Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki

Podstawy Programowania Komputerów

Darwin

autor	Piotr Magiera
prowadzący	dr Ewa Lach
rok akademicki	2019/2020
kierunek	Informatyka
rodzaj studiów	SSI
semestr	1
termin laboratorium	pon., 12:00 – 13:30
sekcja	6
termin oddania sprawozdania	2019-01-27

1 Treść zadania

Napisać program symulujący ewolucję populacji osobników. Populacja może liczyć dowolną liczbę osobników. Każdy osobnik zawiera chromosom, który jest ciągiem liczb naturalnych. Chromosomy mogą być różnej długości. W każdym pokoleniu wylosowywanych jest k par osobników, które się następnie krzyżują. Krzyżowanie polega na tym, że u każdego osobnika dochodzi do pęknięcia chromosomu w dowolnym miejscu. Część początkowa chromosomu jednego osobnika łączy się z częścią końcową drugiego. Inaczej mówiąc: osobniki wymieniają się fragmentami swoich chromosomów. Jeden osobnik może być wylosowany do kilku krzyżowań. Po dokonaniu wszystkich krzyżowań w pokoleniu sprawdzane jest przystosowanie osobników do warunków środowiska. W tym celu dla każdego osobnika wyznaczana jest wartość P [0, 1] funkcji dopasowania. Osobniki, dla których wartość P < w (gdzie w jest progiem wymierania), są usuwane z populacji. Osobniki, dla których P > r (gdzie r jest progiem rozmnażania) są klonowane. A osobniki, dla których w =< P =< r pozostają w populacji, ale się nie rozmnażają.

2 Analiza zadania

Zagadnienie przedstawia problem modyfikacji listy jednokierunkowej, w której składową każdego elementu listy jest kolejna lista jednokierunkowa.

2.1 Struktury danych

W programie wykorzystano listę jednokierunkową podwieszaną do przechowywania osobników i ich chromosomów. Każdy osobnik zawiera wskaźnik na następnego osobnika oraz wskaźnik na początek swojego chromosomu. Każdy element listy, która reprezentuje chromosom, zawiera jednocyfrową liczbę całkowitą reprezentującą kawałek chromosomu oraz wskaźnik na następny kawałek element tej listy. Lista podwieszana została wybrana z uwagi na to, że cały chromosom mógłby nie zmieścić się w zmiennej typu long long int.

2.2 Algorytmy

Program krzyżuje osobników zgodnie z treścią zadania losując ich oraz miejsca przerwania ich chromosomów za pomocą funkcji rand(). Program następnie wyznacza przystosowanie osobników do warunków środowiska za pomocą funkcji wyznaczającej przystosowanie P, której wzór znajduje się na poniższym rysunku. Na koniec program usuwa, pozostawia lub klonuje osobników zgodnie z warunkami zadania, tworząc nowe pokolenie. Operacje te są powtarzane tyle razy, ile jest to potrzebne do otrzymania szukanego pokolenia.

$$P = \sum_{i=1}^{d} \frac{k_i \; mod \; 9}{8d}$$

d – długość chromosomu danego osobnika

k_i – wartość i-tego kawałka chromosomu danego osobnika

3 Specyfikacja zewnętrzna

Program uruchamiany jest z linii poleceń z wykorzystaniem następujących przełączników (kolejność przełączników jest dowolna):

-i plik wejściowy z populacją

-o plik wyjściowy z populacją

-w współczynnik wymierania w [0, 1]

-r współczynnik rozmnażania r [0, 1]

-p liczba pokoleń p

-k liczba k par osobników losowanych do krzyżowania

Plik wejściowy ma następującą postać: Każda linia zawiera jednego osobnika. Osobnik charakteryzowany jest chromosomem, który jest przedstawiony jako ciąg liczb naturalnych rozdzielonych białymi znakami. Przykładowy plik wejściowy zawierający populację złożoną z czterech osobników:

2 9 84 9 5 6 25 12

2 98 56 2 54

5 2

8 5 22 5 48 6 1 9 8 7 554 25 235 32

Plik wyjściowy ma identyczny format.

W przypadku podania nieprawidłowej liczby argumentów, wyświetli się komunikat

Nieprawidlowa liczba argumentow

Jeśli współczynnik wymierania lub rozmnażania nie zawierają się w przedziale [0, 1] (mogą też nie być podane) lub współczynnik rozmnażania jest mniejszy bądź równy współczynnikowi wymierania wyświetli się komunikat

Bledny wspolczynnik wymierania/rozmnazania

Jeśli liczba pokoleń jest mniejsza lub równa, 0 lub nie jest podana, wyświetli się komunikat

Bledna liczba pokoleń

Jeśli liczba par jest mniejsza lub równa 0, lub nie jest podana, wyświetli się komunikat

Bledna liczba par

Jeśli podana jest nieprawidłowa ścieżka do pliku wejściowego (może też nie być podana) lub system operacyjny uniemożliwia dostęp do tego pliku, wyświetli się komunikat

Brak dostepu do pliku wejsciowego/plik wejsciowy nie istnieje

Jeśli w pliku wejściowym niektóre linie zawierają pojedyncze znaki, wyświetli się komunikat

Nieprawidlowy format pliku (pojedyncze chromosomy)

Jeśli w pliku wejściowym niektóre linie nie składają się z cyfr i białych znaków, wyświetli się komunikat

Nieprawidlowy format pliku (chromosomy nie skladaja sie z cyfr)

Jeśli w pliku wejściowym niektóre linie są puste lub plik jest pusty, wyświetli się komunikat

Nieprawidlowy format pliku (enter lub pusty plik)

4 Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem strukturalnym. W programie rozdzielono interfejs aplikacji (komunikację z użytkownikiem) od logiki aplikacji (modyfikacje listy podwieszanej).

4.1 Ogólna struktura programu

W funkcji głównej na początku wywoływana jest funkcja *srand(time(NULL))*. Następnie za pomocą funkcji *pobierzDane* pobierane są dane podane przez użytkownika w konsoli. Otwierany jest plik wejściowy i sprawdzana jest poprawność argumentów za pomocą utworzenia zmiennej *argumenty* i przypisania do niej wartości funkcji *sprawdzArgumenty*. Jeśli zmienna *argumenty* ma wartość false, program kończy swoje działanie. Następnie do listy podwieszanej pobierane są dane z pliku wejściowego za pomocą funkcji *pobierzZPliku*, której wartość przechowywana jest w utworzonej zmiennej *poprawnosc*. Jeśli zmienna *poprawnosc* ma wartość false, program kończy swoje działanie. Następnie zamykany jest plik wejściowy oraz wywoływana jest funkcja *zrealizujRozwojPopulacji*, która modyfikuje listę tak, aby otrzymać w niej pokolenie, którego postać jest poszukiwana przez użytkownika. Plik wyjściowy jest otwierany, po czym następuje wypisanie listy podwieszanej do pliku wyjściowego za pomocą funkcji *wypiszOsobnikow*. Plik wyjściowy jest zamykany, a następnie utworzona przez nas lista jest usuwana za pomocą funkcji *usunListe*. Kolejnym krokiem jest zakończenie działania programu.

