w podanej populacji.
- void **mutacja** (**populacja** &popula)
Funkcja mutujaca losowa trase w populacji.
- std::pair< int, int > **losowa2** (const int &poczatek, const int & **koniec**)
Funkcja losujaca dwie unikalne liczby z podanego zakresu.
- **populacja selekcja_turniejowa** (**populacja** &popula)
Funkcja wykonujaca selekcje turniejowa na danej populacji.
- void **zapisz_do_pliku** (const **populacja** &popula, const std::string wyjsciowy)
Funkcja zapisuje najlepszy osobnik danej populacji do podanego pliku.
- void **generuj_punkty_prostokat** (const int LICZBA_PUNKTOW, const double X_MIN, const double X_MAX, const double Y_MIN, const double Y_MAX, std::string &wejscie)
Funkcja generujaca losowe miasta.

5.1.1 Function Documentation

5.1.1.1 dlugosc_trasy()

```
void dlugosc_trasy (
    trasa & tras )
```

Funkcja oblicza i aktualizuje dlugosc trasy.

Parameters

<i>tras</i>	trasa ktorej dlugosc bedzie obliczana
-------------	---------------------------------------

Returns

Funkcja void - nie zwraca

5.1.1.2 error()

```
void error ( )
```

Funkcja wyswietla blad zlego wprowadzenia paramentorw wejsciowych.

Returns

Funkcja void - nie zwraca

5.1.1.3 generowanie_populacji()

```
populacja generowanie_populacji (
    const int & liczba_osobnikow,
    const std::vector< miasto > miasta )
```

Funkcja generuje losowa populacje.

Parameters

<i>liczba_osobnikow</i>	ilosc osobnikow w generowanej populacji
<i>miasta</i>	miasta z ktorych beda generowane osobniki do trasy

Returns

zwraca generowana populacje

5.1.1.4 generowanie_trasy()

```
trasa generowanie_trasy (
    std::vector< miasto > miasta )
```

Funkcja generuje losowa trase.

Parameters

<i>miasta</i>	miasta z ktorych trasa bedzie generowana
---------------	--

Returns

funkcja zwraca wygenerowana trase

5.1.1.5 generuj_punkty_prostokat()

```
void generuj_punkty_prostokat (
    const int LICZBA_PUNKTOW,
    const double X_MIN,
    const double X_MAX,
    const double Y_MIN,
    const double Y_MAX,
    std::string & wejście )
```

Funkcja generujaca losowe miasta.

Parameters

<i>LICZBA_PUNKTOW</i>	ilosc punktow ktore maja sie wygenerowac
<i>X_MIN</i>	minimalna wartosc x
<i>X_MAX</i>	maksymalna wartosc x
<i>Y_MIN</i>	minimalna wartosc y
<i>Y_MAX</i>	maksymalna wartosc y
<i>wejscie</i>	plik wejscowy do ktorego sa generowane miasta

Returns

Funkcja void - nie zwraca

5.1.1.6 krzyzowanie_pmx()

```
void krzyzowanie_pmx (
    populacja & popula,
    std::vector< miasto > miasta )
```

Funkcja krzyzująca dwie losowe trasy w podanej populacji.

Parameters

<i>popula</i>	populacja poddawana krzyzowaniu
<i>miasta</i>	miasta czytana z pliku

Returns

Funkcja void - nie zwraca

5.1.1.7 losowa()

```
int losowa (
    const int pocz,
    const int koniec )
```

5.1.1.8 losowa2()

```
std::pair< int, int > losowa2 (
    const int & poczatek,
    const int & koniec )
```

Funkcja losująca dwie unikalne liczby z podanego zakresu.

Parameters

<i>poczatek</i>	poczatek losowanego zakresu
<i>koniec</i>	koniec losowanego zakresu

Returns

para unikalnych liczb int

5.1.1.9 mutacja()

```
void mutacja (
    populacja & poplua )
```

Funkcja mutująca losowa trasę w populacji.

