Politechnika Śląska w Gliwicach
Wydział Automatyki, Elektroniki i Informatyki

Programowanie Komputerów 3

GRA W ŻYCIE

autor	Tomasz Sojka
prowadzący	mg inż. Grzegorz Kwiatkowski
rok akademicki	2018/2019
kierunek	informatyka
rodzaj studiów	SSI
semestr	3
termin laboratorium / ćwiczeń	wtorek, 13:45 – 15:15
grupa	2

1 Informacja

1.1 Diagram UML

Pełen diagram UML mozna znalezc pod linkiem:
`https://go.gliffy.com/go/share/sbejb2lxs2s3y1t601g8`

1.2 Kod i pliki testowe

Pełny kod programu oraz pliki tekstowe do testowania znajdują się na moim repozytorium:
`https://github.com/tomaszsojka/my/tree/master/projekt_pk3`

1.3 Specyfikacja wewnętrzna

Poniżej znajduje się dokumentacja klas i plików.
Między innymi opis klas i ich metod, opis funkcji oraz operatorów, grafy wywołania, diagramy dziedziczenia i współpracy klas.
Zaawansowany c++: W programie użyto wątków w celu synchronizacji dwóch metod klasy menue. Ich opis można znaleźć w dokumentacji metod klasy menue: **petla** i **zatrzymaj_petle**, oraz w pełnym kodzie w pliku metody_menue.cpp w metodzie **poprowadz_uzytkownika**.

GRA W ZYCIE

Wygenerowano przez Doxygen 1.8.14

Spis treści

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

Rozdział 2

Indeks klas

2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

Rozdział 3

Indeks plików

3.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

Rozdział 4

Dokumentacja klas

4.1 Dokumentacja klasy dane_planszy

`#include <klasa_dane_planszy.h>`

Diagram dziedziczenia dla dane_planszy

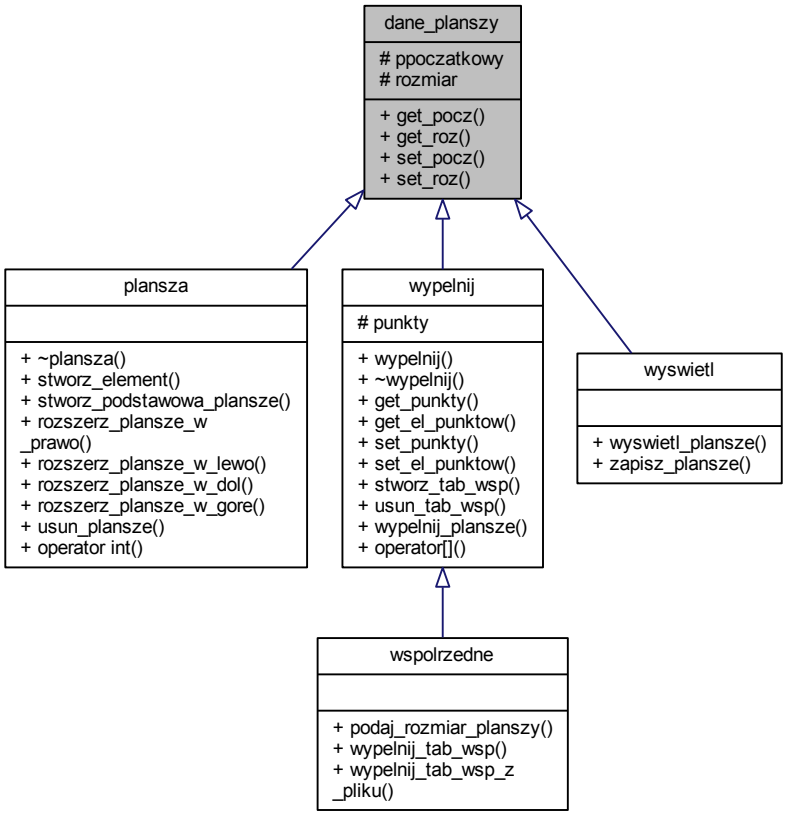
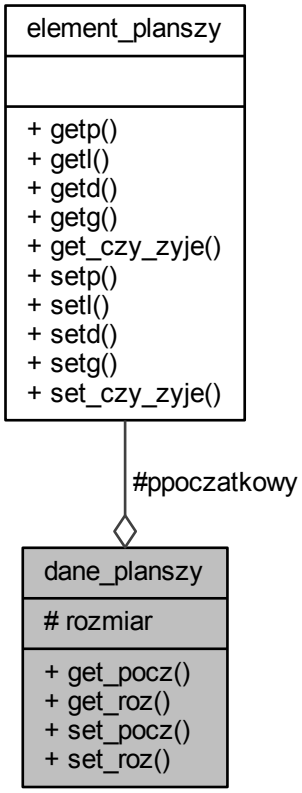


Diagram współpracy dla dane_planszy:



Metody publiczne

- element_planszy ∗ get_pocz ()
- int get_roz ()
- void set_pocz (element_planszy ∗pp)
- void set_roz (int r)

Atrybuty chronione

- element_planszy ∗ ppoczatkowy
- int rozmiar

4.1.1 Opis szczegółowy

Klasa zawiera podtawowe dane planszy do gry w zycie. Jest klasa bazowa klas: plansza, wyjwietl, wypelnij(po ktorej dziedziczy klasa wspolrzedne).

4.1.2 Dokumentacja funkcji składowych

4.1.2.1 get_pocz()

```
element_planszy * dane_planszy::get_pocz ( )
```

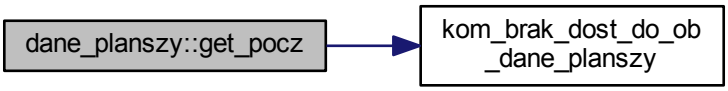
Metoda zwraca pole ppoczatkowy.

Metoda zostala stworzona aby uzyskac dostep do pola ppoczatkowy przez klasy niezaprzyjaznione i niepochodne. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

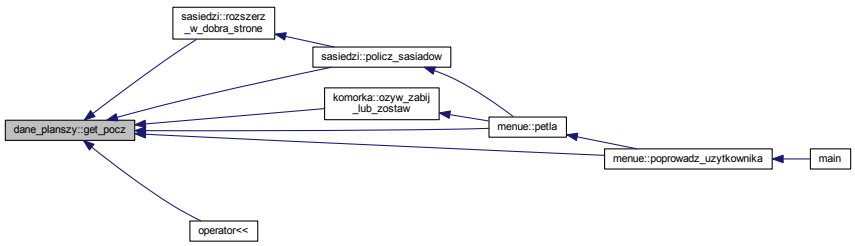
Zwraca

ppoczatkowy, jesli jest dotep do obiektu klasy dane_planszy. Jesli nie, zwraca nullptr.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.1.2.2 get_roz()

```
int dane_planszy::get_roz ( )
```

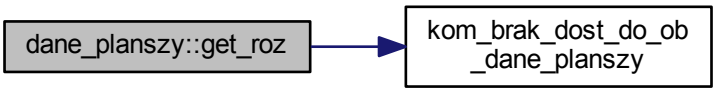
Metoda zwraca pole rozmiar.

Metoda zostala stworzona aby uzyskac dostep do pola rozmiar przez klasy niezaprzyjaznione i niepochodne. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

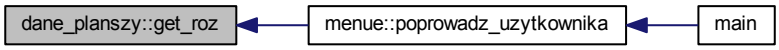
Zwraca

rozmiar, jesli jest dotep do obiektu klasy dane_planszy. Jesli nie, zwraca -1.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.1.2.3 set_pocz()

```
void dane_planszy::set_pocz (
            element_planszy * pp )
```

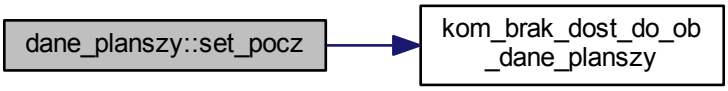
Metoda ustawia pole ppoczatkowy.

Metoda zostala stworzona aby uzyskac dostep i pozwolic na zmiane wartosci pola ppoczatkowy przez klasy niezaprzyjaznione i niepochodne. Do pola przypisywana jest wartosc przekazana przez argument. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

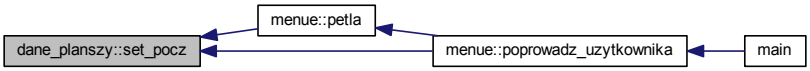
Parametry

pp	wskaznik na komorke w lewym gornym rogu.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.1.2.4 set_roz()

```
void dane_planszy::set_roz (
            int r )
```

Metoda ustawia pole rozmiar.

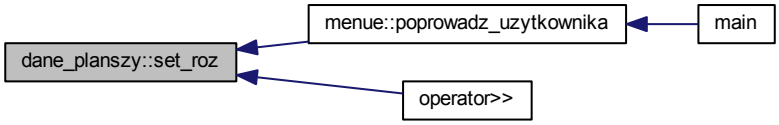
Metoda zostala stworzona aby uzyskac dostep i pozwolic na zmiane wartosci pola rozmiar przez klasy niezaprzyjaznione i niepochodne. Do pola przypisywana jest wartosc przekazana przez argument. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

Parametry

r	dlugosc boku podstawowej planszy kwadratowej.

Oto graf wywołań dla tej funkcji:

Oto graf wywoływań tej funkcji:



4.1.3 Dokumentacja atrybutów składowych

4.1.3.1 ppoczatkowy

```
element_planszy* dane_planszy::ppoczatkowy  [protected]
```

Pole ppoczatkowy jest wskaznikiem na pierwszy element planszy, ktory znajduje sie w lewym gornym rogu.

4.1.3.2 rozmiar

```
int dane_planszy::rozmiar  [protected]
```

Pole rozmiar przechowuje rozmiar kwadratowej poczatkowej planszy (dlugosc boku), ktora podal uzytkownik, tak aby zmiescily sie w niej wszystkie komorki do ozywienia.

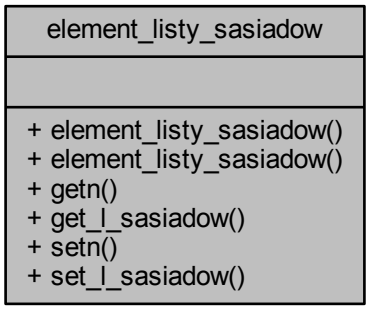
Dokumentacja dla tej klasy została wygenerowana z plików:

- klasa_dane_planszy.h
- metody_dane_planszy.cpp

4.2 Dokumentacja klasy element_listy_sasiadow

`#include <klasa_element_listy_sasiadow.h>`

Diagram współpracy dla element_listy_sasiadow:



Metody publiczne

- element_listy_sasiadow ()
- element_listy_sasiadow (element_listy_sasiadow ∗pn, int l_s)
- element_listy_sasiadow ∗ getn ()
- int get_l_sasiadow ()
- void setn (element_listy_sasiadow ∗pn)
- void set_l_sasiadow (int l_s)

4.2.1 Opis szczegółowy

Klasa reprezentuje element listy sasiadow. Obiekt klasy zawiera dwa pola : wskaznik na nastepny element listy oraz liczbe sasiadow danego elementu planszy.

4.2.2 Dokumentacja konstruktora i destruktora

4.2.2.1 element_listy_sasiadow() [1/2]

`element_listy_sasiadow::element_listy_sasiadow ( )`

Konstruktor klasy element_listy_sasiadow.

Nie ustawia zadnych wartosci. Zostal stworzony na potrzebe konstuktora wieloargumentowego.

4.2.2.2 element_listy_sasiadow() [2/2]

```
element_listy_sasiadow::element_listy_sasiadow (
            element_listy_sasiadow * pn,
            int l_s )
```

Konstruktor klasy element_listy_sasiadow.

Konstruktor przypisuje polom wartosci podane jako argumenty. Zostal stworzony, aby szybciej tworzyc obiekty klasy element_listy_sasiadow.

4.2.3 Dokumentacja funkcji składowych

4.2.3.1 get_l_sasiadow()

```
int element_listy_sasiadow::get_l_sasiadow ( )
```

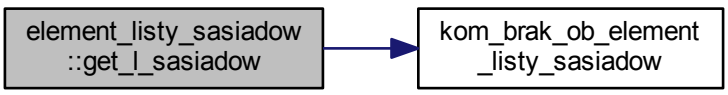
Metoda zwraca pole liczba_sasiadow.

Metoda zostala stworzona aby uzyskac dostep do pola liczba_sasiadow przez klasy niezaprzyjaznione i niepochodne. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

Zwraca

rozmiar, jesli jest dotep do obiektu klasy dane_planszy. Jesli nie, zwraca -1.

Oto graf wywołań dla tej funkcji:



4.2.3.2 getn()

```
element_listy_sasiadow * element_listy_sasiadow::getn ( )
```

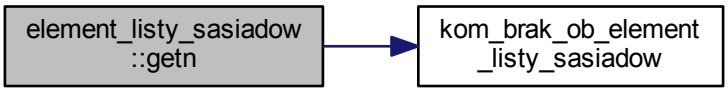
Metoda zwraca pole pnext.

Metoda zostala stworzona aby uzyskac dostep do pola pnext przez klasy niezaprzyjaznione i niepochodne. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

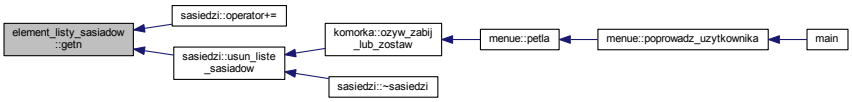
Zwraca

ppoczatkowy, jesli jest dotep do obiektu klasy dane_planszy. Jesli nie, zwraca nullptr.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.2.3.3 set_l_sasiadow()

```
void element_listy_sasiadow::set_l_sasiadow (
            int l_s )
```

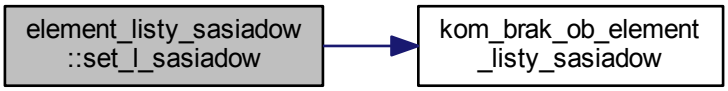
Metoda ustawia pole liczba_sasiadow.

Metoda zostala stworzona aby uzyskac dostep i pozwolic na zmiane pola liczba_sasiadow przez klasy niezaprzyjaznione i niepochodne. Do pola przypisywana jest wartosc przekazana przez argument. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

Parametry

l↩ _↩ *s*	dlugosc boku podstawowej planszy kwadratowej.

Oto graf wywołań dla tej funkcji:



4.2.3.4 setn()

```
void element_listy_sasiadow::setn (
            element_listy_sasiadow * pn )
```

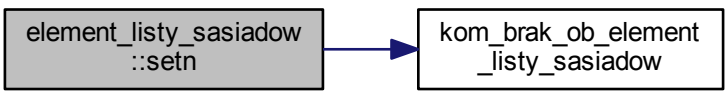
Metoda ustawia pole pnext.