4.2 Szczegółowy opis typów i funkcji

Szczegółowy opis typów i funkcji zawarty jest w załączniku.

5 Testowanie

Program został przetestowany na różnego rodzaju danych. Pliki wejściowe, których format jest nieprawidłowy, powodują wyświetlenie odpowiedniego komunikatu o błędzie. Niepoprawna liczba argumentów powoduje wyświetlenie odpowiedniego komunikatu, tak samo niepoprawne dane. Podanie ścieżki do nieistniejącego pliku wyjściowego powoduje utworzenie pliku wyjściowego o podanej nazwie w podanej lokalizacji. Program działa poprawnie zarówno dla pliku wejściowego zawierającego kilka linii zapełnionych cyframi i znakami białymi jak i wtedy, gdy plik ten zawiera tylko jedną taką linię. Jeśli w populacji wymrą wszyscy osobnicy, wtedy plik wyjściowy zawiera tekst *Lista jest pusta*. Program został sprawdzony pod kątem wycieków pamięci.

6 Wnioski

Program ten jest programem prostym, ale zarazem nietrywialnym. Najbardziej wymagającą częścią zadania była interpretacja polecenia (można zrozumieć, że przy krzyżowaniu osobników tworzony jest nowy osobnik jeśli uzna się, że autor zadania założył znajomość mechanizmów biologicznych u czytającego). Również kłopotliwe okazało się klonowanie osobnika – trzeba stworzyć nowego osobnika i stworzyć mu taki sam chromosom, jak chromosom kopiowanego osobnika, a następnie dodać tego osobnika do listy – przy próbie dodania już istniejącego osobnika lista osobników zapętlała się.

Dodatek
Szczegółowy opis typów i funkcji

Projekt Darwin PPK

Generated by Doxygen 1.8.17

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

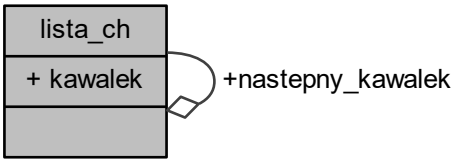
Chapter 3

Class Documentation

3.1 lista_ch Struct Reference

`#include <struktury.h>`

Collaboration diagram for lista_ch:



Public Attributes

- int kawalek

  liczba reprezentujaca kawalek chromosomu
- lista_ch * nastepny_kawalek

  wskaznik na nastepny kawalek chromosomu

3.1.1 Detailed Description

Element listy jednokierunkowej reprezentujacej chromosom osobnika.

3.1.2 Member Data Documentation

3.1.2.1 kawalek

```
int lista_ch::kawalek
```

liczba reprezentujaca kawalek chromosomu

3.1.2.2 nastepny_kawalek

```
lista_ch* lista_ch::nastepny_kawalek
```

wskaznik na nastepny kawalek chromosomu

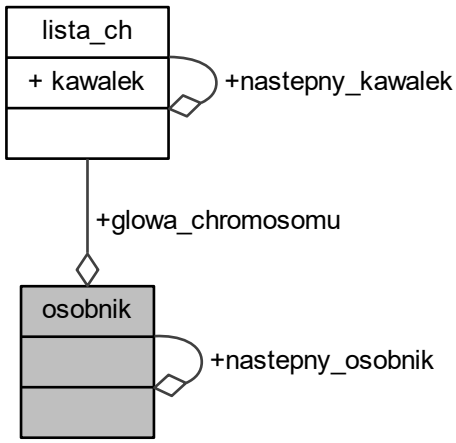
The documentation for this struct was generated from the following file:

- struktury.h

3.2 osobnik Struct Reference

```
#include <struktury.h>
```

Collaboration diagram for osobnik:



Public Attributes

- lista_ch * glowa_chromosomu
  wskaznik na poczatek chromosomu osobnika
- osobnik * nastepny_osobnik
  wskaznik na nastepnego osobnika

3.2.1 Detailed Description

Element listy jednokierunkowej osobnikow.

3.2.2 Member Data Documentation

3.2.2.1 glowa_chromosomu

```
lista_ch* osobnik::glowa_chromosomu
```

wskaznik na poczatek chromosomu osobnika

3.2.2.2 nastepny_osobnik

```
osobnik* osobnik::nastepny_osobnik
```

wskaznik na nastepnego osobnika

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

- struktury.h

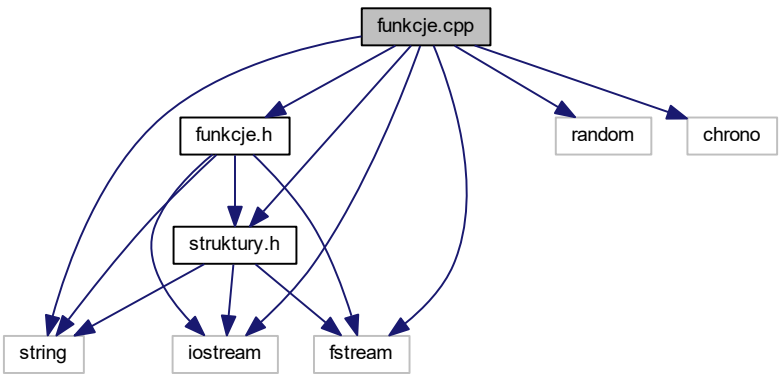
Chapter 4

File Documentation

4.1 funkcje.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include <random>
#include <chrono>
#include "funkcje.h"
#include "struktury.h"
```

Include dependency graph for funkcje.cpp:



Functions

- void pobierzDane (int argc, char ∗argv[ ], string &plik_wejsciowy, string &plik_wyjsciowy, double &wspolczynnik_wymierania, double &wspolczynnik_rozmnazania, int &liczba_pokolen, int &liczba_par)
- bool sprawdzArgumenty (int argc, ifstream &wejscie, double wspolczynnik_wymierania, double wspolczynnik_rozmnazania, int liczba_pokolen, int liczba_par)
- void dodajNaKoniecListyOsobnikow (osobnik ∗&pierwszy, osobnik ∗jakis)
- void dodajNaKoniecChromosomu (osobnik ∗&wskaznik_na_osobnika, int kawalek)

- bool pobierzZPliku (osobnik ∗&pierwszy, ifstream &wejscie)
- void wypiszChromosom (lista_ch ∗poczatek_chromosomu, ostream &strumien)
- void wypiszOsobnikow (osobnik ∗pierwszy, ostream &strumien)
- int liczbaOsobnikow (osobnik ∗pierwszy)
- int dlugoscChromosomu (lista_ch ∗poczatek_chromosomu)
- int dlugoscListy (osobnik ∗pierwszy)
- osobnik ∗ losujOsobnika (osobnik ∗pierwszy)
- lista_ch ∗ losujPrzerwanie (osobnik ∗jakis)
- void krzyzujOsobnikow (osobnik ∗&osobnik1, osobnik ∗&osobnik2)
- void zrealizujRozwojPopulacji (osobnik ∗&pierwszy, double wspolczynnik_rozmnazania, double wspolczynnik↩ _wymierania, int liczba_pokolen, int liczba_par)
- double wyznaczPrzystosowanie (osobnik ∗jakis)
- osobnik ∗ znajdzPoprzedniOsobnik (osobnik ∗pierwszy, osobnik ∗jakis)
- lista_ch ∗ znajdzPoprzedniKawalekChromosomu (lista_ch ∗poczatek, lista_ch ∗jakis)
- void usunKawalekChromosomu (lista_ch ∗&poczatek, lista_ch ∗&jakis, lista_ch ∗&poprzedni)
- void usunCalyChromosom (lista_ch ∗&poczatek_chromosomu)
- void usunOsobnika (osobnik ∗&pierwszy, osobnik ∗&jakis, osobnik ∗&poprzedni)
- void usunListe (osobnik ∗&pierwszy)

4.1.1 Function Documentation

4.1.1.1 dlugoscChromosomu()