Parameters

<i>popula</i>	populacja na ktorej bedzie wykonywana mutacja
---------------	---

Returns

Funkcja void - nie zwraca

5.1.1.10 odczyt_miast()

```
std::pair< class_error, std::vector< miasto > > odczyt_miast (
    const std::string & plik_wejscowy )
```

Funkcja zaczytuje miasta podane w pliku wejsciowym.

Parameters

<i>plik_wejscowy</i>	nazwa pliku z lista miast
----------------------	---------------------------

Returns

zwraca pare: czy nastapil blad i wektor odczytanych miast

5.1.1.11 odleglosc_miast()

```
double odleglosc_miast (
    const miasto & miasto1,
    const miasto & miasto2 )
```

Funkcja oblicza odleglosc od siebie podanych dwoch miast.

Parameters

<i>miasto1</i>	pierwsze miasto
<i>miasto</i>	2 drugie miasto

Returns

zwraca obliczona odleglosc

5.1.1.12 parametry_wejscowe()

```
class_error parametry_wejscowe (
    const int ile_param,
    char * param[],
    std::string & plik_wejscowy,
    std::string & plik_wyjscowy,
    int & liczba_pokolen,
    int & liczba_osobnikow,
    int & generowanie_miast )
```

Funkcja czytuje i sprawdza poprawnosc paramentrow wejscowych.

Parameters

<i>ile_param</i>	liczba parametrow wejscowych
<i>param</i> []	tablica parametrow wejscowych
<i>plik_wejscowy</i>	plik z danymi wejscowymi
<i>plik_wyjscowy</i>	plik z danymi wyjscowymi (najlepszymi osobnikami kazdej populacji)
<i>liczba_pokolen</i>	ilosc pokolen algorytmu genetycznego
<i>liczba_osobnikow</i>	ilosc osobnikow w kazdym pokoleniu

Returns

zwraca czy parametry wejscowe zostaly podane prawidlowo

5.1.1.13 selekcja_turniejowa()

```
populacja selekcja_turniejowa (
    populacja & popula )
```

Funkcja wykonujaca selekcje turniejowa na danej populacji.

Parameters

<i>popula</i>	populacja na ktorej bedzie wykonywana selekcja
---------------	--

Returns

zwraca populacje po selekcji

Todo wiecej z jednego shuffle

5.1.1.14 wypisz_miasta()

```
void wypisz_miasta (
    const std::vector< miasto > & miasta )
```

Funkcja wypisuje zczytane miasta.

Parameters

<i>miasta</i>	wektor miast ktore beda wyswietlane
---------------	-------------------------------------

Returns

Funkcja void - nie zwraca

5.1.1.15 wypisz_populacje()

```
void wypisz_populacje (
    const populacja & populacja1 )
```

Funkcja wypisuje podana populacje.

Parameters

<i>populacja1</i>	wypisywana populacja
-------------------	----------------------

Returns

Funkcja void - nie zwraca

5.1.1.16 wypisz_trase()

```
void wypisz_trase (
    const trasa & droga1 )
```

Funkcja wypisuje podana trase.

Parameters

<i>droga1</i>	wypisywana trasa
---------------	------------------

Returns

Funkcja void - nie zwraca

5.1.1.17 zapisz_do_pliku()

```
void zapisz_do_pliku (
    const populacja & popula,
    const std::string wyjsciowy )
```

Funkcja zapisuje najlepszy osobnik danej populacji do podanego pliku.

Parameters

<i>popula</i>	populacja z ktorej zapisujemy najlepszy osobnik
<i>wyjsciowy</i>	nazwa pliku do ktorego bedzie zapisany wynik

Returns

Funkcja void - nie zwraca

5.2 Geny/Geny/funkcje.h File Reference

```
#include <string>
#include <vector>
#include "struct.h"
```

Functions

- **class_error parametry_wejsciowe** (const int ile_param, char *param[], std::string &plik_wejsciowy, std::string &plik_wyjsciowy, int &liczba_pokolen, int &liczba_osobnikow, int &generowanie_miast)
Funkcja czytuje i sprawdza poprawnosc paramentrow wejsciowych.
- void **error** ()