Metoda zostala stworzona aby uzyskac dostep i pozwolic na zmiane wartosci pola pnext przez klasy niezaprzyjaznione i niepochodne. Do pola przypisywana jest wartosc przekazana przez argument. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

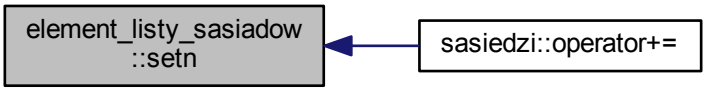
Parametry

pn	wskaznik na nastepny element listy sasiadow.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



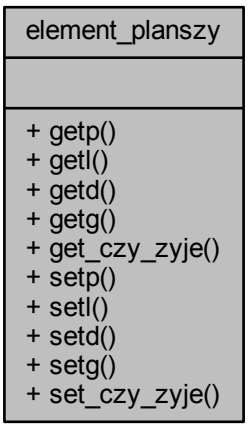
Dokumentacja dla tej klasy została wygenerowana z plików:

- klasa_element_listy_sasiadow.h
- metody_element_listy_sasiadow.cpp

4.3 Dokumentacja klasy element_planszy

`#include <klasa_element_planszy.h>`

Diagram współpracy dla element_planszy:



Metody publiczne

- element_planszy ∗ getp ()
- element_planszy ∗ getl ()
- element_planszy ∗ getd ()
- element_planszy ∗ getg ()
- bool get_czy_zyje ()
- void setp (element_planszy ∗pp)
- void setl (element_planszy ∗pl)
- void setd (element_planszy ∗pd)
- void setg (element_planszy ∗pg)
- void set_czy_zyje (bool zyje)

4.3.1 Opis szczegółowy

Klasa reprezentuje element planszy do gry w zycie. Obiekt klasy zawiera trzy pola wskaznikowe zawierajace adresy sasiadow danego elementu oraz pole typu bool, ktore informuje o tym, czy komorka zyje .

4.3.2 Dokumentacja funkcji składowych

4.3.2.1 get_czy_zyje()

```
bool element_planszy::get_czy_zyje ( )
```

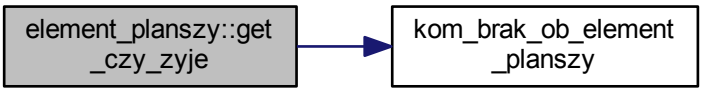
Metoda zwraca pole czy_zyje.

Metoda zostala stworzona aby uzyskac dostep do pola pgorny przez klasy niezaprzyjaznione i niepochodne. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

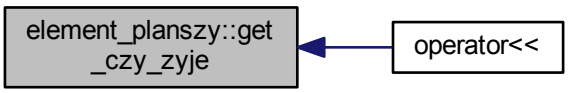
Zwraca

pgorny, jesli jest dotep do obiektu klasy dane_planszy. Jesli nie, zwraca nullptr.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.3.2.2 getd()

```
element_planszy * element_planszy::getd ( )
```

Metoda zwraca pole pdolny.

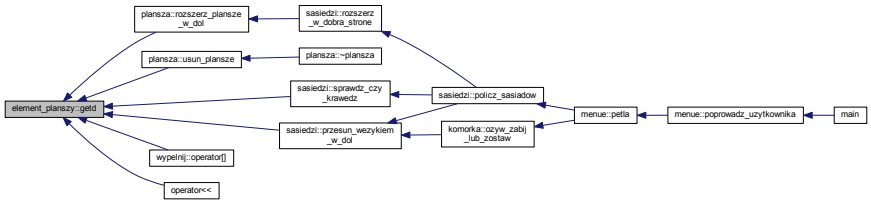
Metoda zostala stworzona aby uzyskac dostep do pola pdolny przez klasy niezaprzyjaznione i niepochodne. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

Zwraca

pdolny, jesli jest dotep do obiektu klasy dane_planszy. Jesli nie, zwraca nullptr.

Oto graf wywołań dla tej funkcji:

Oto graf wywoływań tej funkcji:



4.3.2.3 getg()

```
element_planszy * element_planszy::getg ( )
```

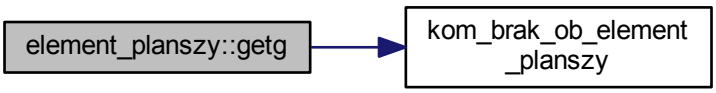
Metoda zwraca pole pgorny.

Metoda zostala stworzona aby uzyskac dostep do pola pgorny przez klasy niezaprzyjaznione i niepochodne. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

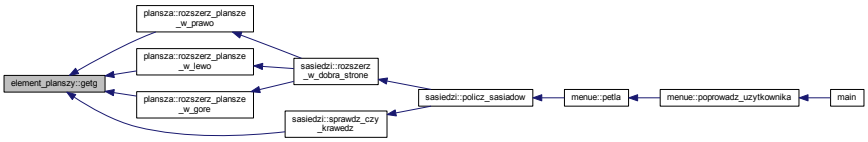
Zwraca

pgorny, jesli jest dotep do obiektu klasy dane_planszy. Jesli nie, zwraca nullptr.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.3.2.4 getl()

```
element_planszy * element_planszy::getl ( )
```

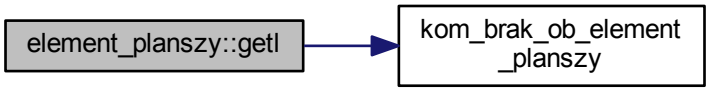
Metoda zwraca pole plewy.

Metoda zostala stworzona aby uzyskac dostep do pola plewy przez klasy niezaprzyjaznione i niepochodne. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

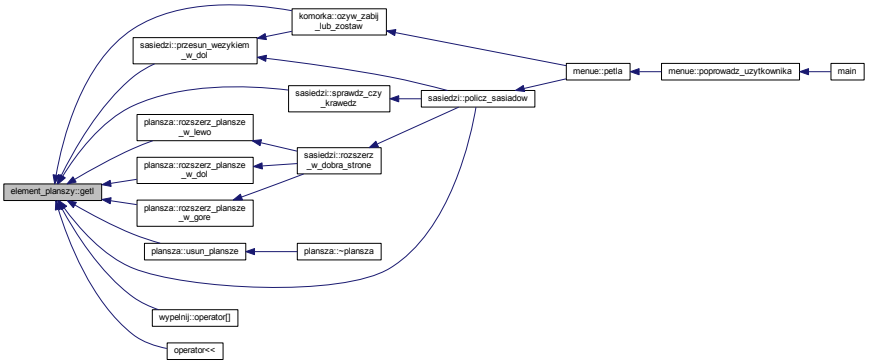
Zwraca

plewy, jesli jest dotep do obiektu klasy dane_planszy. Jesli nie, zwraca nullptr.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.3.2.5 getp()

```
element_planszy * element_planszy::getp ( )
```

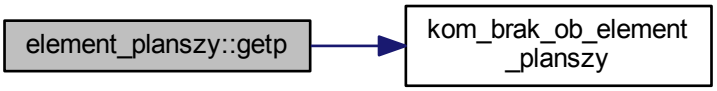
Metoda zwraca pole pprawy.

Metoda zostala stworzona aby uzyskac dostep do pola pprawy przez klasy niezaprzyjaznione i niepochodne. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

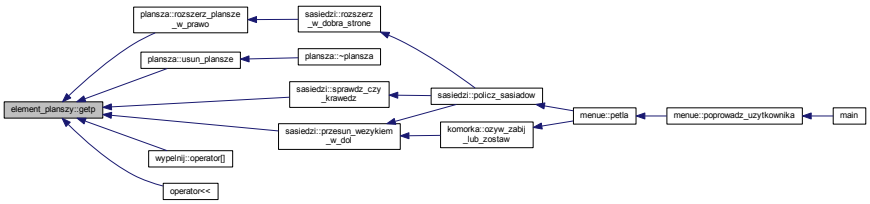
Zwraca

pprawy, jesli jest dotep do obiektu klasy dane_planszy. Jesli nie, zwraca nullptr.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.3.2.6 set_czy_zyje()

```
void element_planszy::set_czy_zyje (
            bool zyje )
```

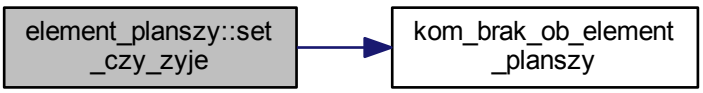
Metoda ustawia pole czy_zyje.

Metoda zostala stworzona aby uzyskac dostep i pozwolic na zmiane pola czy_zyje przez klasy niezaprzyjaznione i niepochodne. Do pola przypisywana jest wartosc przekazana przez argument. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

Parametry

zyje	stan w jaki komorka ma byc wprowadzona (ozywiona lub zabita).

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

4.3.2.7 setd()

```
void element_planszy::setd (
            element_planszy * pd )
```

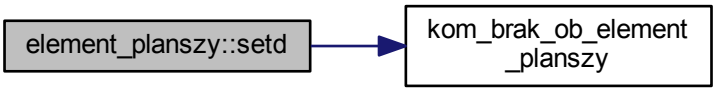
Metoda ustawia pole pdolny.

Metoda zostala stworzona aby uzyskac dostep i pozwolic na zmiane pola pdolny przez klasy niezaprzyjaznione i niepochodne. Do pola przypisywana jest wartosc przekazana przez argument. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

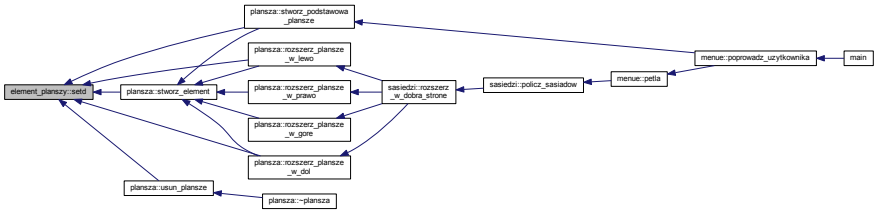
Parametry

pd	wskaznik na element na dole.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.3.2.8 setg()

```
void element_planszy::setg (
            element_planszy * pg )
```

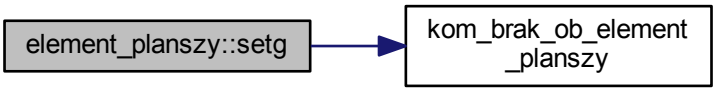
Metoda ustawia pole pgorny.

Metoda zostala stworzona aby uzyskac dostep i pozwolic na zmiane pola pgorny przez klasy niezaprzyjaznione i niepochodne. Do pola przypisywana jest wartosc przekazana przez argument. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

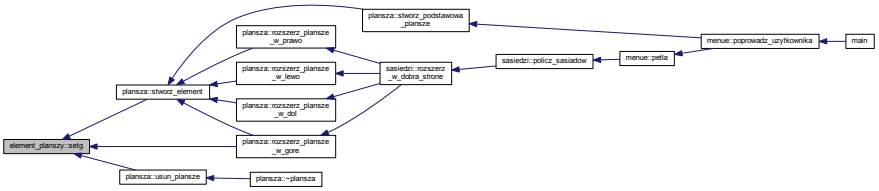
Parametry

pg	wskaznik na element na gorze.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.3.2.9 setl()

```
void element_planszy::setl (
            element_planszy * pl )
```

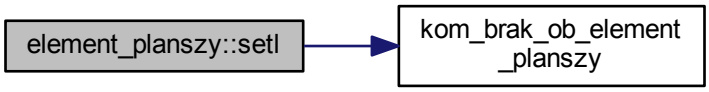
Metoda ustawia pole plewy.

Metoda zostala stworzona aby uzyskac dostep i pozwolic na zmiane pola plewy przez klasy niezaprzyjaznione i niepochodne. Do pola przypisywana jest wartosc przekazana przez argument. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

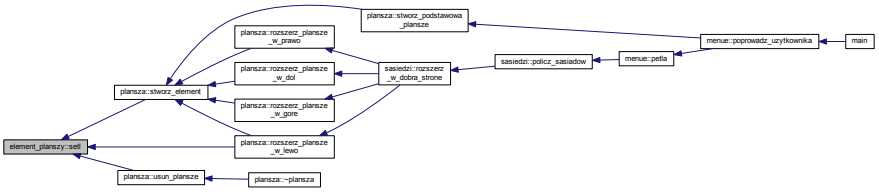
Parametry

pl	wskaznik na element po lewej.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.3.2.10 setp()

```
void element_planszy::setp (
            element_planszy * pp )
```

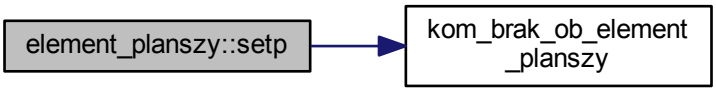
Metoda ustawia pole pprawy.