```
int dlugoscChromosomu (
            lista_ch * poczatek_chromosomu )
```

Funkcja oblicza dlugosc chromosomu.

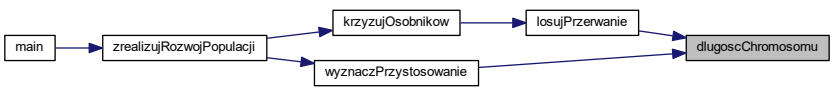
Parameters

poczatek_chromosomu	wskaznik na pierwszy kawalek chromosomu

Returns

dlugosc chromosomu

Here is the caller graph for this function:



4.1.1.2 dlugoscListy()

```
int dlugoscListy (
            osobnik * pierwszy )
```

Funkcja oblicza dlugosc listy osobnikow.

Parameters

pierwszy	wskaznik na pierwszego osobnika w liscie

Returns

dlugosc listy osobnikow

Here is the caller graph for this function:

4.1.1.3 dodajNaKoniecChromosomu()

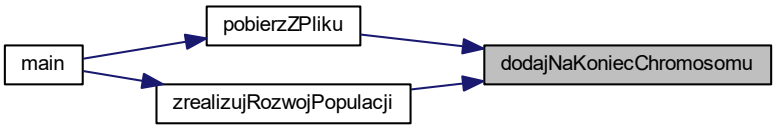
```
void dodajNaKoniecChromosomu (
            osobnik *& wskaznik_na_osobnika,
            int kawalek )
```

Funkcja dodaje liczbe na koniec chromosomu.

Parameters

in,out	*wskaznik_na_osobnika*	wskaznik na osobnika, na ktorego koniec chromosomu chcemy dodac dana liczbe
	kawalek	liczba, ktora chcemy dodac na koniec chromosomu

Here is the caller graph for this function:



4.1.1.4 dodajNaKoniecListyOsobnikow()

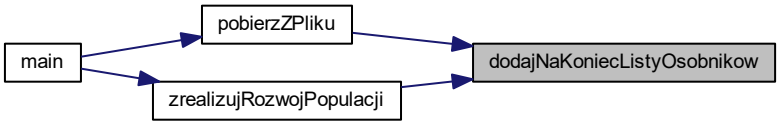
```
void dodajNaKoniecListyOsobnikow (
            osobnik *& pierwszy,
            osobnik * jakis )
```

Funkcja dodaje osobnika na koniec listy.

Parameters

in,out	*pierwszy*	wskaznik na pierwszego osobnika w liscie
	jakis	wskaznik na osobnika, ktorego chcemy dodac na koniec listy

Here is the caller graph for this function:



4.1.1.5 krzyzujOsobnikow()

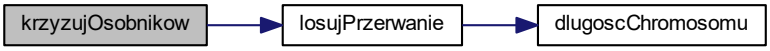
```
void krzyzujOsobnikow (
            osobnik *& osobnik1,
            osobnik *& osobnik2 )
```

Funkcja krzyzuje osobnikow z populacji.

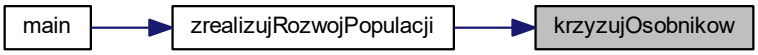
Parameters

osobnik1	wskaznik na pierwszego z osobnikow, ktorego chcemy skrzyzowac
osobnik2	wskaznik na drugiego z osobnikow, ktorego chcemy skrzyzowac

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.1.6 liczbaOsobnikow()

```
int liczbaOsobnikow (
            osobnik * pierwszy )
```

Funkcja oblicza liczbe osobnikow w liscie.

Parameters

pierwszy	wskaznik na pierwszego osobnika w liscie

Returns

liczba osobnikow w liscie

Here is the caller graph for this function:

4.1.1.7 losujOsobnika()

```
osobnik* losujOsobnika (
            osobnik * pierwszy )
```

Funkcja losuje osobnika z listy.

Parameters

pierwszy	wskaznik na pierwszego osobnika w liscie

Returns

wskaznik na wylosowanego z listy osobnika

Here is the call graph for this function:

Here is the caller graph for this function:

4.1.1.8 losujPrzerwanie()

```
lista_ch* losujPrzerwanie (
            osobnik * jakis )
```

Funkcja losuje miejsce przerwania chromosomu.

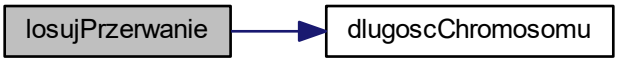
Parameters

jakis	wskaznik na osobnika, dla ktorego chcemy wylosowac miejsce przerwania chromosomu

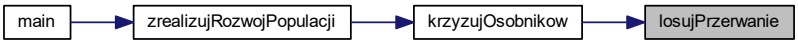
Returns

wskaznik na wylosowany kawalek chromosomu (miejsce przerwania)

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.1.9 pobierzDane()

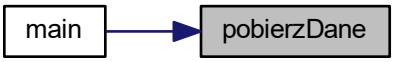
```
void pobierzDane (
            int argc,
            char * argv[],
            string & plik_wejsciowy,
            string & plik_wyjsciowy,
            double & wspolczynnik_wymierania,
            double & wspolczynnik_rozmnazania,
            int & liczba_pokolen,
            int & liczba_par )
```

Funkcja pobiera dane z konsoli i wczytuje je do zmiennych.

Parameters

	argc	liczba argumentow podanych w konsoli
	argv	tablica wskaznikow na argumenty w konsoli
in,out	*plik_wejsciowy*	nazwa sciezki do pliku wejsciowego
in,out	*plik_wyjciowy*	nazwa sciezki do pliku wyjsciowego
in,out	*wspolczynnik_wymierania*	wspolczynnik wymierania dla populacji osobnikow
in,out	*wspolczynnik_rozmnazania*	wspolczynnik rozmnazania dla populacji osobnikow
in,out	*liczba_pokolen*	numer pokolenia, ktorego postac nalezy podac w pliku koncowym
in,out	*liczba_par*	liczba par osobnikow, ktorych nalezy ze soba skrzyzowac

Here is the caller graph for this function:



4.1.1.10 pobierzZPliku()