- Funkcja wyswietla blad zlego wprowadzenia paramentorw wejsciowych.*
- `std::pair< class_error, std::vector< miasto > > odczyt_miast (const std::string &plik_wejsciowy)`
- Funkcja czytuje miasta podane w pliku wejsciowym.*
- `void wypisz_miasta (const std::vector< miasto > &miasta)`
- Funkcja wypisuje zczytane miasta.*
- `trasa generowanie_trasy (std::vector< miasto > miasta)`
- Funkcja generuje losowa trase.*
- `void wypisz_trase (const trasa &droga1)`
- Funkcja wypisuje podana trase.*
- `populacja generowanie_populacji (const int &liczba_osobnikow, const std::vector< miasto > miasta)`
- Funkcja generuje losowa populacje.*
- `void wypisz_populacje (const populacja &populacja1)`
- Funkcja wypisuje podana populacje.*
- `void dlugosc_trasy (trasa &tras)`
- Funkcja oblicza i aktualizuje dlugosc trasy.*
- `double odleglosc_miast (const miasto &miasto1, const miasto &miasto2)`
- Funkcja oblicza odleglosc od siebie podanych dwoch miast.*
- `int losowa (const int pocz, const int koniec)`
- `void krzyzowanie_pmx (populacja &popula, std::vector< miasto > miasta)`
- Funkcja krzyzujaca dwie losowe trasy w podanej populacji.*
- `void mutacja (populacja &poplua)`
- Funkcja mutujaca losowa trase w populacji.*
- `std::pair< int, int > losowa2 (const int &poczatek, const int &koniec)`
- Funkcja losujaca dwie unikalne liczby z podanego zakresu.*
- `populacja selekcja_turniejowa (populacja &popula)`
- Funkcja wykonujaca selekcje turniejowa na danej populacji.*
- `void zapisz_do_pliku (const populacja &popula, const std::string wyjsciowy)`
- Funkcja zapisuje najlepszy osobnik danej populacji do podanego pliku.*
- `void generuj_punkty_prostokat (const int LICZBA_PUNKTOW, const double X_MIN, const double X_MAX, const double Y_MIN, const double Y_MAX, std::string &wejscie)`
- Funkcja generujaca losowe miasta.*

5.2.1 Function Documentation

5.2.1.1 `dlugosc_trasy()`

```
void dlugosc_trasy (
    trasa & tras )
```

Funkcja oblicza i aktualizuje dlugosc trasy.

Parameters

<i>tras</i>	trasa ktorej dlugosc bedzie obliczana
-------------	---------------------------------------

Returns

Funkcja void - nie zwraca

5.2.1.2 error()

```
void error ( )
```

Funkcja wyswietla blad zlego wprowadzenia paramentorw wejsciowych.

Returns

Funkcja void - nie zwraca

5.2.1.3 generowanie_populacji()

```
populacja generowanie_populacji (
    const int & liczba_osobnikow,
    const std::vector< miasto > miasta )
```

Funkcja generuje losowa populacje.

Parameters

<i>liczba_osobnikow</i>	ilosc osobnikow w generowanej populacji
<i>miasta</i>	miasta z ktorych beda generowane osobniki do trasy

Returns

zwraca generowana populacje

5.2.1.4 generowanie_trasy()

```
trasa generowanie_trasy (
    std::vector< miasto > miasta )
```

Funkcja generuje losowa trase.

Parameters

<i>miasta</i>	miasta z ktorych trasa bedzie generowana
---------------	--

Returns

funkcja zwraca wygenerowana trase

5.2.1.5 generuj_punkty_prostokat()

```
void generuj_punkty_prostokat (
    const int LICZBA_PUNKTOW,
    const double X_MIN,
    const double X_MAX,
    const double Y_MIN,
    const double Y_MAX,
    std::string & wejście )
```

Funkcja generująca losowe miasta.