Metoda zostala stworzona aby uzyskac dostep i pozwolic na zmiane pola pprawy przez klasy niezaprzyjaznione i niepochodne. Do pola przypisywana jest wartosc przekazana przez argument. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy dane_planszy (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

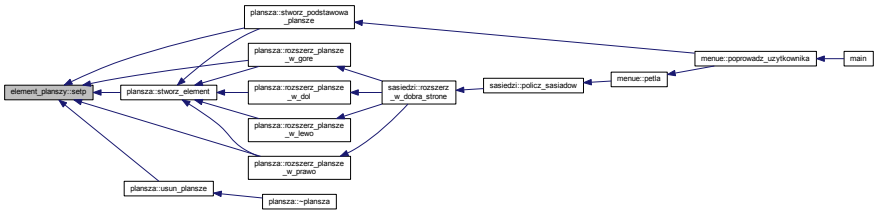
Parametry

pp	wskaznik na element po prawej.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

- klasa_element_planszy.h
- metody_element_planszy.cpp

4.4 Dokumentacja klasy komorka

```
#include <klasa_komorka.h>
```

Diagram dziedziczenia dla komorka

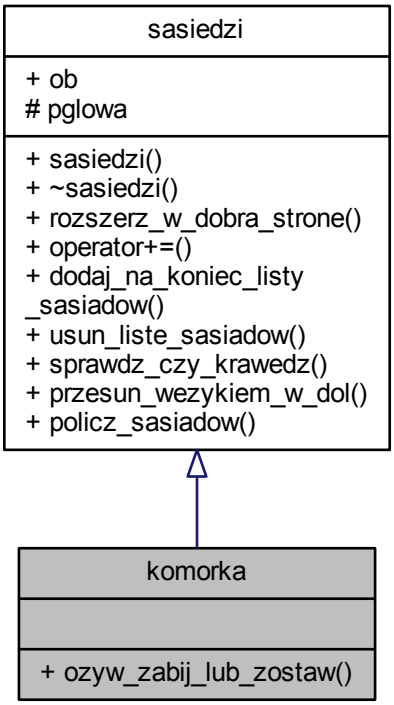
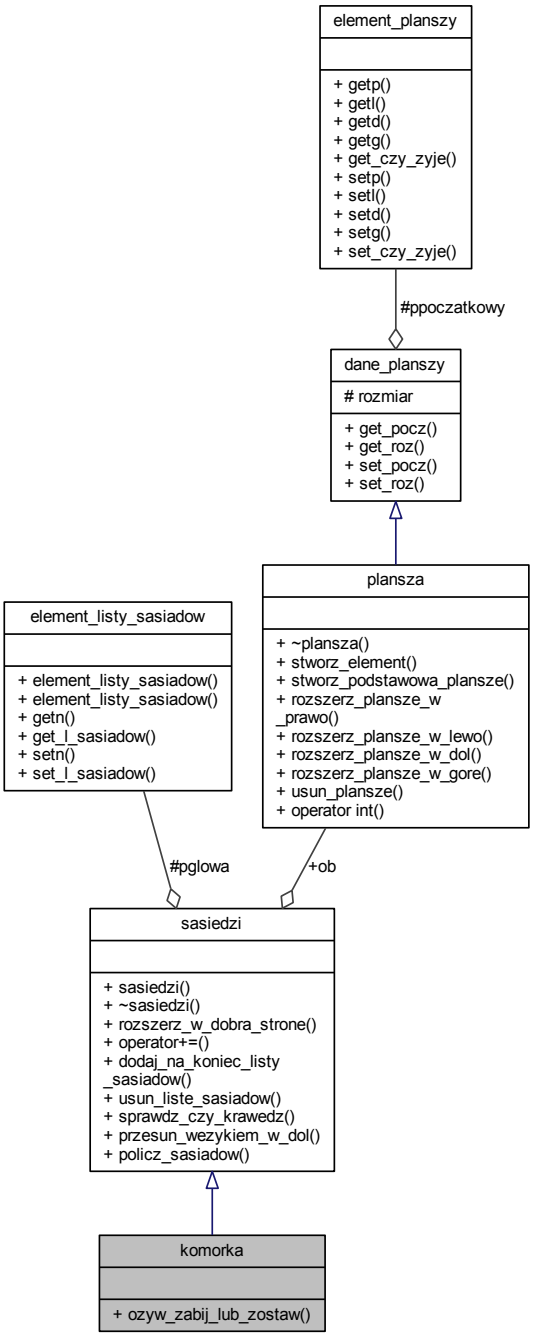


Diagram współpracy dla komorka:



Metody publiczne

- void ozyw_zabij_lub_zostaw ()

Dodatkowe Dziedziczone Składowe

4.4.1 Opis szczegółowy

Klasa sluzy do "ozywiania i zabijania komorek". Dziedziczy po klasie sasiedzi i dodaje tylko jedna metode. Nie zawiera dodatkowychy pol.

4.4.2 Dokumentacja funkcji składowych

4.4.2.1 ozyw_zabij_lub_zostaw()

```
void komorka::ozyw_zabij_lub_zostaw ( )
```

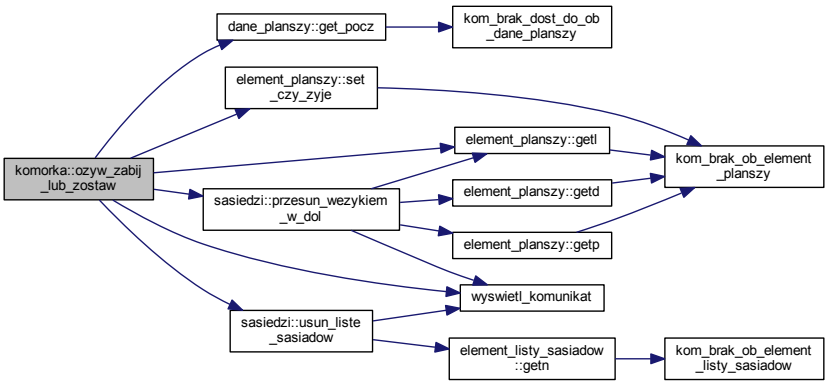
Metoda w zaleznosci od ilosci sasiadow komorki zabija ozywia lub zostawia w poprzednim stanie komorke.

Metoda uzywa dwoch wskaznikow pomocniczych. Pierwszy, tmp_lista jest uzyty do przechodzenia listy sasiadow i odczytywania z niej wartosci. Drugi, tmp jest uzyty do przechodzenia planszy "wezykiem w dol"(wywolana jest do tego metoda klasy sasiedzi przesun_wezykiem_w_dol) i sprawdzania oraz ustawiania stanu komorek w zaleznosci od ilosci sasiadow pod wskaznikiem tmp_lista. Metoda lapie ewentualne wyjatki wyrzucone przez uzyte w niej inne metody.

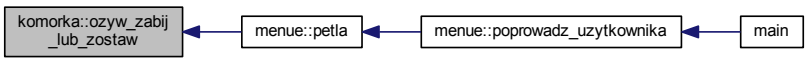
Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

- klasa_komorka.h
- metody_komorka.cpp

4.5 Dokumentacja klasy menue

`#include <klasa_menue.h>`

Diagram współpracy dla menue:

Statyczne metody publiczne

- static void kom_wpr_danych ()
- static void kom_wpr_danych_w_petli ()
- static void kom_s_czy_s ()
- static void kom_s_czy_s_w_petli ()
- static void kom_zapis ()
- static void kom_zapis_w_petli ()
- static void kom_stan_planszy ()
- static void kom_start ()
- static int wybor (int &zmiana_wyb)
- static void petla (komorka ko, wyswietl wys)
- static void zatrzymaj_petle ()
- static void poprowadz_uzytkownika ()

4.5.1 Opis szczegółowy

Klasa sluzy do laczenia wszystkich operacji, aby gra w zycie dzialala poprawnie oraz do komunikacji z uzytkownikiem.

4.5.2 Dokumentacja funkcji składowych

4.5.2.1 kom_s_czy_s()

```
void menue::kom_s_czy_s ( )  [static]
```

Metoda wyswietla komentarz na strumien wyjsciowy.

Wyswietlany komentarz prosi uzytkownika o wybor, czy uzytkownik chce rozpoczac gre w zycie z ozywionymi przez siebie komorkami, czy wylaczyc program i cos jeszcze zmienic.

Zwraca

nic

Oto graf wywoływań tej funkcji:

4.5.2.2 kom_s_czy_s_w_petli()

```
void menue::kom_s_czy_s_w_petli ( )  [static]
```

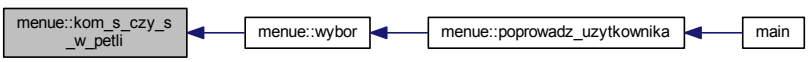
Metoda wyswietla komentarz na strumien wyjsciowy.

Wyswietlany komentarz dotyczy wyboru, czy uzytkownik chce wystartowac gre, czy zakonczyc program i cos zmienic. Jest on wyswietlany w petli, ktora konczy sie gdy uzytkownik poda poprawna odpowiedz (1 lub 2).

Zwraca

nic

Oto graf wywoływań tej funkcji:



4.5.2.3 kom_stan_planszy()

```
void menue::kom_stan_planszy ( )  [static]
```

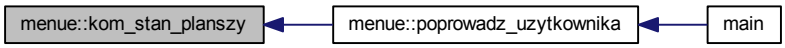
Metoda wyswietla komentarz na strumien wyjsciowy.

Wyswietlany komentarz informuje, ze zaraz zostanie wyswietlony aktualny stan planszy. Metoda jest uzyta przed wyswietleniem planszy w wersji zaraz po wprowadzeniu danych przez uzytkownika.

Zwraca

nic

Oto graf wywoływań tej funkcji:



4.5.2.4 kom_start()

```
void menue::kom_start ( )  [static]
```

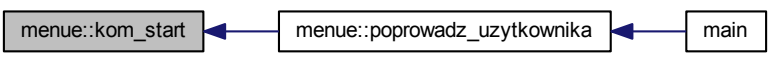
Metoda wyswietla komentarz na strumien wyjsciowy.

Wyswietlany komentarz informuje, ze zaraz rozpocznie sie gra w zycie oraz jak ja zakonczyc. Metoda jest uzyta tuz przed startem gry.

Zwraca

nic

Oto graf wywoływań tej funkcji:



4.5.2.5 kom_wpr_danych()

```
void menue::kom_wpr_danych ( )  [static]
```

Metoda wyswietla komentarz na strumien wyjsciowy.

Wyswietlany komentarz prosi uzytkownika o wybor sposobu wprowadzenia wspolrzednych komorek do ozywienia.

Zwraca

nic

Oto graf wywoływań tej funkcji:

4.5.2.6 kom_wpr_danych_w_petli()

```
void menue::kom_wpr_danych_w_petli ( )  [static]
```

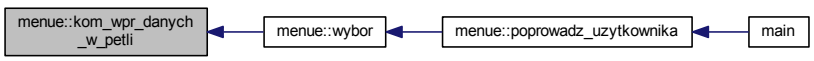
Metoda wyswietla komentarz na strumien wyjsciowy.

Wyswietlany komentarz dotyczy wyboru sposobu wprowadzenia wspolrzednych. Jest on wyswietlany w petli, ktora konczy sie gdy uzytkownik poda poprawna odpowiedz (1 lub 2).

Zwraca

nic

Oto graf wywoływań tej funkcji:



4.5.2.7 kom_zapis()

```
void menue::kom_zapis ( )  [static]
```

Metoda wyswietla komentarz na strumien wyjsciowy.

Wyswietlany komentarz prosi uzytkownika o wybor, czy chce on zapisac aktualny stan(po zatrzymaniu) planszy do pliku.

Zwraca

nic

Oto graf wywoływań tej funkcji:

4.5.2.8 kom_zapis_w_petli()

```
void menue::kom_zapis_w_petli ( )  [static]
```

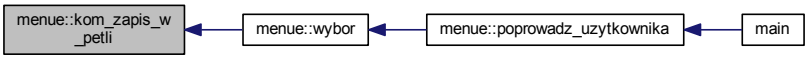
Metoda wyswietla komentarz na strumien wyjsciowy.

Wyswietlany komentarz dotyczy wyboru, czy uzytkownik chce zapisac plansze do pliku. Jest on wyswietlany w petli, ktora konczy sie gdy uzytkownik poda poprawna odpowiedz (1 lub 2).

Zwraca

nic

Oto graf wywoływań tej funkcji:



4.5.2.9 petla()

```
void menue::petla (
            komorka ko,
            wyswietl wys )  [static]
```

Metoda zawiera nieskonczona petle do gry w zycie.

W metodzie, w nieskonczonej petli, wywolywane sa wszelkie metody konieczne do zmiany stanu planszy(ozywianie i zabijanie) oraz do wyswietlenia jej. Z petli da sie wyjsc przez zmiane wartosci pola statycznego klasy menue warunek_wyjscia. Za zmiane tego pola odpowiedzialna jest zsynchronizowany watek metody zatrzymaj_petle. Obiekty podane jako argumenty musialy byc kopiowane, poniewaz tak jest domyslnie w watku, ktory przyjmuje funkcje i jej rgumenty. Zwykla referencja nic nie zmieniala. Niestety nie potrafie jeszcze dodac do watku argumentow bez kopiwania i dzialac na oryginale.

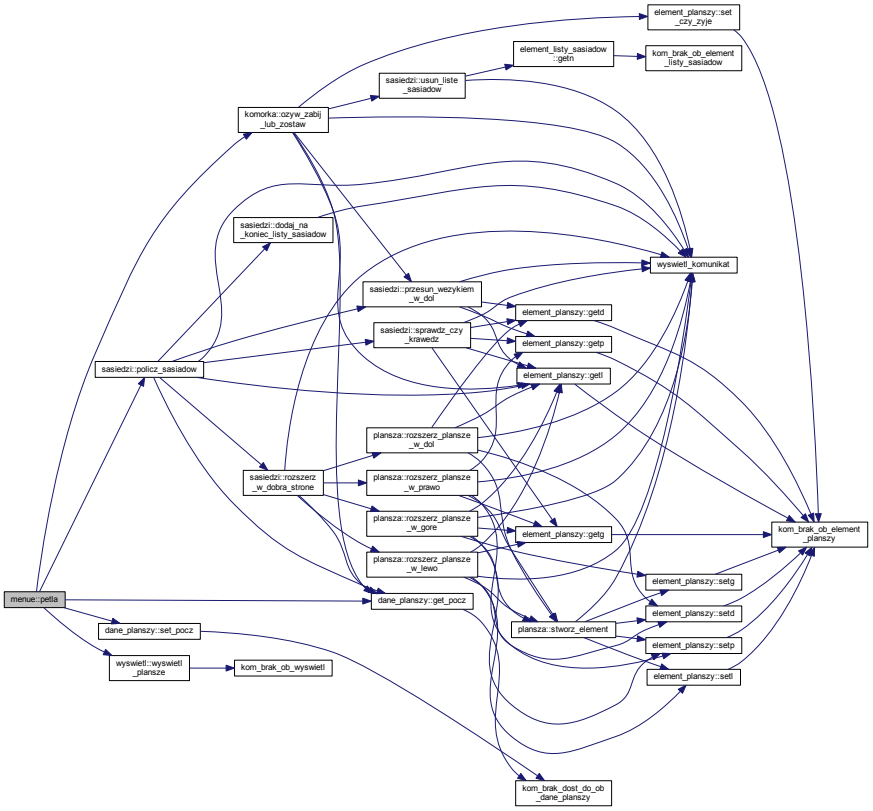
Parametry

ko	kopia obiektu klasy komorka, za pomoca metod tego obiektu beda ozywiane i zabijane komorki
wys	kopia obiektu klasy wyswietl, za pomoca metody tego obiektu bedzie wyswietlana plansza do gry w zycie i jej aktualny stan.