```
bool pobierzZPliku (
            osobnik *& pierwszy,
            ifstream & wejscie )
```

Funkcja pobiera dane z pliku i umieszcza je w liscie, zwracajac informacje o poprawnosci danych w pliku.

Parameters

in,out	*pierwszy*	wskaznik na pierwszego osobnika w liscie
in,out	*wejscie*	strumien, przez ktory funkcja ma pobrac dane

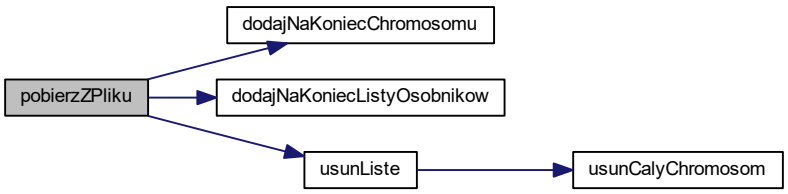
Returns

wartosc logiczna zdania "Dane w pliku maja poprawny format."

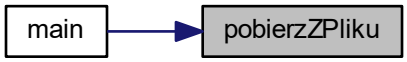
Warning

gdy dane w pliku sa niepoprawne, funcja wypisuje do konsoli odpowiedni komunikat

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.1.11 sprawdzArgumenty()

```
bool sprawdzArgumenty (
            int argc,
            ifstream & wejscie,
            double wspolczynnik_wymierania,
            double wspolczynnik_rozmnazania,
            int liczba_pokolen,
            int liczba_par )
```

Funkcja sprawdza poprawnosc podanych przez uzytkownika argumentow, zwracajac informacje o ich poprawnosci.

Parameters

	argc	liczba argumentow podanych w konsol
in,out	*wejscie*	strumien umozliwiajacy odczytanie danych z pliku wejsciowego
	wspolczynnik_wymierania	wspolczynnik wymierania dla populacji osobnikow
	wspolczynnik_rozmnazania	wspolczynnik rozmnazania dla populacji osobnikow
	liczba_pokolen	numer pokolenia, ktorego postac nalezy podac w pliku koncowym
	liczba_par	liczba par osobnikow, ktorych nalezy ze soba skrzyzowac

Returns

wartosc logiczna zdania "Argumenty sa poprawne."

Warning

gdy argumenty sa niepoprawne, funkcja wypisuje do konsoli odpowiedni komunikat

Here is the caller graph for this function:

4.1.1.12 usunCalyChromosom()

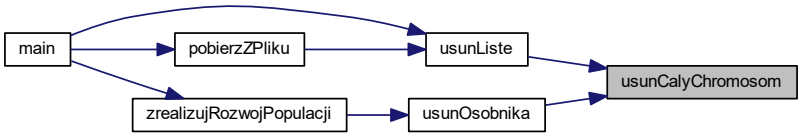
```
void usunCalyChromosom (
            lista_ch *& poczatek_chromosomu )
```

Funkcja usuwa caly chromosom.

Parameters

in,out	*poczatek_chromosomu*	wskaznik na pierwszy kawalek chromosomu, ktorego chcemy usunac

Here is the caller graph for this function:



4.1.1.13 usunKawalekChromosomu()

```
void usunKawalekChromosomu (
            lista_ch *& poczatek,
            lista_ch *& jakis,
            lista_ch *& poprzedni )
```

Funkcja usuwa dany kawalek z chromosomu.

Parameters

in,out	*poczatek*	wskaznik na pierwszy kawalek chromosomu
in,out	*wskaznik*	na kawalek chromosomu, ktorego chcemy usunac
in,out	*poprzedni*	wskaznik na poprzedni, wzgledem tego, ktorego chcemy usunac, kawalek chromosomu

4.1.1.14 usunListe()

```
void usunListe (
            osobnik *& pierwszy )
```

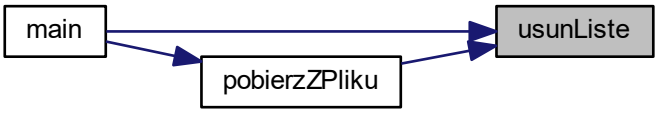
Funkcja usuwa cala liste osobnikow razem z ich chromosomami.

Parameters

in,out	*pierwszy*	wskaznik na pierwszego osobnika w liscie

Here is the call graph for this function:

Here is the caller graph for this function:



4.1.1.15 usunOsobnika()

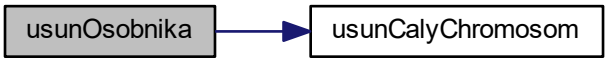
```
void usunOsobnika (
            osobnik *& pierwszy,
            osobnik *& jakis,
            osobnik *& poprzedni )
```

Funkcja usuwa danego osobnika razem z jego chromosomem.

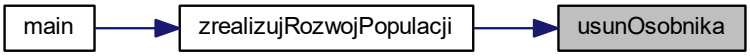
Parameters

in,out	*pierwszy*	wskaznik na pierwszego osobnika w liscie
in,out	*jakis*	wskaznik na osobnika, ktorego chcemy usunac
in,out	*poprzedni*	wskaznik na poprzedniego, wzgledem tego, ktorego chcemy usunac, osobnika

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.1.16 wypiszChromosom()

```
void wypiszChromosom (
            lista_ch * poczatek_chromosomu,
            ostream & strumien )
```

Funkcja wypisuje chromosom.

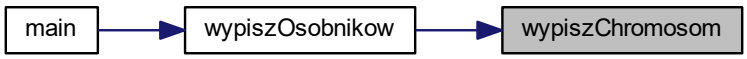
Parameters

	poczatek_chromosomu	wskaznik na poczatek wypisywanego chromosomu
in,out	*strumien*	strumien, do ktorego funkcja ma wypisac chromosom

Warning

gdy chromosom nie istnieje, funkcja wypisuje do strumienia komunikat "Osobnik nie istnieje"

Here is the caller graph for this function:



4.1.1.17 wypiszOsobnikow()

```
void wypiszOsobnikow (
            osobnik * pierwszy,
            ostream & strumien )
```

Funkcja wypisuje chromosomy osobnikow z listy.

Parameters

	pierwszy	wskaznik na pierwszego osobnika z listy
in,out	*strumien*	strumien, do ktorego funkcja ma wypisac osobnikow

Warning

gdy pierwszy osobnik nie istnieje, funkcja wypisuje do strumienia komunikat "Lista jest pusta"

Here is the call graph for this function:

Here is the caller graph for this function:

4.1.1.18 wyznaczPrzystosowanie()

```
double wyznaczPrzystosowanie (
            osobnik * jakis )
```

Funkcja wyznacza przystosowanie osobnika do warunkow srodowiska.

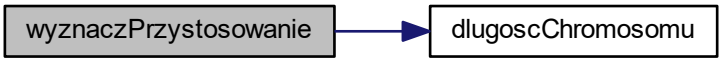
Parameters

jakis	wskaznik na osobnika, ktorego przystosowanie do warunkow srodowiska chcemy wyznaczyc

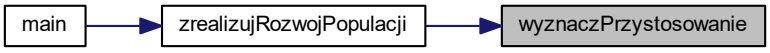
Returns

wspolczynnik przystosowania osobnika do warunkow srodowiska

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.1.19 znajdzPoprzedniKawalekChromosomu()