Parameters

<i>LICZBA_PUNKTOW</i>	ilosc punktow ktore maja sie wygenerowac
<i>X_MIN</i>	minimalna wartosc x
<i>X_MAX</i>	maksymalna wartosc x
<i>Y_MIN</i>	minimalna wartosc y
<i>Y_MAX</i>	maksymalna wartosc y
<i>wejście</i>	plik wejsciowy do ktorego sa generowane miasta

Returns

Funkcja void - nie zwraca

5.2.1.6 krzyzowanie_pmx()

```
void krzyzowanie_pmx (
    populacja & popula,
    std::vector< miasto > miasta )
```

Funkcja krzyżująca dwie losowe trasy w podanej populacji.

Parameters

<i>popula</i>	populacja poddawana krzyżowaniu
<i>miasta</i>	miasta zczytana z pliku

Returns

Funkcja void - nie zwraca

5.2.1.7 losowa()

```
int losowa (
    const int pocz,
    const int koniec )
```

5.2.1.8 losowa2()

```
std::pair< int, int > losowa2 (
    const int & poczatek,
    const int & koniec )
```

Funkcja losująca dwie unikalne liczby z podanego zakresu.

Parameters

<i>poczatek</i>	poczatek losowanego zakresu
<i>koniec</i>	koniec losowanego zakresu

Returns

para unikalnych liczb int

5.2.1.9 mutacja()

```
void mutacja (
    populacja & poplua )
```

Funkcja mutująca losową trasę w populacji.

Parameters

<i>popula</i>	populacja na której będzie wykonywana mutacja
---------------	---

Returns

Funkcja void - nie zwraca

5.2.1.10 odczyt_miast()

```
std::pair< class_error, std::vector< miasto > > odczyt_miast (
    const std::string & plik_wejsciowy )
```

Funkcja czytuje miasta podane w pliku wejściowym.

Parameters

<i>plik_wejscowy</i>	nazwa pliku z lista miast
----------------------	---------------------------

Returns

zwraca pare: czy nastapil blad i wektor odczytanych miast

5.2.1.11 `odleglosc_miast()`

```
double odleglosc_miast (
    const miasto & miasto1,
    const miasto & miasto2 )
```

Funkcja oblicza odleglosc od siebie podanych dwoch miast.

Parameters

<i>miasto1</i>	pierwsze miasto
<i>miasto</i>	2 drugie miasto

Returns

zwraca obliczona odleglosc

5.2.1.12 `parametry_wejscowe()`

```
class_error parametry_wejscowe (
    const int ile_param,
    char * param[],
    std::string & plik_wejscowy,
    std::string & plik_wyjscowy,
    int & liczba_pokolen,
    int & liczba_osobnikow,
    int & generowanie_miast )
```

Funkcja zaczytuje i sprawdza poprawnosc paramentrow wejsciowych.

Parameters

<i>ile_param</i>	liczba parametrow wejsciowych
<i>param[]</i>	tablica parametrow wejsciowych
<i>plik_wejscowy</i>	plik z danymi wejsciowymi
<i>plik_wyjscowy</i>	plik z danymi wyjsciowymi (najlepszymi osobnikami kazdej populacji)
<i>liczba_pokolen</i>	ilosc pokolen algorytmu genetycznego
<i>liczba_osobnikow</i>	ilosc osobnikow w kazdym pokoleniu

Returns

zwraca czy parametry wejsciowe zostaly podane prawidlowo

5.2.1.13 selekcja_turniejowa()

```
populacja selekcja_turniejowa (
    populacja & popula )
```

Funkcja wykonujaca selekcje turniejowa na danej populacji.

Parameters

<i>popula</i>	populacja na ktorej bedzie wykonywana selekcja
---------------	--

Returns

zwraca populacje po selekcji

Todo wiecej z jednego shuffle

5.2.1.14 wypisz_miasta()

```
void wypisz_miasta (
    const std::vector< miasto > & miasta )
```

Funkcja wypisuje zczytane miasta.

Parameters

<i>miasta</i>	wektor miast ktore beda wyswietlane
---------------	-------------------------------------

Returns

Funkcja void - nie zwraca

5.2.1.15 wypisz_populacje()

```
void wypisz_populacje (
    const populacja & populacja1 )
```

Funkcja wypisuje podana populacje.