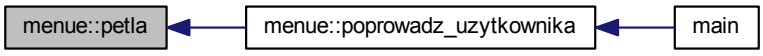
Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.5.2.10 poprowadz_uzytkownika()

```
void menue::poprowadz_uzytkownika ( )  [static]
```

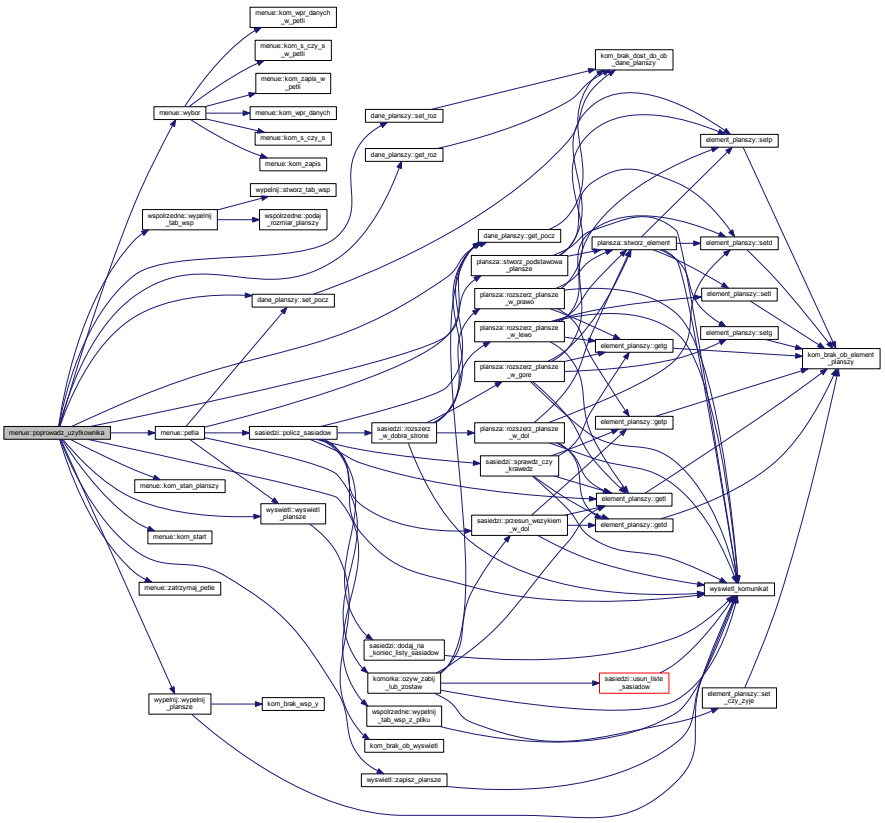
Glowna metoda w ktorej sa utworzone wszystkie obiekty oraz wywolane wszystkie metody konieczne do poprawnego dzialania gry w zycie

Kolejnosc dzialania: Uzytkownika czeka pierwszy wybor, czy chce wprowadzic wspolrzedne komorek do ozywienia z pliku czy wpisac je recznie. Po sczytaniu wspolrzednych (jednym ze sposobow) do tablicy dynamicznej utworzona zostaje postawowa plansza, a nastepnie wypelniona. Stan planszy jest wyswietlony. Jesli wszystko sie zgadza z oczekiwaniami, uzytkownik moze wystartowac gre w zycie. Jesli nie, moze wylaczyc program aby podac jeszcze raz inne wspolrzedne. Gdy uzytkownik wybral start gry, zostaje wyswietlony komunikat o sposobie zakonczenia gry. Gdy uzytkownik przerwie dzialanie programu (poprzez dzialanie opisane przy metodzie zatrzymaj_petle) pytany jest, czy chce zapisac aktualny stan planszy do pliku. Jesli tak, program pyta o nazwe nowego pliku do utworzenia. Program sie konczy. Jest to jedyna metoda wywolana w funkcji main.

Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

4.5.2.11 wybor()

```
int menue::wybor (
            int & zmiana_wyb )  [static]
```

Metoda do wyboru miedzy dwoma opcjami.

Metoda zaleznosci od parametru podanego jako argument wyswietla komunikaty zwiazane z roznym wyborem. 1 - wybor sposobu wprowadzenia wspolrzednych punktow do ozywienia 2 - wybor czy wystartowac gre w zycie, czy ja zakonczyc 3 - wybor czy zapisac aktualny stan planszy po zatrzymaniu nieskonczonej petli gry w zycie. Metoda nie konczy sie dopoki uzytkownik nie poda liczby 1 lub 2 na wejscie. Po zakonczeniu zwraca nr wyboru, ktory pozniej jest uzyty do wywolania odpowiednich metod.

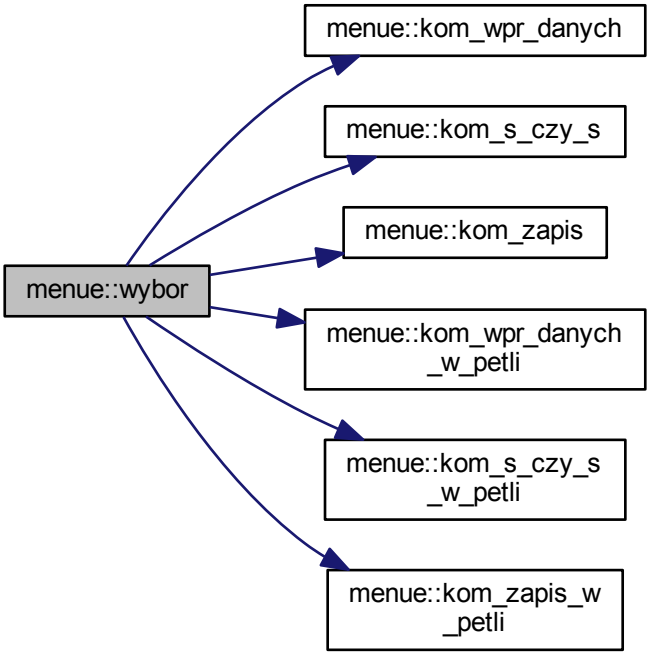
Parametry

zmiana_wyboru	jest indeksem wyboru, w zaleznosci od jego wartosci wyswietlaja sie inne opcje wyboru(1 - jak wprowadzic wspolrzedne, 2 - czy wystartowac gre w zycie, 3 - czy zapisac aktualny stan planszy).

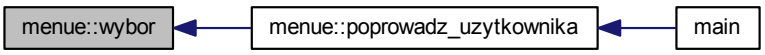
Zwraca

wybor - 48 (-48 , poniewaz wybrane opcje zostaja sczytane jako kod ascii)

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.5.2.12 zatrzymaj_petle()

```
void menue::zatrzymaj_petle ( )  [static]
```

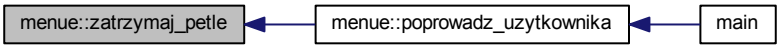
Metoda sluzy do zatrzymania nieskonczonej petli do gry w zycie.

Metoda czeka na podanie przez uzytkownika danych na wejscie. Jesli podana dana jest litera n, metoda zmienia wartosc pola klasy menue warunek_wyjscia na true. Nastepnie po sprawdzeniu tego warunku w nieskonczonej petli do gry w zycie petla sie zatrzymuje. Jesli uzytkownik podal cos innego, bufor sie czysci i metoda jest wywolana ponownie rekurencyjnie. Rekurencja byla konieczna, poniewaz inaczej metoda konczyla sie przed moetoda petla. Uniemozliwialo to wyjscie z nieskonczonej petli w sytuacji, gdy uzytkownik najpierw podal niepoprawna dana (cos co nie bylo litera n). Watek tej metody jest zsynchornizowany z watkiem metody petla.

Zwraca

nic

Oto graf wywoływań tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

- klasa_menue.h
- metody_menue.cpp

4.6 Dokumentacja klasy plansza

`#include <klasa_plansza.h>`

Diagram dziedziczenia dla plansza

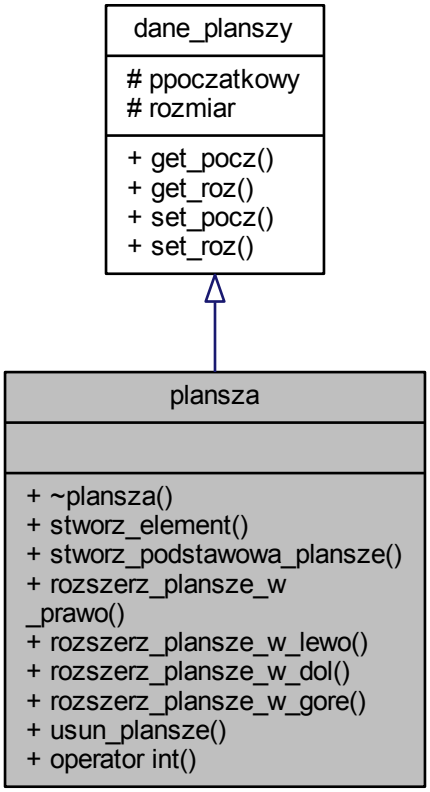
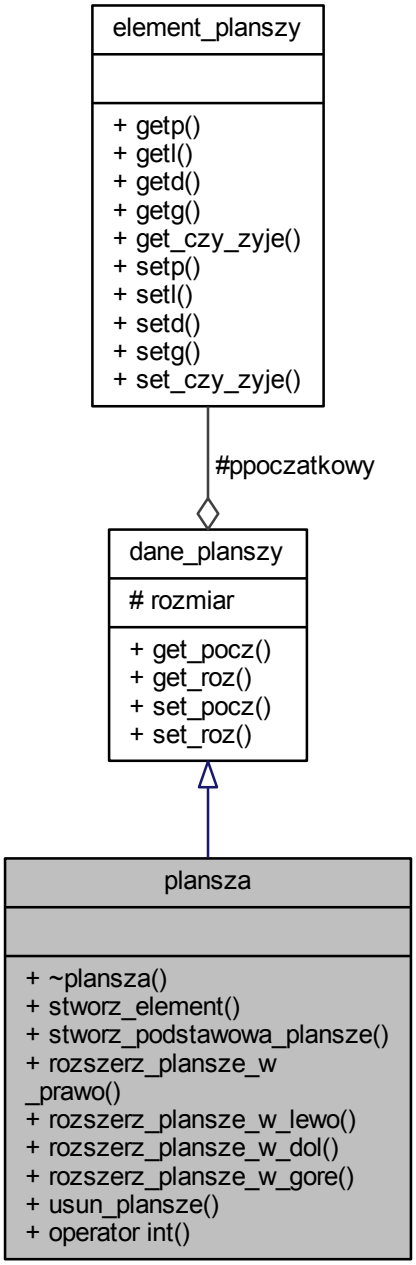


Diagram współpracy dla plansza:



Metody publiczne

- ~plansza ()
- element_planszy * stworz_element (element_planszy *pp, element_planszy *pl, element_planszy *pd, element_planszy *pg)
- void stworz_podstawowa_plansze ()
- void rozszerz_plansze_w_prawo (element_planszy *el)

- void rozszerz_plansze_w_lewo (element_planszy ∗el)
- void rozszerz_plansze_w_dol (element_planszy ∗el)
- void rozszerz_plansze_w_gore (element_planszy ∗el)
- void usun_plansze ()
- operator int ()

Dodatkowe Dziedziczone Składowe

4.6.1 Opis szczegółowy

Klasa umozliwia operacje na planszy. Jest klasa pochodna klasy dane_planszy.

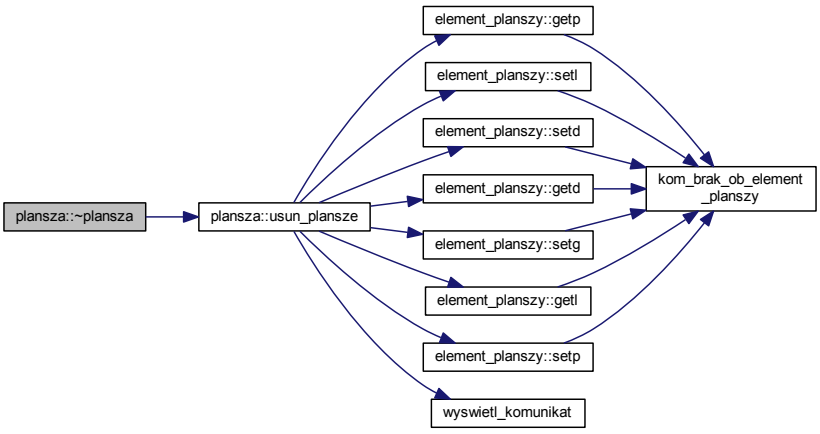
Klasa nie rozszerza klasy dane_planszy o zadne pole. Zawera jedynie dodatkowe metody, takie jak tworzenie elementu planszy, tworzenie z nich podstawowej planszy, rozszerzanie planszy w dana strone.

4.6.2 Dokumentacja konstruktora i destruktora

4.6.2.1 ∼plansza()

```
plansza::~plansza ( )
```

Destruktor klasy plansza. Poza usunieciem obiektu klasy plansza zwalnia pamiec zajmowana przez elementy planszy do gry w zycie. Jesli ppoczatkowy na nic nie wskazuje destruktor tylko usuwa obiekt. Oto graf wywołań dla tej funkcji:



4.6.3 Dokumentacja funkcji składowych

4.6.3.1 operator int()

```
plansza::operator int ( )
```

Przeciazony operator konwersji na int zlicza ilosc elementow planszy. Nic nie pobiera i zwraca ilosc elementow planszy.

Operator ustawia wskaznik chwilowy tmp na adres wskazywany przez wskaznik ppoczatkowy po czym przesuwa go w prawo zliczajac kolumny. Nastepnie przesuwa ten sam chwilowy wskaznik w dol i liczy wiersze. Poniewaz plansza ma zawsze ksztalt prostakatny wystarczy pomnozyc ilosc kolumn razy ilosc wierszy.

Zwraca

ilosc elementow planszy

4.6.3.2 rozszerz_plansze_w_dol()

```
void plansza::rozszerz_plansze_w_dol (
            element_planszy * el )
```

Metoda rozszerza plansze o wiersz w dol. Pobiera wartosc typu wskaznikowego na obiekt klasy element planszy i nic nie zwraca.

Metoda jest uzywana do rozszerzenia planszy w sytuacji gdy komorka "ozyje"(pole bool czy_zyje klasy element_planszy = 1) na dolnej krawedzi. Dzieki niej plansza jest nieskonczona z dolu (ograniczona pamiecia komputera). Jesli cos poszlo nie tak i wskaznik el wskazuje na el w srodku planszy, metoda nic nie robi.

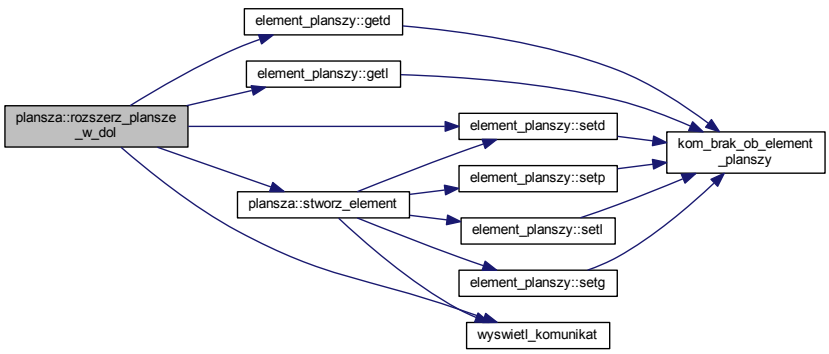
Parametry

el	wskaznik na komorke na dolnej krawedzi.