```
lista_ch* znajdzPoprzedniKawalekChromosomu (
            lista_ch * poczatek,
            lista_ch * jakis )
```

Funkcja znajduje poprzedni kawalek chromosomu.

Parameters

poczatek	wskaznik na pierwszy kawalek chromosomu
jakis	wskaznik na kawalek chromosomu, ktorego poprzednika chcemy znalezc

Returns

wskaznik na poprzedni kawalek chromosomu

Warning

funkcja zwraca nullptr, gdy kawalek, ktorego poprzednika szukamy jest jednoczesnie pierwszym kawalkiem chromosomu lub gdy pierwszy kawalek nie istnieje

4.1.1.20 znajdzPoprzedniOsobnik()

```
osobnik* znajdzPoprzedniOsobnik (
            osobnik * pierwszy,
            osobnik * jakis )
```

Funkcja znajduje poprzedniego osobnika w liscie.

Parameters

pierwszy	wskaznik na pierwszego osobnika w liscie
jakis	wskaznik na osobnika, ktorego poprzednika chcemy wyznaczyc

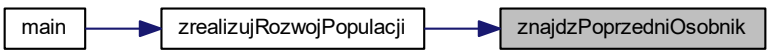
Returns

wskaznik na poprzedniego osobnika w liscie

Warning

funkcja zwraca nullptr, gdy osobnik, ktorego poprzednika szukamy, jest pierwszy w liscie lub gdy pierwszy osobnik nie istnieje

Here is the caller graph for this function:



4.1.1.21 zrealizujRozwojPopulacji()

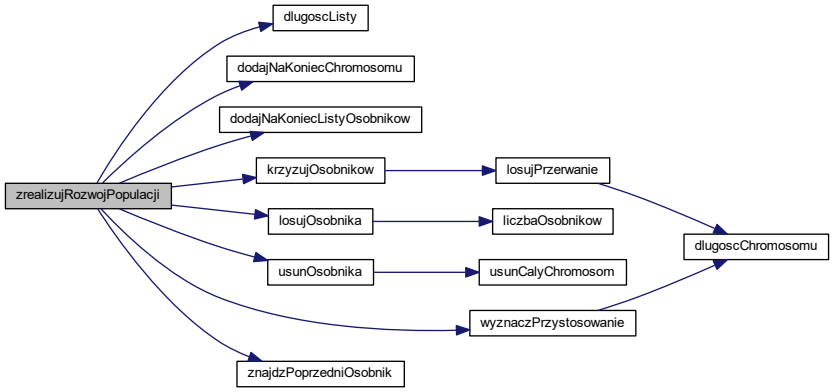
```
void zrealizujRozwojPopulacji (
            osobnik *& pierwszy,
            double wspolczynnik_rozmnazania,
            double wspolczynnik_wymierania,
            int liczba_pokolen,
            int liczba_par )
```

Funkcja modyfikuje liste osobnikow zgodnie z trescia zadania (wykonuje krzyzowania i modyfikacje w zaleznosci od przystosowania do warunkow srodowiska, wyznaczajac pokolenie szukane przez uzytkownika).

Parameters

in,out	*pierwszy*	wskaznik na pierwszego osobnika w liscie
	wspolczynnik_rozmnazania	wspolczynnik rozmnazania dla populacji osobnikow
	wspolczynnik_wymierania	wspolczynnik wymierania dla populacji osobnikow
	liczba_pokolen	numer pokolenia, ktorego postac nalezy podac w pliku koncowym
	liczba_par	liczba par osobnikow, ktorych nalezy ze soba skrzyzowac

Here is the call graph for this function:

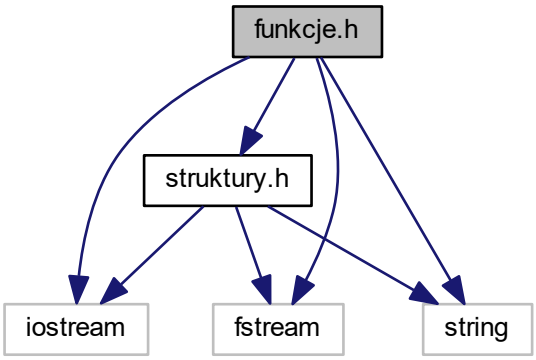


Here is the caller graph for this function:

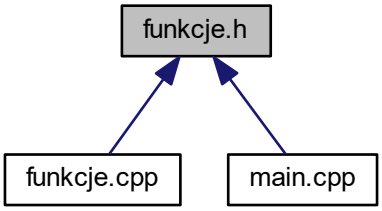
4.2 funkcje.h File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include "struktury.h"
```

Include dependency graph for funkcje.h:



This graph shows which files directly or indirectly include this file:



Functions

- void pobierzDane (int argc, char ∗argv[ ], string &plik_wejsciowy, string &plik_wyjsciowy, double &wspolczynnik_wymierania, double &wspolczynnik_rozmnazania, int &liczba_pokolen, int &liczba_par)
- bool sprawdzArgumenty (int argc, ifstream &wejscie, double wspolczynnik_wymierania, double wspolczynnik_rozmnazania, int liczba_pokolen, int liczba_par)
- void dodajNaKoniecListyOsobnikow (osobnik ∗&pierwszy, osobnik ∗jakis)
- void dodajNaKoniecChromosomu (osobnik ∗&wskaznik_na_osobnika, int kawalek)
- bool pobierzZPliku (osobnik ∗&pierwszy, ifstream &wejscie)
- void wypiszChromosom (lista_ch ∗poczatek_chromosomu, ostream &strumien)
- void wypiszOsobnikow (osobnik ∗pierwszy, ostream &strumien)
- int liczbaOsobnikow (osobnik ∗pierwszy)
- int dlugoscChromosomu (lista_ch ∗poczatek_chromosomu)
- int dlugoscListy (osobnik ∗pierwszy)
- osobnik ∗ losujOsobnika (osobnik ∗pierwszy)
- lista_ch ∗ losujPrzerwanie (osobnik ∗jakis)

- void krzyzujOsobnikow (osobnik ∗&osobnik1, osobnik ∗&osobnik2)
- void zrealizujRozwojPopulacji (osobnik ∗&pierwszy, double wspolczynnik_rozmnazania, double wspolczynnik↩ _wymierania, int liczba_pokolen, int liczba_par)
- double wyznaczPrzystosowanie (osobnik ∗jakis)
- osobnik ∗ znajdzPoprzedniOsobnik (osobnik ∗pierwszy, osobnik ∗jakis)
- lista_ch ∗ znajdzPoprzedniKawalekChromosomu (lista_ch ∗poczatek, lista_ch ∗jakis)
- void usunKawalekChromosomu (lista_ch ∗&poczatek, lista_ch ∗&jakis, lista_ch ∗&poprzedni)
- void usunCalyChromosom (lista_ch ∗&poczatek_chromosomu)
- void usunOsobnika (osobnik ∗&pierwszy, osobnik ∗&jakis, osobnik ∗&poprzedni)
- void usunListe (osobnik ∗&pierwszy)

4.2.1 Function Documentation

4.2.1.1 dlugoscChromosomu()

```
int dlugoscChromosomu (
            lista_ch * poczatek_chromosomu )
```

Funkcja oblicza dlugosc chromosomu.