Parameters

<i>populacja1</i>	wypisywana populacja
-------------------	----------------------

Returns

Funkcja void - nie zwraca

5.2.1.16 wypisz_trase()

```
void wypisz_trase (
    const trasa & droga1 )
```

Funkcja wypisuje podana trase.

Parameters

<i>droga1</i>	wypisywana trasa
---------------	------------------

Returns

Funkcja void - nie zwraca

5.2.1.17 zapisz_do_pliku()

```
void zapisz_do_pliku (
    const populacja & popula,
    const std::string wyjsciowy )
```

Funkcja zapisuje najlepszy osobnik danej populacji do podanego pliku.

Parameters

<i>popula</i>	populacja z ktorej zapisujemy najlepszy osobnik
<i>wyjsciowy</i>	nazwa pliku do ktorego bedzie zapisany wynik

Returns

Funkcja void - nie zwraca

5.3 funkcje.h

[Go to the documentation of this file.](#)

```

00001
00002 #include <string>
00003 #include <vector>
00004
00005 #include "struct.h"
00006
00017 class_error parametry_wejsciove(const int ile_param, char * param[], std::string& plik_wejsciovy,
std::string& plik_wyjsciovy, int& liczba_pokolen, int& liczba_osobnikow, int& generowanie_miast);
00018
00023 void error();
00024
00030 std::pair<class_error, std::vector<miasto> odczyt_miast(const std::string& plik_wejsciovy);
00031
00037 void wypisz_miasta (const std::vector<miasto>& miasta);
00038
00044 trasa generowanie_trasy(std::vector<miasto> miasta);
00045
00051 void wypisz_trase(const trasa& drogal);
00052
00059 populacja generowanie_populacji(const int& liczba_osobnikow, const std::vector<miasto> miasta);
00060
00066 void wypisz_populacje(const populacja& populacjal);
00067
00073 void dlugosc_trasy(trasa& tras);
00074
00081 double odleglosc_miast(const miasto& miasto1, const miasto& miasto2);
00082
00083 int losowa(const int pocz, const int koniec);
00084
00091 void krzyzowanie_pmx(populacja& popula, std::vector<miasto> miasta);
00092
00098 void mutacja(populacja& poplua);
00099
00106 std::pair<int, int> losowa2(const int& poczatek, const int& koniec);
00107
00113 populacja selekcja_turniejowa(populacja& popula);
00114
00121 void zapisz_do_pliku(const populacja& popula, const std::string wyjsciowy);
00122
00133 void generuj_punkty_prostokat(const int LICZBA_PUNKTOW, const double X_MIN, const double X_MAX, const
double Y_MIN, const double Y_MAX, std::string& wejscie);
00134

```

5.4 Geny/Geny/main.cpp File Reference

```

#include <iostream>
#include <string>
#include <vector>
#include "funkcje.h"
#include "struct.h"

```

Functions

- int **main** (int ile_param, char *param[])

5.4.1 Function Documentation

5.4.1.1 main()

```

int main (
    int ile_param,
    char * param[] )

```

Todo main -r wejscie.txt działa

make: *** [makefile:34: release] Naruszenie ochrony pamięci (zrzut pamięci) działa

5.5 Geny/Geny/struct.h File Reference

Classes

- struct **miasto**
Struktura miasta.
- struct **trasa**
Struktura trasy.
- struct **populacja**
Fstruktura populacji.