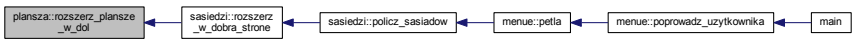
Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.6.3.3 rozszerz_plansze_w_gore()

```
void plansza::rozszerz_plansze_w_gore (
            element_planszy * el )
```

Metoda rozszerza plansze o wiersz w gore. Pobiera wartosc typu wskaznikowego na obiekt klasy element planszy i nic nie zwraca.

Metoda jest uzywana do rozszerzenia planszy w sytuacji gdy komorka "ozyje"(pole bool czy_zyje klasy element_planszy = 1) na gornej krawedzi. Dzieki niej plansza jest nieskonczona z gory (ograniczona pamiecia komputera). Jesli cos poszlo nie tak i wskaznik el wskazuje na el w srodku planszy, metoda nic nie robi. Metoda przesuwa wskaznik ppoczatkowy z klasy dane_planszy na pierwszy element dodawanej kolumny(element w lewym gornym rogu).

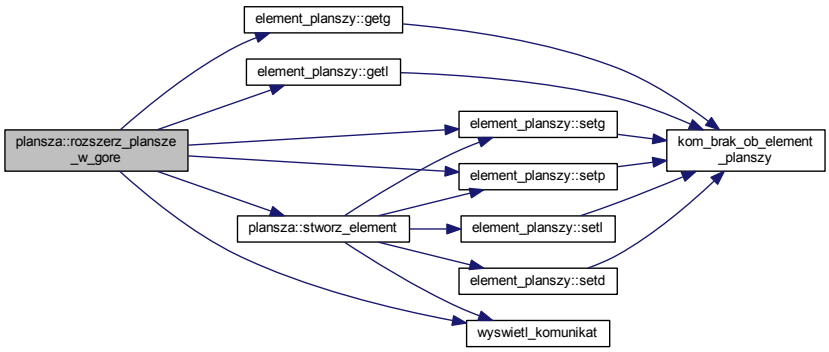
Parametry

el	wskaznik na komorke na gornej krawedzi.

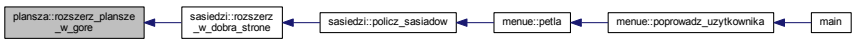
Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.6.3.4 rozszerz_plansze_w_lewo()

```
void plansza::rozszerz_plansze_w_lewo (
            element_planszy * el )
```

Metoda rozszerza plansze o kolumne w lewo. Pobiera wartosc typu wskaznikowego na obiekt klasy element planszy i nic nie zwraca.

Metoda jest uzywana do rozszerzenia planszy w sytuacji gdy komorka "ozyje"(pole bool czy_zyje klasy element_planszy = 1) na lewej krawedzi. Dzieki niej plansza jest nieskonczona z lewej strony (ograniczona pamiecia komputera). Jesli cos poszlo nie tak i wskaznik el wskazuje na el w srodku planszy, metoda nic nie robi. Metoda przesuwa wskaznik ppoczatkowy z klasy dane_planszy na pierwszy element dodawanej kolumny(element w lewym gornym rogu).

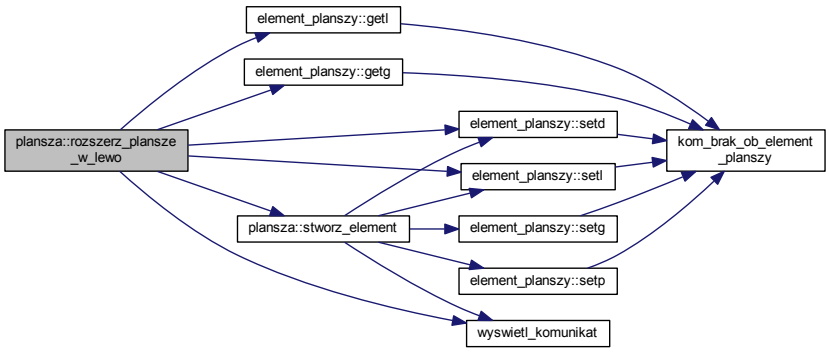
Parametry

el	wskaznik na komorke na lewej krawedzi.

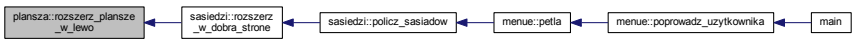
Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.6.3.5 rozszerz_plansze_w_prawo()

```
void plansza::rozszerz_plansze_w_prawo (
            element_planszy * el )
```

Metoda rozszerza plansze o kolumne w prawo. Pobiera wartosc typu wskaznikowego na obiekt klasy element planszy i nic nie zwraca.

Metoda jest uzywana do rozszerzenia planszy w sytuacji gdy komorka "ozyje"(pole bool czy_zyje klasy element_planszy = 1) na prawej krawedzi. Dzieki niej plansza jest nieskonczona z prawej strony (ograniczona pamiecia komputera). Jesli cos poszlo nie tak i wskaznik el wskazuje na el w srodku planszy, funkcja nic nie robi.

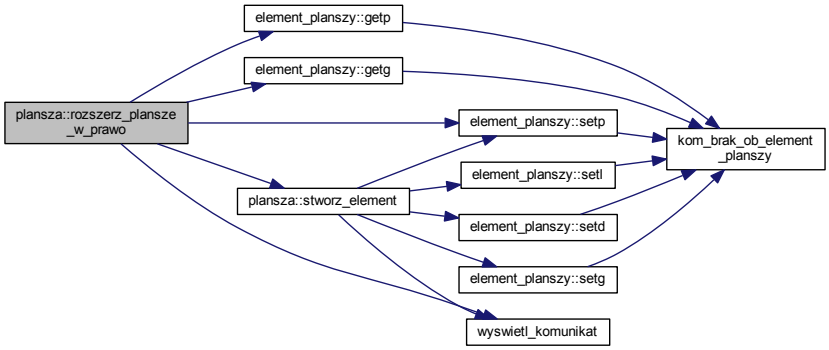
Parametry

el	wskaznik na komorke na prawej krawedzi.

Zwraca

nowy - wskaznik na nowo utworzony obiekt klasy element_planszy.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

4.6.3.6 stworz_element()

```
element_planszy * plansza::stworz_element (
            element_planszy * pp,
            element_planszy * pl,
            element_planszy * pd,
            element_planszy * pg )
```

Metoda tworzy element planszy, ktory jest reprezentacja komorki z gry w zycie Pobiera cztery wartosci typu wskaznikowego na obiekt klasy element planszy i zwraca rowniez wskaznik na obiekt klasy element_planszy.

W metodzie utworzony jest obiekt klasy element_planszy i ustawiony na niego wskaznik nowy. Nastepnie ustawiane sa wszystkie wskazniki "sasiadow" na elementy podane w parametrach funkcji.

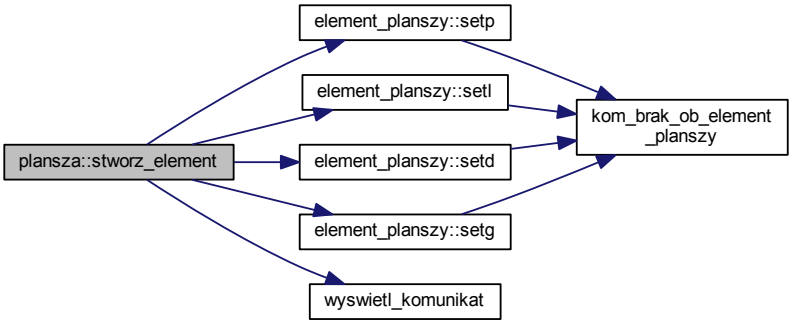
Parametry

pp	wskaznik na komorke sasiadujaca po prawej lub na nullptr gdy tworzony element znajduje sie na prawej krawedzi planszy
pl	wskaznik na komorke sasiadujaca po lewej lub na nullptr gdy tworzony element znajduje sie na lewej krawedzi planszy
pd	wskaznik na komorke sasiadujaca z dolu lub na nullptr gdy tworzony element znajduje sie na dolnej krawedzi planszy
pg	wskaznik na komorke sasiadujaca z gory lub na nullptr gdy tworzony element znajduje sie na gornej krawedzi planszy

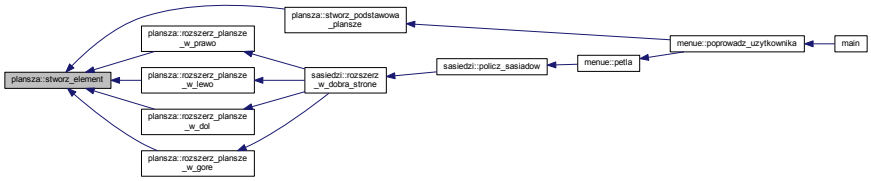
Zwraca

nowy - wskaznik na nowo utworzony obiekt klasy element_planszy.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.6.3.7 stworz_podstawowa_plansze()

```
void plansza::stworz_podstawowa_plansze ( )
```

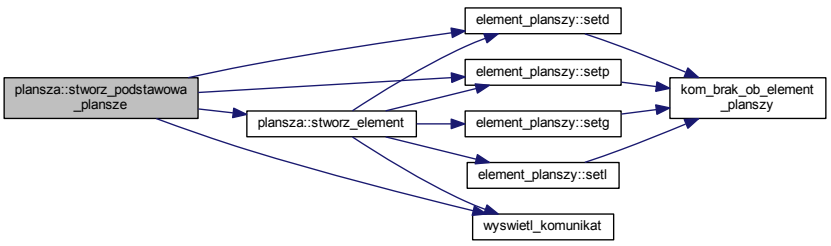
Metoda tworzy plansze o boku rownym polu klasy dane_planszy - rozmiar , na ktorej bedzie odbywala sie gra w zycie. Nie pobiera zadnych parametrow i nic nie zwraca.

Gdy pole rozmiar jest rowne 0 plansza w ogole sie nie tworzy. Jesli nie, funkcja najpierw tworzy glowna kolumne o dlugosci rozmiar, nastepnie tworzy kolejne kolumny, ktorych jest rowniez rozmiar. W metodzie wykozystane sa trzy wskazniki chwilowe: tmp_w_dol - do przesuwania sie w dol po utworzonej lewej kolumnie i laczenia elementow wskaznikami pprawy z nowo tworzonymi. tmp_prawy_w_dol - do przesuwania sie w dol i tworzenia nowej kolumny oraz laczenia jej z utworzona po lewej za pomoca wskaznikow plewy. tmp_w_prawo - do przesuniecia powyzszych wskaznikow w prawo tak aby tmp_w_dol wskazywal na pierwszy el istniejacej kolumny, a tmp_prawy_w_dol na pierwszy nowo utworzony w nastepnej.

Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

4.6.3.8 usun_plansze()

```
void plansza::usun_plansze ( )
```

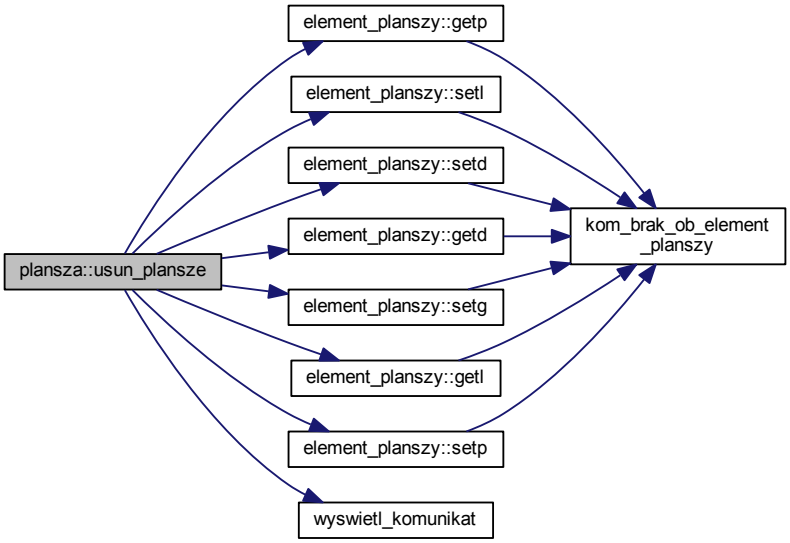
Metoda usuwa plansze do gry w zycie. Nic nie pobiera i nic nie zwraca.

Metoda ustawia wskaznik chwilowy tmp na adres wskazywany przez wskaznik ppoczatkowy po czym przesuwa wskaznik ppoczatkowy w prawo. Nastepnie usuwa komorke wskazywana przez tmp oraz ustawia wskaznik plewy komorki wskazywanej przez ppoczatkowy na nullptr. Kiedy wskaznik ppoczatkowy "dojdzie" do prawej krawedzi planszy, jesli to mozliwe, przesuwa sie w dol i wykonywana jest ten sam algorytm na nastepnym wierszy, tylko w lewo. Ostatni usuniety element ma wszystkie wskazniki "sasiadow" ustawione na nullptr poza pgorny. Taka sytuacja spelnia instrukcje warunkowa else if i ustawia wskaznik ppoczatkowy na nullptr i pozwala na wyjscie z petli while. Metoda zostala stworzona aby zapobiedz wyciekowi pamieci. Jest uzyta w destruktorze, dzieki czemu przy usuwaniu obiektu klasy plansza zostaje zwolniona pamiec zajmowana przez wszystkie elementy planszy.