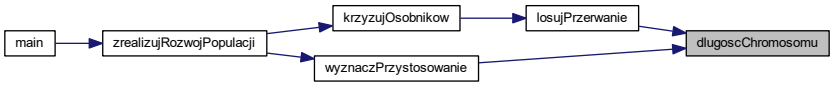
Parameters

poczatek_chromosomu	wskaznik na pierwszy kawalek chromosomu

Returns

dlugosc chromosomu

Here is the caller graph for this function:



4.2.1.2 dlugoscListy()

```
int dlugoscListy (
            osobnik * pierwszy )
```

Funkcja oblicza dlugosc listy osobnikow.

Parameters

pierwszy	wskaznik na pierwszego osobnika w liscie

Returns

dlugosc listy osobnikow

Here is the caller graph for this function:

4.2.1.3 dodajNaKoniecChromosomu()

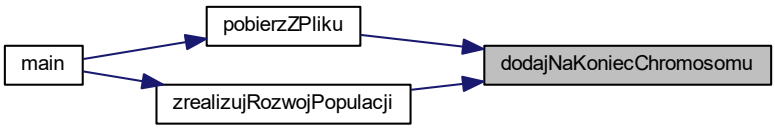
```
void dodajNaKoniecChromosomu (
            osobnik *& wskaznik_na_osobnika,
            int kawalek )
```

Funkcja dodaje liczbe na koniec chromosomu.

Parameters

in,out	*wskaznik_na_osobnika*	wskaznik na osobnika, na ktorego koniec chromosomu chcemy dodac dana liczbe
	kawalek	liczba, ktora chcemy dodac na koniec chromosomu

Here is the caller graph for this function:



4.2.1.4 dodajNaKoniecListyOsobnikow()

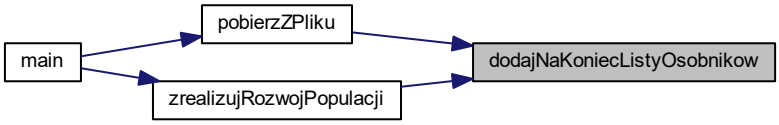
```
void dodajNaKoniecListyOsobnikow (
            osobnik *& pierwszy,
            osobnik * jakis )
```

Funkcja dodaje osobnika na koniec listy.

Parameters

in,out	*pierwszy*	wskaznik na pierwszego osobnika w liscie
	jakis	wskaznik na osobnika, ktorego chcemy dodac na koniec listy

Here is the caller graph for this function:



4.2.1.5 krzyzujOsobnikow()

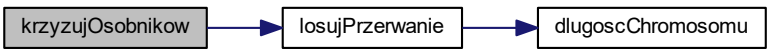
```
void krzyzujOsobnikow (
            osobnik *& osobnik1,
            osobnik *& osobnik2 )
```

Funkcja krzyzuje osobnikow z populacji.

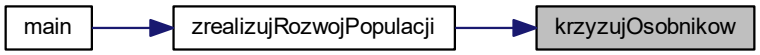
Parameters

osobnik1	wskaznik na pierwszego z osobnikow, ktorego chcemy skrzyzowac
osobnik2	wskaznik na drugiego z osobnikow, ktorego chcemy skrzyzowac

Here is the call graph for this function:



Here is the caller graph for this function:



4.2.1.6 liczbaOsobnikow()

```
int liczbaOsobnikow (
            osobnik * pierwszy )
```

Funkcja oblicza liczbe osobnikow w liscie.

Parameters

pierwszy	wskaznik na pierwszego osobnika w liscie

Returns

liczba osobnikow w liscie

Here is the caller graph for this function:

4.2.1.7 losujOsobnika()

```
osobnik* losujOsobnika (
            osobnik * pierwszy )
```

Funkcja losuje osobnika z listy.

Parameters

pierwszy	wskaznik na pierwszego osobnika w liscie

Returns

wskaznik na wylosowanego z listy osobnika

Here is the call graph for this function:

Here is the caller graph for this function:

4.2.1.8 losujPrzerwanie()

```
lista_ch* losujPrzerwanie (
            osobnik * jakis )
```

Funkcja losuje miejsce przerwania chromosomu.

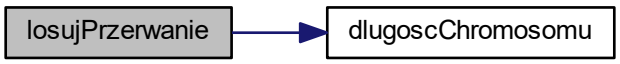
Parameters

jakis	wskaznik na osobnika, dla ktorego chcemy wylosowac miejsce przerwania chromosomu

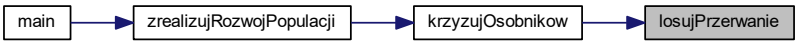
Returns

wskaznik na wylosowany kawalek chromosomu (miejsce przerwania)

Here is the call graph for this function:



Here is the caller graph for this function:



4.2.1.9 pobierzDane()

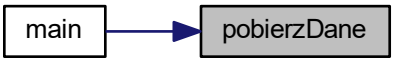
```
void pobierzDane (
            int argc,
            char * argv[],
            string & plik_wejsciowy,
            string & plik_wyjsciowy,
            double & wspolczynnik_wymierania,
            double & wspolczynnik_rozmnazania,
            int & liczba_pokolen,
            int & liczba_par )
```

Funkcja pobiera dane z konsoli i wczytuje je do zmiennych.

Parameters

	argc	liczba argumentow podanych w konsoli
	argv	tablica wskaznikow na argumenty w konsoli
in,out	*plik_wejsciowy*	nazwa sciezki do pliku wejsciowego
in,out	*plik_wyjciowy*	nazwa sciezki do pliku wyjsciowego
in,out	*wspolczynnik_wymierania*	wspolczynnik wymierania dla populacji osobnikow
in,out	*wspolczynnik_rozmnazania*	wspolczynnik rozmnazania dla populacji osobnikow
in,out	*liczba_pokolen*	numer pokolenia, ktorego postac nalezy podac w pliku koncowym
in,out	*liczba_par*	liczba par osobnikow, ktorych nalezy ze soba skrzyzowac

Here is the caller graph for this function:



4.2.1.10 pobierzZPliku()

```
bool pobierzZPliku (
            osobnik *& pierwszy,
            ifstream & wejscie )
```

Funkcja pobiera dane z pliku i umieszcza je w liscie, zwracajac informacje o poprawnosci danych w pliku.

Parameters

in,out	*pierwszy*	wskaznik na pierwszego osobnika w liscie
in,out	*wejscie*	strumien, przez ktory funkcja ma pobrac dane

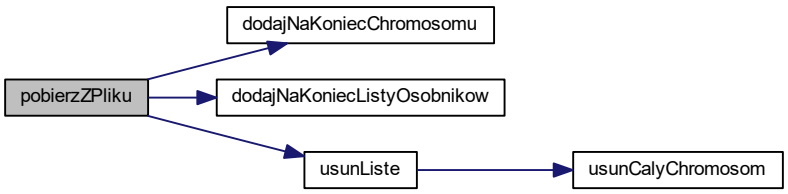
Returns

wartosc logiczna zdania "Dane w pliku maja poprawny format."