Enumerations

- enum class **class_error** { **error** , **ok** , **koniec** }
enum class do zwracania bladów

5.5.1 Enumeration Type Documentation

5.5.1.1 class_error

```
enum class class_error [strong]
```

enum class do zwracania bladów

Parameters

<i>error</i>	zwracany kiedy chce sie wyswietlic error i zakonczyc program
<i>ok</i>	zwracany kiedy program ma dzialac dalej
<i>koniec</i>	zwracany kiedy program ma sie zakonczyc

Enumerator

error	
ok	
koniec	

5.6 struct.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003
00010 struct miasto {
00011     int numer;
00012     double x, y;
```

```
00013 };
00014
00020 struct trasa {
00021     std::vector <miasto> miasta; // std::vector<miasto> miasta;
00022     double dlugosc;
00023 };
00029 struct populacja {
00030     int nr_pokolenia;
00031     std::vector<trasa> trasy; // std::vector<trasa> trasy;
00032 };
00033
00040 enum class class_error
00041 {
00042     error, ok, koniec
00043 };
```


Index

class_error
 struct.h, 27

dlugosc
 trasa, 9

dlugosc_trasy
 funkcje.cpp, 12
 funkcje.h, 19

error
 funkcje.cpp, 12
 funkcje.h, 20
 struct.h, 27

funkcje.cpp
 dlugosc_trasy, 12
 error, 12
 generowanie_populacji, 12
 generowanie_trasy, 13
 generuj_punkty_prostokat, 13
 krzyzowanie_pmx, 14
 losowa, 14
 losowa2, 14
 mutacja, 15
 odczyt_miast, 15
 odleglosc_miast, 15
 parametry_wejscowe, 16
 selekcja_turniejowa, 16
 wypisz_miasta, 17
 wypisz_populacje, 17
 wypisz_trase, 17
 zapisz_do_pliku, 18

funkcje.h
 dlugosc_trasy, 19
 error, 20
 generowanie_populacji, 20
 generowanie_trasy, 20
 generuj_punkty_prostokat, 21
 krzyzowanie_pmx, 21
 losowa, 21
 losowa2, 22
 mutacja, 22
 odczyt_miast, 22
 odleglosc_miast, 23
 parametry_wejscowe, 23
 selekcja_turniejowa, 24
 wypisz_miasta, 24
 wypisz_populacje, 24
 wypisz_trase, 25
 zapisz_do_pliku, 25

generowanie_populacji
 funkcje.cpp, 12
 funkcje.h, 20
generowanie_trasy
 funkcje.cpp, 13
 funkcje.h, 20
generuj_punkty_prostokat
 funkcje.cpp, 13
 funkcje.h, 21
Geny/Geny/funkcje.cpp, 11
Geny/Geny/funkcje.h, 18, 25
Geny/Geny/main.cpp, 26
Geny/Geny/struct.h, 27

koniec
 struct.h, 27
krzyzowanie_pmx
 funkcje.cpp, 14
 funkcje.h, 21

losowa
 funkcje.cpp, 14
 funkcje.h, 21
losowa2
 funkcje.cpp, 14
 funkcje.h, 22

main
 main.cpp, 26

main.cpp
 main, 26

miasta
 trasa, 9

miasto, 7
 numer, 7
 x, 7
 y, 8

mutacja
 funkcje.cpp, 15
 funkcje.h, 22

nr_pokolenia
 populacja, 8

numer
 miasto, 7

odczyt_miast
 funkcje.cpp, 15
 funkcje.h, 22

odleglosc_miast
 funkcje.cpp, 15

- funkcje.h, 23
- ok
 - struct.h, 27
- parametry_wejscowe
 - funkcje.cpp, 16
 - funkcje.h, 23
- populacja, 8
 - nr_pokolenia, 8
 - trasy, 8
- selekcja_turniejowa
 - funkcje.cpp, 16
 - funkcje.h, 24
- struct.h
 - class_error, 27
 - error, 27
 - koniec, 27
 - ok, 27
- trasa, 9
 - dlugosc, 9
 - miasta, 9
- trasy
 - populacja, 8
- wypisz_miasta
 - funkcje.cpp, 17
 - funkcje.h, 24
- wypisz_populacje
 - funkcje.cpp, 17
 - funkcje.h, 24
- wypisz_trase
 - funkcje.cpp, 17
 - funkcje.h, 25
- x
 - miasto, 7
- y
 - miasto, 8
- zapisz_do_pliku
 - funkcje.cpp, 18
 - funkcje.h, 25