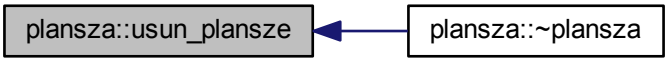
Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

- klasa_plansza.h
- metody_plansza.cpp

4.7 Dokumentacja klasy sasiedzi

```
#include <klasa_sasiedzi.h>
```

Diagram dziedziczenia dla sasiedzi

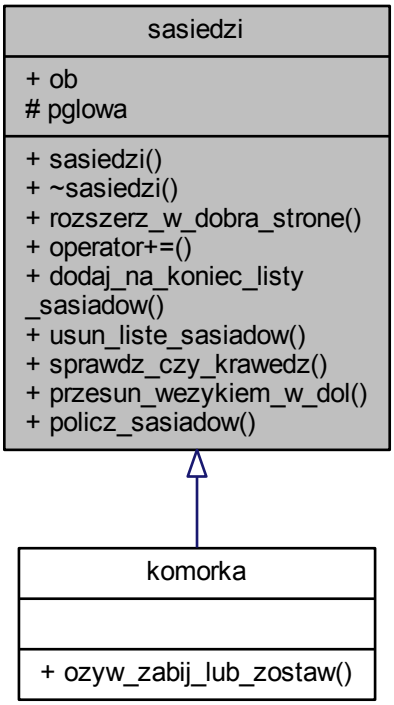
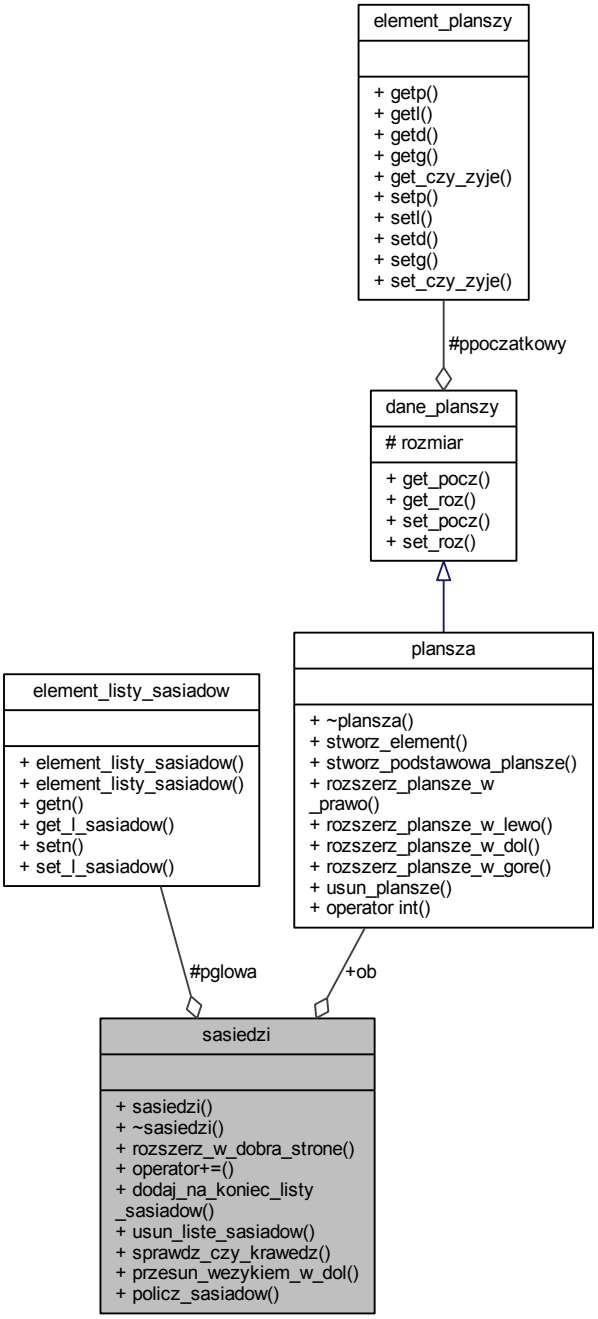


Diagram współpracy dla sasiedzi:



Metody publiczne

- sasiedzi ()
- ~sasiedzi ()
- void rozszerz_w_dobra_strone ()
- element_listy_sasiadow * operator+= (int l_s)
- void dodaj_na_koniec_listy_sasiadow (int l_s)

- void usun_liste_sasiadow ()
- int sprawdz_czy_krawedz (element_planszy ∗el)
- void przesun_wezykiem_w_dol (element_planszy ∗&el, element_planszy ∗&el_za)
- void policz_sasiadow ()

Atrybuty publiczne

- plansza ∗ ob

Atrybuty chronione

- element_listy_sasiadow ∗ pglowa

4.7.1 Opis szczegółowy

Klasa sluzy do zliczania sasiadow komore planszy do gry w zycie. Jest klasa bazowa klasy komorka

4.7.2 Dokumentacja konstruktora i destruktora

4.7.2.1 sasiedzi()

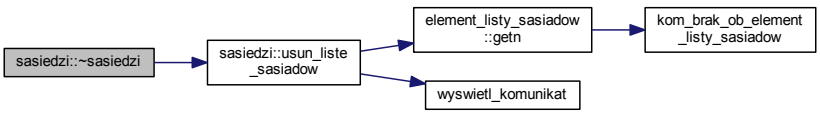
```
sasiedzi::sasiedzi ( )
```

Konstruktor klasy sasiedzi. Utworzony, aby ustawiac pole pglowa na nullptr przy tworzeniu nowego obiektu.

4.7.2.2 ∼sasiedzi()

```
sasiedzi::∼sasiedzi ( )
```

Destruktor klasy sasiedzi. Utworzony, aby dodatkwo zwalniac pamiec zajmowana przez liste sasiadow (wywolana funkcja usun_liste_sasiadow). Jesli pole pglowa na cos wskazuje, zostaje to usuniete. Oto graf wywołań dla tej funkcji:



4.7.3 Dokumentacja funkcji składowych

4.7.3.1 dodaj_na_koniec_listy_sasiadow()

```
void sasiedzi::dodaj_na_koniec_listy_sasiadow (
            int l_s )
```

Metoda dodaje nowy element na koniec listy sasiadow.

W metodzie uzyty jest przeciazony operator arytmetyczny +=.

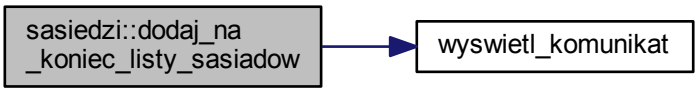
Parametry

l_s	liczba sasiadow komorki, ktora ma byc dodana do listy

Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

4.7.3.2 operator+=()

```
element_listy_sasiadow * sasiedzi::operator+= (
            int l_s )
```

Przeciazony operator arytmetyczny +=, ktory sluzy do dodawania elementow na koniec listy sasiadow.

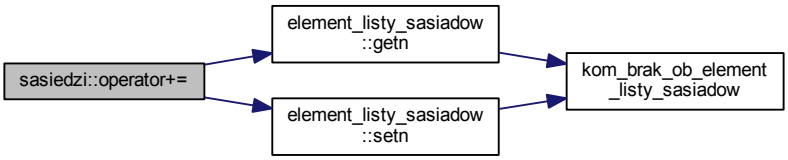
Parametry

l_s	liczba sasiadow komorki, ktora ma byc dodana do listy

Zwraca

nowy przeksztalcony wskaznik na pierwszy element listy sasiadow.

Oto graf wywołań dla tej funkcji:



4.7.3.3 policz_sasiadow()

```
void sasiedzi::policz_sasiadow ( )
```

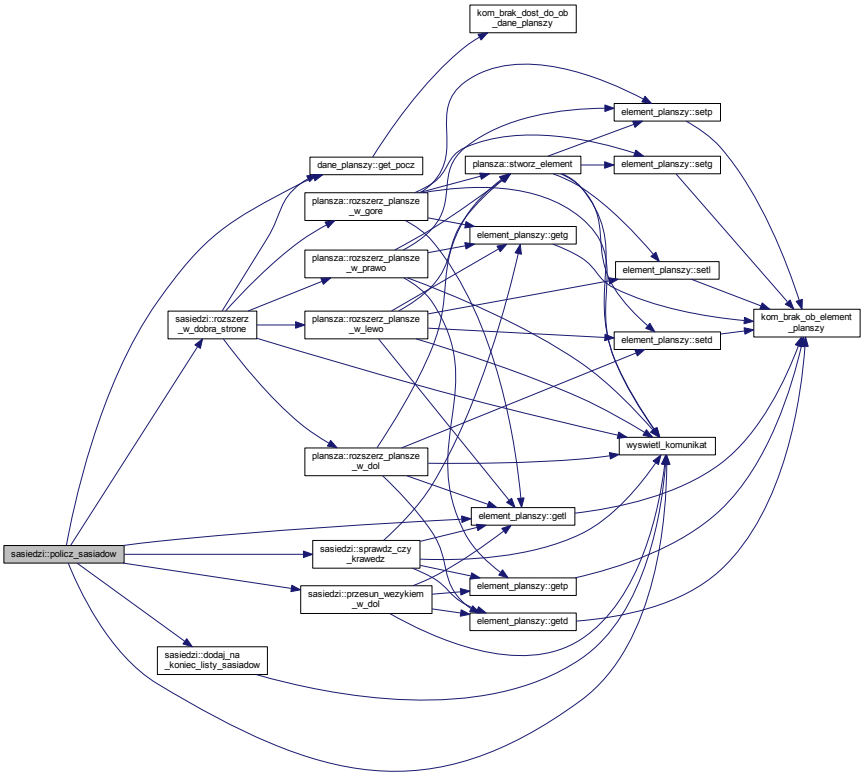
Metoda liczy sasiadow wszystkich komorek w planszy i dodaje ich liczbe na koniec listy sasiadow.

W metodzie wywolane sa prawie wszytkie metody klasy sasiedzi. Na poczatku plansza jest rozszerzona w odpowiednie strony (metoda rozszerz_w_dobra_strone), aby moc sprawdzic ilosc sasiadow komorek, ktore wczesniej nie istnialy. Zainicjowane sa wskazniki pomocnicze do przechodzenia planszy i liczenia sasiadow. W petli najpierw sprawdzana jest pozycja komorki(metoda sprawdz_czy_krawedz). Nastepnie w zaleznosci od wartosci zwroconej przez metode sprawdz_czy_krawedz(czytaj opis tej metody) sprawdzana jest ilosc sasiadow. Ilosc ta jest dodana na koniec listy sasiadow(metoda dodaj_na_koniec_listy_sasiadow). operacja powtarzana az zostana zliczeni sasiedzi wszystkich komorek rozszerzonej planszy.

Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

4.7.3.4 przesun_wezykiem_w_dol()

```
void sasiedzi::przesun_wezykiem_w_dol (
            element_planszy *& el,
            element_planszy *& el_za )
```

Metoda przesuwa wskaznik po planszy "wezykiem w dol".

Metoda przesuwa wskaznik podany jako parametr el o jedna pozycje. Najpierw do konca w prawo, jak juz sie nie da, raz w dol, potem w lewo. Przesuniecie jest mozliwe dopoki wskaznik na dolnego sasiada wskazuje na cos.

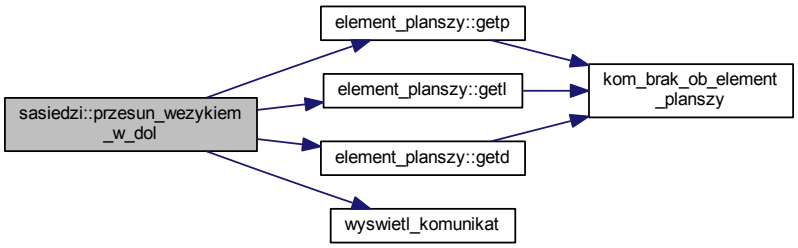
Parametry

el	wskaznik na element planszy (do przesuniecia)
el_za	wskaznik na element za el w sensie przechodzenia wezykiem w dol(wskazuje na sasiada po lewej, jak metoda przesuwa w prawo, odwrotnie gdy w lewo)

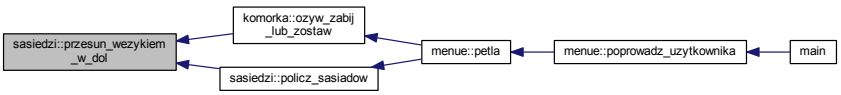
Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.7.3.5 rozszerz_w_dobra_strone()

```
void sasiedzi::rozszerz_w_dobra_strone ( )
```

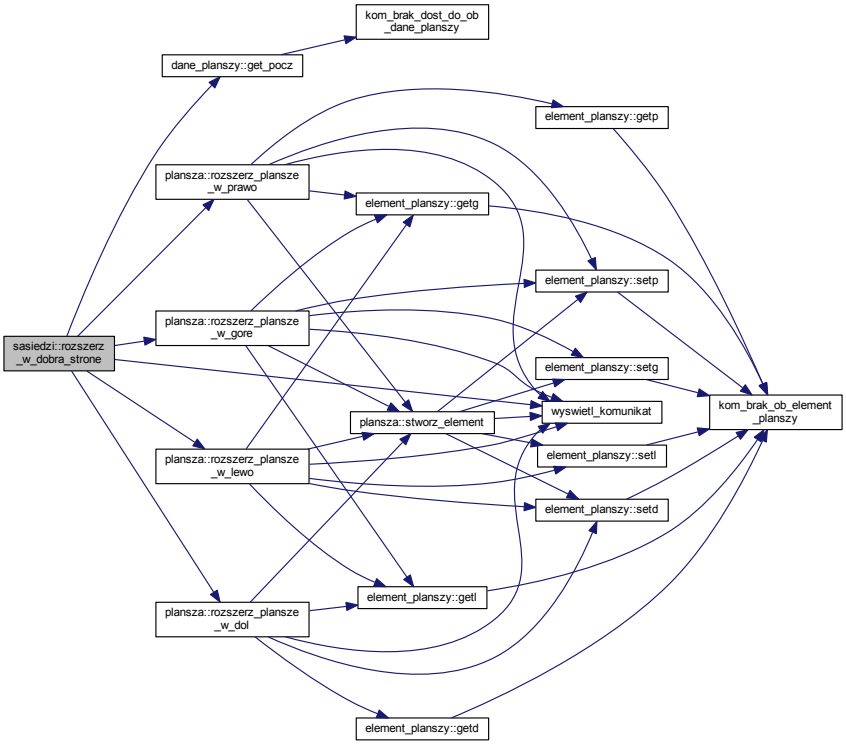
Metoda rozszerza plansze do gry w zycie w odpowiedna strone

Metoda uzywa pomocniczych wskaznikow i przesuwa jeden z nich wokol planszy sprawdzajac, czy na krawedzi jest jakas zywa komorka. Jesli tak, plansza jest rozszerzana(wywolana jest odpowiednia metoda klasy plansza) w strone krawedzi przy ktorej znaleziona zostala zywa komorka.

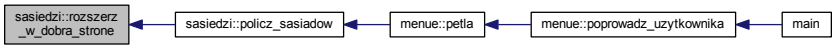
Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.7.3.6 sprawdz_czy_krawedz()

```
int sasiedzi::sprawdz_czy_krawedz (
            element_planszy * el )
```

Metoda sprawdza polozenie komorki i w zaleznosci od niego zwraca odpowiednia liczbe.

Znaczenie zwracanej liczby, to lokalizacja komorki w: 0 - prawa krawedz 10 - prawy dolny rog 20 - prawy gorny rog 1 - lewa krawedz 11 - lewy dolny rog 21 - lewy gorny rog 2 - dolna krawedz 3 - gorna krawedz Dzieki tej metodzie metoda policz_sasiadow nie sprawdza sasiadow, ktorzy nie maja prawa byc zywymi.