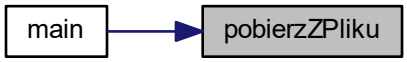
Warning

gdy dane w pliku sa niepoprawne, funcja wypisuje do konsoli odpowiedni komunikat

Here is the call graph for this function:



Here is the caller graph for this function:



4.2.1.11 sprawdzArgumenty()

```
bool sprawdzArgumenty (
            int argc,
            ifstream & wejscie,
            double wspolczynnik_wymierania,
            double wspolczynnik_rozmnazania,
            int liczba_pokolen,
            int liczba_par )
```

Funkcja sprawdza poprawnosc podanych przez uzytkownika argumentow, zwracajac informacje o ich poprawnosci.

Parameters

	argc	liczba argumentow podanych w konsol
in,out	*wejscie*	strumien umozliwiajacy odczytanie danych z pliku wejsciowego
	wspolczynnik_wymierania	wspolczynnik wymierania dla populacji osobnikow
	wspolczynnik_rozmnazania	wspolczynnik rozmnazania dla populacji osobnikow
	liczba_pokolen	numer pokolenia, ktorego postac nalezy podac w pliku koncowym
	liczba_par	liczba par osobnikow, ktorych nalezy ze soba skrzyzowac

Returns

wartosc logiczna zdania "Argumenty sa poprawne."

Warning

gdy argumenty sa niepoprawne, funkcja wypisuje do konsoli odpowiedni komunikat

Here is the caller graph for this function:

4.2.1.12 usunCalyChromosom()

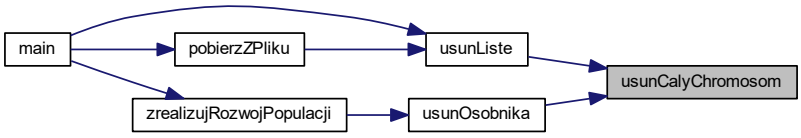
```
void usunCalyChromosom (
            lista_ch *& poczatek_chromosomu )
```

Funkcja usuwa caly chromosom.

Parameters

in,out	*poczatek_chromosomu*	wskaznik na pierwszy kawalek chromosomu, ktorego chcemy usunac

Here is the caller graph for this function:



4.2.1.13 usunKawalekChromosomu()

```
void usunKawalekChromosomu (
            lista_ch *& poczatek,
            lista_ch *& jakis,
            lista_ch *& poprzedni )
```

Funkcja usuwa dany kawalek z chromosomu.

Parameters

in,out	*poczatek*	wskaznik na pierwszy kawalek chromosomu
in,out	*wskaznik*	na kawalek chromosomu, ktorego chcemy usunac
in,out	*poprzedni*	wskaznik na poprzedni, wzgledem tego, ktorego chcemy usunac, kawalek chromosomu

4.2.1.14 usunListe()

```
void usunListe (
            osobnik *& pierwszy )
```

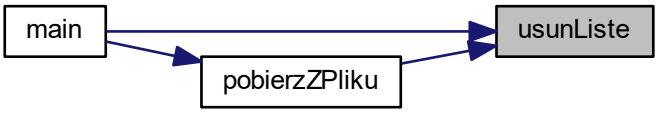
Funkcja usuwa cala liste osobnikow razem z ich chromosomami.

Parameters

in,out	*pierwszy*	wskaznik na pierwszego osobnika w liscie

Here is the call graph for this function:

Here is the caller graph for this function:



4.2.1.15 usunOsobnika()

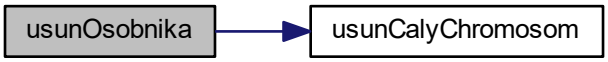
```
void usunOsobnika (
            osobnik *& pierwszy,
            osobnik *& jakis,
            osobnik *& poprzedni )
```

Funkcja usuwa danego osobnika razem z jego chromosomem.

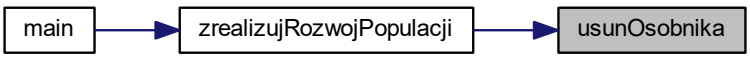
Parameters

in,out	*pierwszy*	wskaznik na pierwszego osobnika w liscie
in,out	*jakis*	wskaznik na osobnika, ktorego chcemy usunac
in,out	*poprzedni*	wskaznik na poprzedniego, wzgledem tego, ktorego chcemy usunac, osobnika

Here is the call graph for this function:



Here is the caller graph for this function:



4.2.1.16 wypiszChromosom()

```
void wypiszChromosom (
            lista_ch * poczatek_chromosomu,
            ostream & strumien )
```

Funkcja wypisuje chromosom.

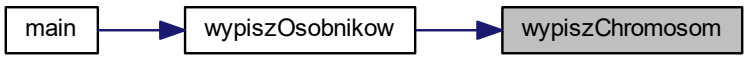
Parameters

	poczatek_chromosomu	wskaznik na poczatek wypisywanego chromosomu
in,out	*strumien*	strumien, do ktorego funkcja ma wypisac chromosom

Warning

gdy chromosom nie istnieje, funkcja wypisuje do strumienia komunikat "Osobnik nie istnieje"

Here is the caller graph for this function:



4.2.1.17 wypiszOsobnikow()

```
void wypiszOsobnikow (
            osobnik * pierwszy,
            ostream & strumien )
```

Funkcja wypisuje chromosomy osobnikow z listy.

Parameters

	pierwszy	wskaznik na pierwszego osobnika z listy
in,out	*strumien*	strumien, do ktorego funkcja ma wypisac osobnikow

Warning

gdy pierwszy osobnik nie istnieje, funkcja wypisuje do strumienia komunikat "Lista jest pusta"

Here is the call graph for this function:

Here is the caller graph for this function:

4.2.1.18 wyznaczPrzystosowanie()

```
double wyznaczPrzystosowanie (
            osobnik * jakis )
```

Funkcja wyznacza przystosowanie osobnika do warunkow srodowiska.

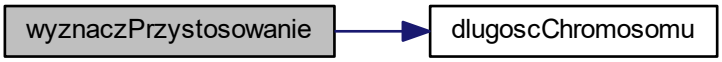
Parameters

jakis	wskaznik na osobnika, ktorego przystosowanie do warunkow srodowiska chcemy wyznaczyc

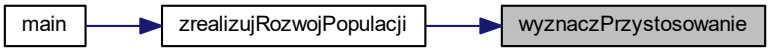
Returns

wspolczynnik przystosowania osobnika do warunkow srodowiska

Here is the call graph for this function:



Here is the caller graph for this function:



4.2.1.19 znajdzPoprzedniKawalekChromosomu()

```
lista_ch* znajdzPoprzedniKawalekChromosomu (
            lista_ch * poczatek,
            lista_ch * jakis )
```

Funkcja znajduje poprzedni kawalek chromosomu.

Parameters

poczatek	wskaznik na pierwszy kawalek chromosomu
jakis	wskaznik na kawalek chromosomu, ktorego poprzednika chcemy znalezc

Returns

wskaznik na poprzedni kawalek chromosomu

Warning

funkcja zwraca nullptr, gdy kawalek, ktorego poprzednika szukamy jest jednoczesnie pierwszym kawalkiem chromosomu lub gdy pierwszy kawalek nie istnieje

4.2.1.20 znajdzPoprzedniOsobnik()

```
osobnik* znajdzPoprzedniOsobnik (
            osobnik * pierwszy,
            osobnik * jakis )
```

Funkcja znajduje poprzedniego osobnika w liscie.

Parameters

pierwszy	wskaznik na pierwszego osobnika w liscie
jakis	wskaznik na osobnika, ktorego poprzednika chcemy wyznaczyc

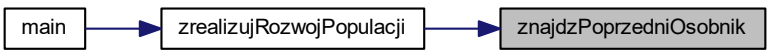
Returns

wskaznik na poprzedniego osobnika w liscie

Warning

funkcja zwraca nullptr, gdy osobnik, ktorego poprzednika szukamy, jest pierwszy w liscie lub gdy pierwszy osobnik nie istnieje

Here is the caller graph for this function:



4.2.1.21 zrealizujRozwojPopulacji()

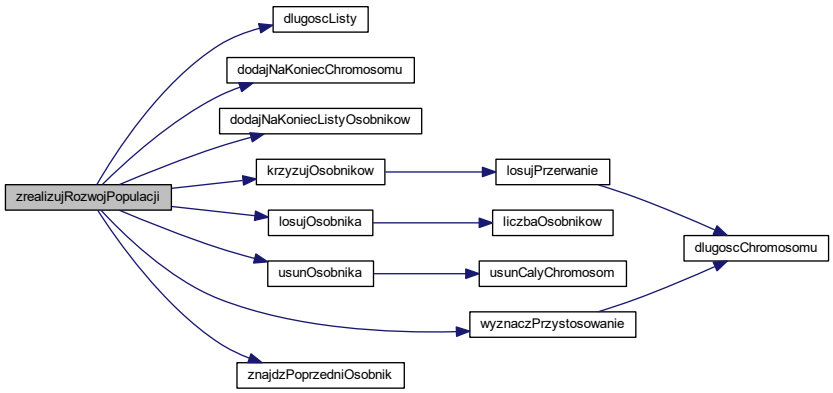
```
void zrealizujRozwojPopulacji (
            osobnik *& pierwszy,
            double wspolczynnik_rozmnazania,
            double wspolczynnik_wymierania,
            int liczba_pokolen,
            int liczba_par )
```

Funkcja modyfikuje liste osobnikow zgodnie z trescia zadania (wykonuje krzyzowania i modyfikacje w zaleznosci od przystosowania do warunkow srodowiska, wyznaczajac pokolenie szukane przez uzytkownika).

Parameters

in,out	*pierwszy*	wskaznik na pierwszego osobnika w liscie
	wspolczynnik_rozmnazania	wspolczynnik rozmnazania dla populacji osobnikow
	wspolczynnik_wymierania	wspolczynnik wymierania dla populacji osobnikow
	liczba_pokolen	numer pokolenia, ktorego postac nalezy podac w pliku koncowym
	liczba_par	liczba par osobnikow, ktorych nalezy ze soba skrzyzowac

Here is the call graph for this function:



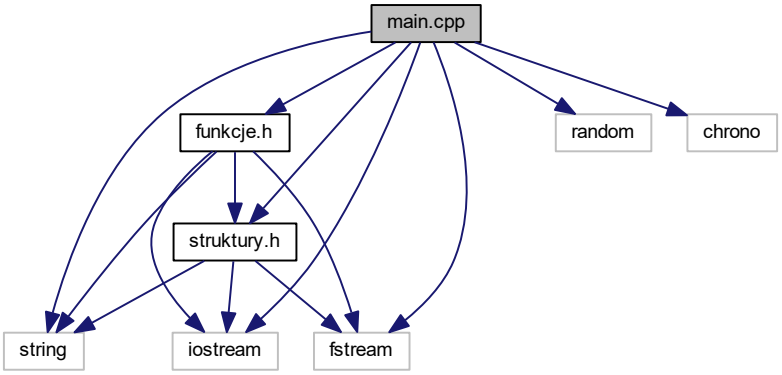
Here is the caller graph for this function:

4.3 main.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include <random>
#include <chrono>
#include "funkcje.h"
```

```
#include "struktury.h"
```

Include dependency graph for main.cpp:



Functions

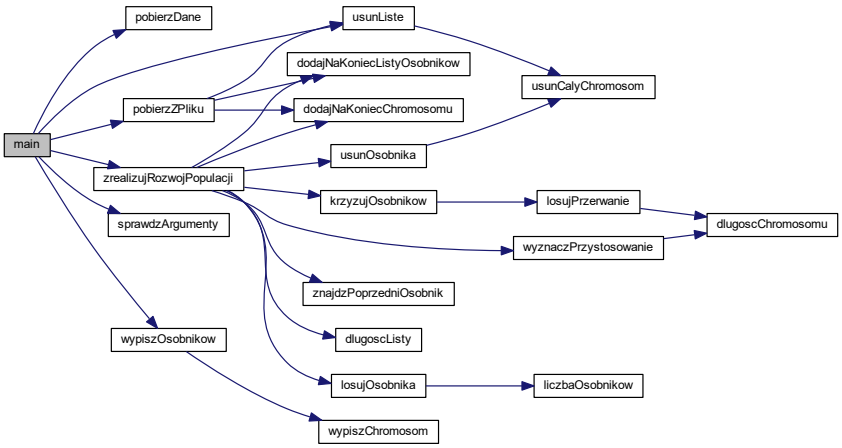
- int main (int argc, char ∗argv[ ])

4.3.1 Function Documentation

4.3.1.1 main()

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

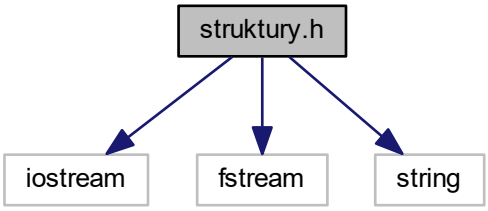
Here is the call graph for this function:



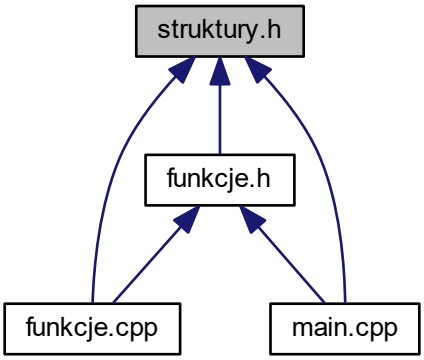
4.4 struktury.h File Reference

```
#include <iostream>
#include <fstream>
#include <string>
```

Include dependency graph for struktury.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct lista_ch
- struct osobnik

Index