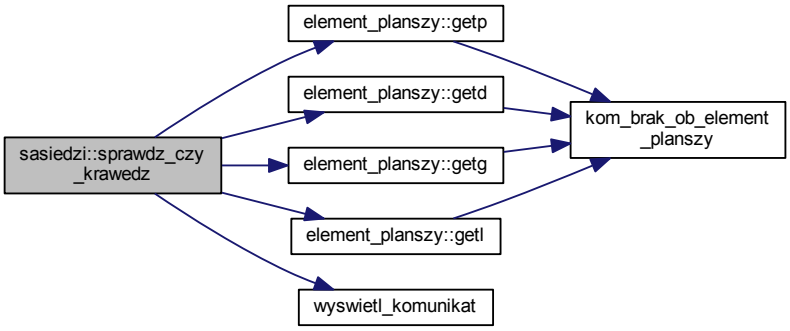
Parametry

el	wskaznik na komorke, ktorej lokalizacje chcemy poznac.

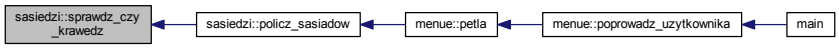
Zwraca

liczba odpowiadajaca lokalizacji komorki.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.7.3.7 usun_liste_sasiadow()

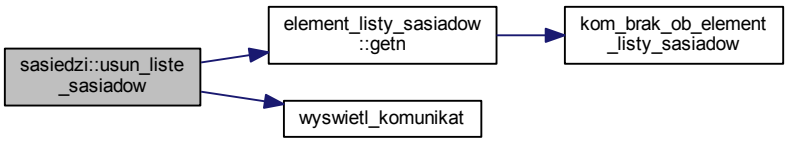
```
void sasiedzi::usun_liste_sasiadow ( )
```

Metoda usuwa i zwalnia pamiec po liscie sasiadow. Metoda wywolana w destruktorze.

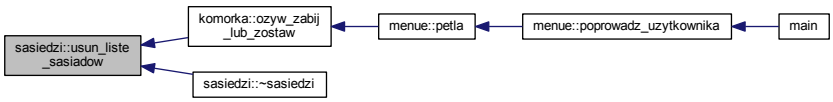
Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.7.4 Dokumentacja atrybutów składowych

4.7.4.1 ob

```
plansza* sasiedzi::ob
```

Pole ob jest wskaznikiem na obiekt klasy plansza. Takie rozwiazanie zostalo zastosowane zamiast dziedziczenia po klasie dane_planszy, aby nie tworzyc tymczasowego obiektu klasy plansza do wywolania metod do rozszerzania planszy.

4.7.4.2 pglowa

```
element_listy_sasiadow* sasiedzi::pglowa  [protected]
```

Pole pglowa jest wskaznikiem na pierwszy element listy sasiadow.

Dokumentacja dla tej klasy została wygenerowana z plików:

- klasa_sasiedzi.h
- metody_sasiedzi.cpp

4.8 Dokumentacja klasy wspolrzedne

```
#include <klasa_wspolrzedne.h>
```

Diagram dziedziczenia dla wspolrzedne

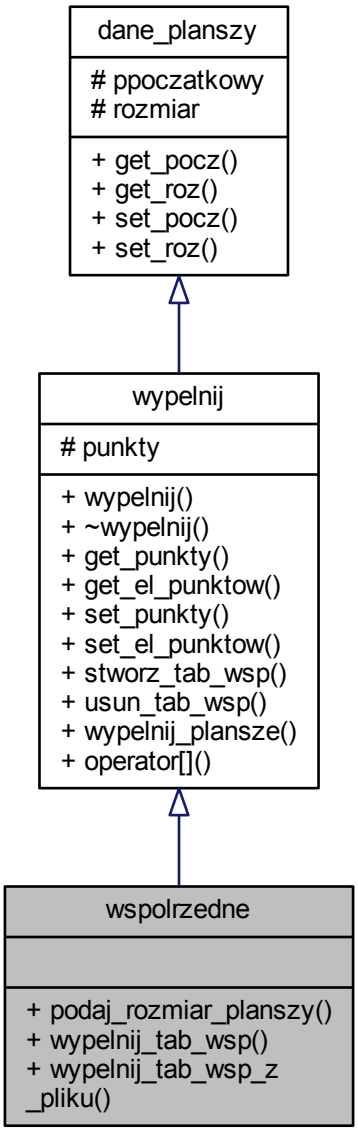
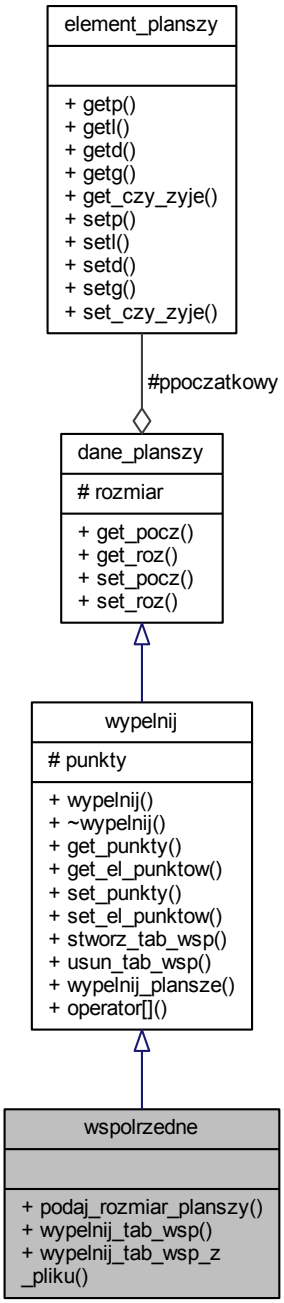


Diagram współpracy dla wspolrzedne:



Metody publiczne

- void podaj_rozmiar_planszy ()
- void wypelnij_tab_wsp ()
- void wypelnij_tab_wsp_z_pliku ()

Dodatkowe Dziedziczone Składowe

4.8.1 Opis szczegółowy

Klasa sluzy do sczytywania wspolrzednych komorek do ozywienia(na planszy do gry w zycie) do tablicy dynamicznej punkty(pole klasy wypelnij). Jest klasa pochodna klasy wypelnij(ktora jest klasa pochodna klasy dane_planszy).

Klasa zostala stworzona aby oddzielic operacje wczytania i wyswietlanie komunikatow zwiazanych z tym od logiki metod klasy sasiedzi. Metody tej klasy zawieraja rowniez operacje zapobiegajace wpisaniu niepoprawnych danych. Wykozystana jest maszyna stanow. Zawera dodatkowo typ wyliczeniowy oraz metody.

4.8.2 Dokumentacja funkcji składowych

4.8.2.1 podaj_rozmiar_planszy()

```
void wspolrzedne::podaj_rozmiar_planszy ( )
```

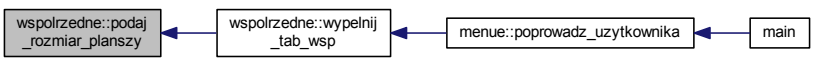
Metoda sluzy do wprowadzenia rozmiaru kwardatowej planszy poczatkowej(dlugosc boku).

Wyswietlany jest komunikat, ktory prosi uzytkonika o podanie rozmiaru planszy , tak aby zmiescily sie wszystkie komorki do ozywienia. Nastepnie uzytkownik powinien podac ten rozmiar. Operacja sie powtarza dopoki uzytkownik nie poda liczby naturalnej. Kazdorazowo przy wpisaniu niepoprawwnych danych wyswietlany jest odpowiedni komunikat.

Zwraca

nic

Oto graf wywoływań tej funkcji:



4.8.2.2 wypelnij_tab_wsp()

```
void wspolrzedne::wypelnij_tab_wsp ( )
```

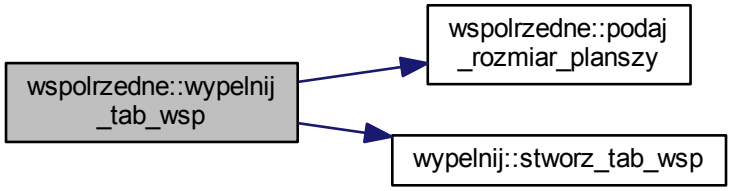
Metoda sluzy do wypelnienia tablicy dynamicznej punkty(pole klasy wypelnij) przez uzytkownika(recznie).

Najpierw sczytany jest rozmiar kwadratowej planszy poczatkowej(metoda podaj_rozmiar_planszy). Tworzona jest tablica wspolrzednych punkty(metoda klasy wypelnij stworz_tab_wsp) o rozmiarze rozmiar $*$ rozmiar $*$ 2 (jest to ilosc komorek na planszy $*$ 2(dwie wspolrzedne)). Wyswitlony jest komunikat jak zakonczyc wpisywanie wspolrzednych komorek do ozywienia. W petli wykorzystana jest maszyna stanow do pobrania od uzytkownika wspolrzednych. Wyswietlany komentarz prosi uzytkownika o wybor sposobu wprowadzenia wspolrzednych komorek do ozywienia.

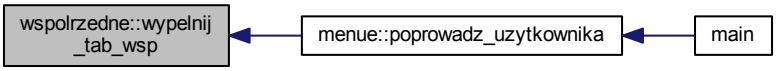
Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.8.2.3 wypelnij_tab_wsp_z_pliku()

```
void wspolrzedne::wypelnij_tab_wsp_z_pliku ( )
```

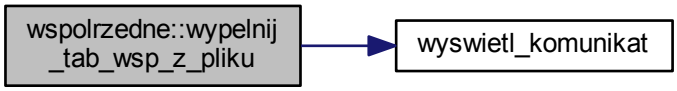
Metoda sluzy do wypelnienia tablicy dynamicznej punkty(pole klasy wypelnij) z pliku o podanej nazwie.

Najpierw zostaje wyswietlony komunikat proszacy o podanie nazwy pliku i wyjasniajacy jak beda interpretowane dane w pliku. Po podaniu nazwy otwieranyn jest plik. Wykorzystany przeciazony operator strumieniowy $>>$. Jezeli plik nie istnieje, wyrzucany jest wyjatek przez ten operator. Jesli istnieje, dane zostaja wczytane z pliku w podany w komunikacie sposob.

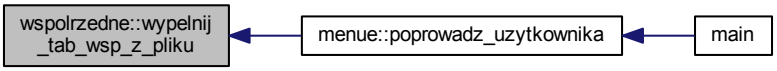
Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

- klasa_wspolrzedne.h
- metody_wspolrzedne.cpp

4.9 Dokumentacja klasy wypelnij

```
#include <klasa_wypelnij.h>
```

Diagram dziedziczenia dla wypelnij

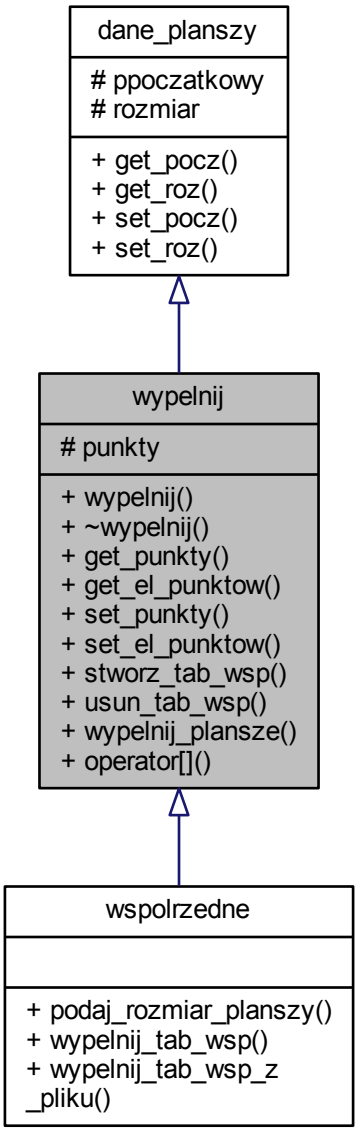
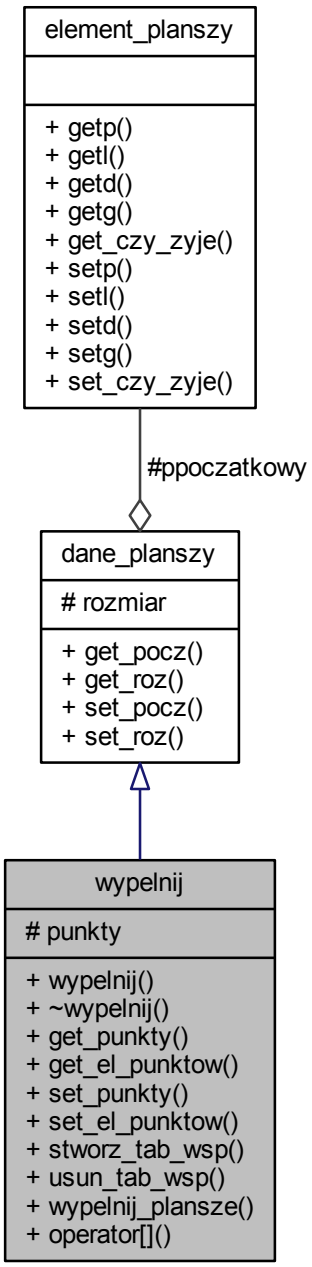


Diagram współpracy dla wypelnij:



Metody publiczne

- wypelnij ()
- ~wypelnij ()
- int * get_punkty ()
- int get_el_punktow (int idx)
- void set_punkty (int *p)

- void set_el_punktow (int idx, int liczba)
- void stworz_tab_wsp ()
- void usun_tab_wsp ()
- void wypelnij_plansze ()
- element_planszy * operator[ ] (size_t el)

Atrybuty chronione

- int * punkty

4.9.1 Opis szczegółowy

Klasa sluzy do wypelniania planszy do gry w zycie zywymi komorkami. Jest klasa bazowa klasy wspolrzedne.

4.9.2 Dokumentacja konstruktora i destruktora

4.9.2.1 wypelnij()

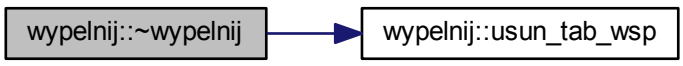
```
wypelnij::wypelnij ( )
```

Konstruktor klasy wypelnij. Utworzony, aby ustawiac pole punkty na nullptr przy tworzeniu nowego obiektu.

4.9.2.2 ~wypelnij()

```
wypelnij::~wypelnij ( )
```

Destruktor klasy wypelnij. Utworzony, aby dodatkwo zwalniac pamiec zajmowana przez tablice dynamiczna punkty (wywolana funkcja usun_tab_wsp). Jesli pole punkty na cos wskazuje, zostaje to usuniete. Oto graf wywołań dla tej funkcji:



4.9.3 Dokumentacja funkcji składowych

4.9.3.1 get_el_punktow()

```
int wypelnij::get_el_punktow (
            int idx )
```

Metoda zwraca element tablicy dynamicznej punkty o podanym w argumencie indeksie.

Metoda zostala stworzona aby uzyskac dostep do jednego elementu pola punkty (wskazaego przez argument) przez klasy niezaprzyjaznione i niepochodne. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy wypelnij (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

Parametry

idx	indeks elementu tablicy dynamicznej do zwrocenia

Zwraca

punkty, jesli jest dotep do obiektu klasy wypelnij. Jesli nie, zwraca -1.

Oto graf wywołań dla tej funkcji:

4.9.3.2 get_punkty()

```
int * wypelnij::get_punkty ( )
```

Metoda zwraca pole punkty.

Metoda zostala stworzona aby uzyskac dostep do pola punkty przez klasy niezaprzyjaznione i niepochodne. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy wypelnij (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

Zwraca

punkty, jesli jest dotep do obiektu klasy dane_planszy. Jesli nie, zwraca nullptr.

Oto graf wywołań dla tej funkcji:

4.9.3.3 operator[]()

```
element_planszy * wypelnij::operator[] (
            size_t el )
```

Przeciazony operator tablicowy [], ktory ustawia pomocniczy wskaznik na elemencie o podanym w argumencie indeksie i zwraca go.

Operator ustawia wskaznik chwilowy tmp na adres elementu planszy do gry w zycie o podanym przez argument indeksie. Indeks jest interpretowany w nastepujacy sposob: Poniewaz metoda przesuwa pomocniczy wskaznik kolejno po wierszach, indeks oznacza liczbe przesuniec konieczna do ustawienia tego wskaznika. Indeks rowny jest pierwszej wspolrzednej x + rozmiar * druga wspolrzedna y. Jesli okaze sie, ze wspolrzedna w tablicy punkty wskazywala na element poza plansza zostaje wyrzucony wyjatek.

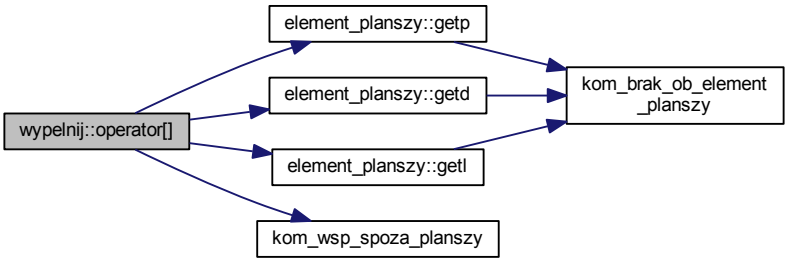
Parametry

el	indeks komorki do ozywienia (ilosc koniecznych przesuniec od elementu poczatkowego, aby dostac sie do pozodanej komorki)

Zwraca

adres na ktory ustawiony jest pomocniczy wskaznik, czyli elementu o podanym w argumencie indeksie.

Oto graf wywołań dla tej funkcji:



4.9.3.4 set_el_punktow()

```
void wypelnij::set_el_punktow (
            int idx,
            int liczba )
```

Metoda ustawia wartosc jednego elementu tablicy dynamicznej punkty. Wartosc podana jako drugi argument wpisana jest do elementu tablicy punkty o podanym indeksie w pierwszym argumencie.

Metoda zostala stworzona aby uzyskac dostep i pozwolic na zmiane wartosci elementu tablicy punkty (wskazanego przez arg idx) przez klasy niezaprzyjaznione i niepochodne. Do pola przypisywana jest wartosc przekazana przez drugi argument. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy wypelnij (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

Parametry

idx	indeks elementu tablicy dynamicznej do zmiany wartosci.
liczba	wartosc , ktora ma byc przypisana elementowi tablicy punkty.

Oto graf wywołań dla tej funkcji:

Oto graf wywoływań tej funkcji:

4.9.3.5 set_punkty()

```
void wypelnij::set_punkty (
             int * p )
```

Metoda ustawia pole punkty.

Metoda zostala stworzona aby uzyskac dostep i pozwolic na zmiane wartosci pola punkty przez klasy niezaprzyjaznione i niepochodne. Do pola przypisywana jest wartosc przekazana przez argument. Jesli cos poszlo nie tak i nie ma dostepu do obiektu klasy wypelnij (lub obiektu klasy pochodnej), metoda wyrzuca wyjatek.

Parametry

p	wskaznik na tablice dynamiczna, na ktora ma wskazywac pole punkty.

Oto graf wywołań dla tej funkcji:

4.9.3.6 stworz_tab_wsp()

```
void wypelnij::stworz_tab_wsp ( )
```

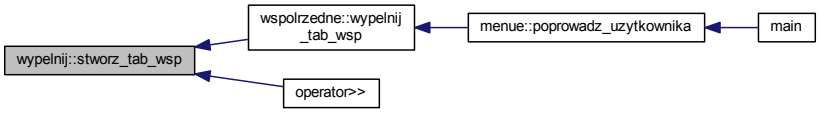
Metoda tworzy tablice dynamiczna punkty o rozmiarze 2 ∗ rozmiar ∗ rozmiar [jest to ilosc komorek na planszy ∗ 2(dwie wspolrzedne)] oraz wypelnia ja wartosciami -1.

Rozmiar stworzonej tablicy odpowia liczbie elementow planszy ∗ 2. Mozna wiec maksymalnie wpisac tyle elementow, zeby zaplenic cala stworzona plansze.

Zwraca

nic

Oto graf wywoływań tej funkcji:



4.9.3.7 usun_tab_wsp()

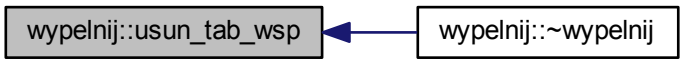
```
void wypelnij::usun_tab_wsp ( )
```

Metoda usuwa i zwalnia pamiec po tablicy dynamicznej punkty. Metoda zabezpieczona przed kilkukrotnym usunieciem Metoda wywolana w destruktorze.

Zwraca

nic

Oto graf wywoływań tej funkcji:



4.9.3.8 wypelnij_plansze()

```
void wypelnij::wypelnij_plansze ( )
```

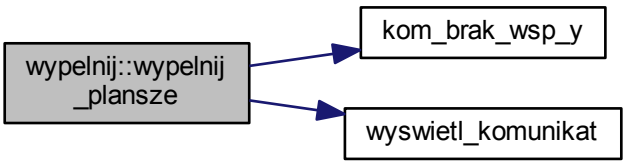
Metoda sluzy do wypelnienia tablicy dynamicznej punkty.

Metoda przesuwa wskaznik pomocniczy na element planszy wskzany przez wspolrzedne w tablicy punkty.\ Nastepnie ozywia ten element (zmiana pola klasy element_planszy czy_zyje na true). Metoda wykorzystuje przeciazony operator tablicowy [] do przesuniecia wskaznika pomocniczego na odpowiedni element.

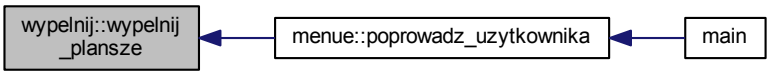
Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.9.4 Dokumentacja atrybutów składowych

4.9.4.1 punkty

```
int* wypelnij::punkty  [protected]
```

Pole wskaznikowe na int, sluzy do tworzenia tablicy dynamicznej z wspolrzednymi punktow do ozywienia.

Dokumentacja dla tej klasy została wygenerowana z plików:

- klasa_wypelnij.h
- metody_wypelnij.cpp

4.10 Dokumentacja klasy wyswietl

```
#include <klasa_wyswietl.h>
```

Diagram dziedziczenia dla wyswietl

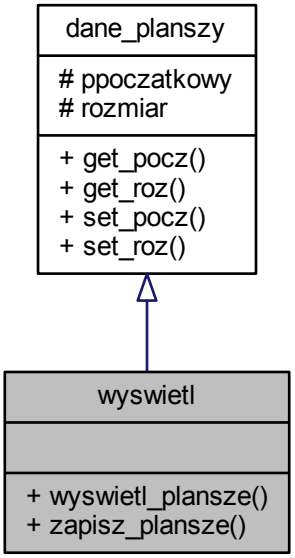
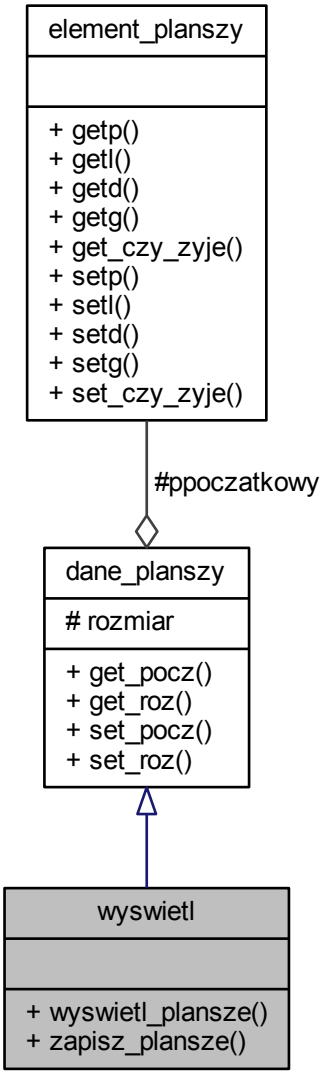


Diagram współpracy dla wyswietl:



Metody publiczne

- void wyswietl_plansze ()
- void zapisz_plansze ()

Dodatkowe Dziedziczone Składowe

4.10.1 Opis szczegółowy

Klasa sluzy do wyswietlania oraz wpisywania aktualnego stanu planszy do pliku. Jest klasa pochodna klasy dane_planszy. Rozszerza tylko o metody.

4.10.2 Dokumentacja funkcji składowych

4.10.2.1 wyswietl_plansze()

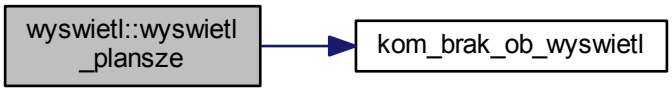
```
void wyswietl::wyswietl_plansze ( )
```

Metoda wyswietla aktualny stan planszy na strumien wyjsciowy. Metoda wykorzystuje przeciazony operator strumieniowy $<<$.

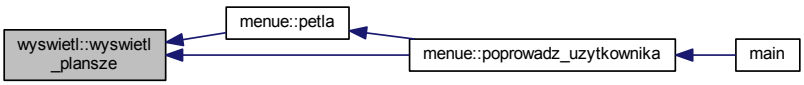
Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.10.2.2 zapisz_plansze()

```
void wyswietl::zapisz_plansze ( )
```

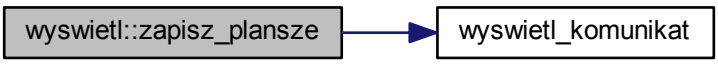
Metoda zapisuje sktualny stan planszy do pliku o podanej przez uzytkownika nazwie.

Najpierw zostaje wyswietlony komunikat proszacy o podanie nazwy pliku. Po podaniu nazwy otwieranyn jest plik. Wykorzystany przeciazony operator strumieniowy $<<$. Jezeli plik nie istnieje, wyrzucany jest wyjatek przez ten operator. Jesli istnieje dane zostaja wpisane do pliku.

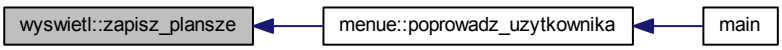
Zwraca

nic

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

- klasa_wyswietl.h
- metody_wyswietl.cpp

Rozdział 5

Dokumentacja plików

5.1 Dokumentacja pliku klasa_dane_planszy.h

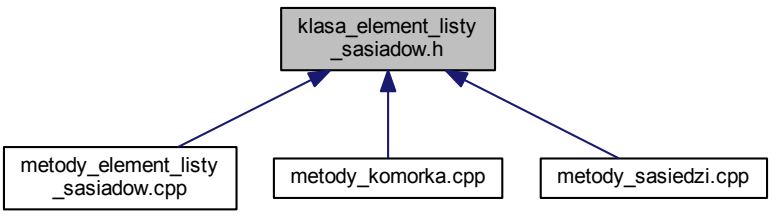
Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Komponenty

- class dane_planszy

5.2 Dokumentacja pliku klasa_element_listy_sasiadow.h

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

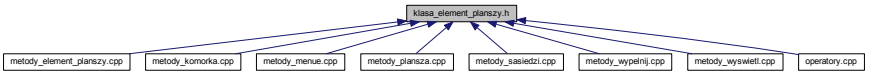


Komponenty

- class element_listy_sasiadow

5.3 Dokumentacja pliku klasa_element_planszy.h

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

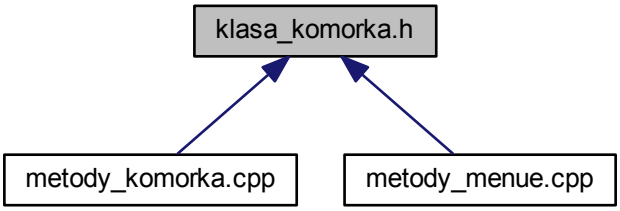


Komponenty

- class element_planszy

5.4 Dokumentacja pliku klasa_komorka.h

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

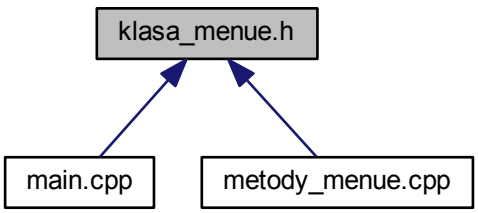


Komponenty

- class komorka

5.5 Dokumentacja pliku klasa_menue.h

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

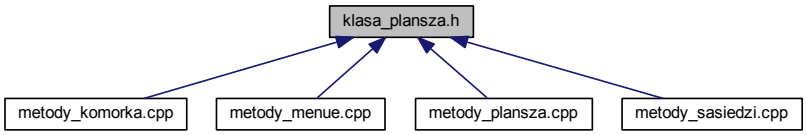


Komponenty

- class menue

5.6 Dokumentacja pliku klasa_plansza.h

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

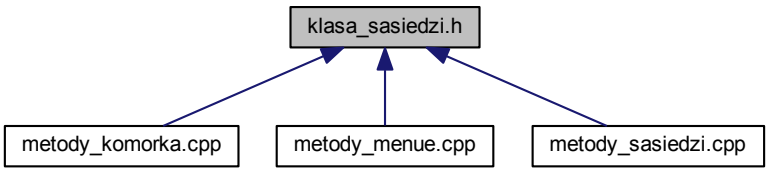


Komponenty

- class plansza

5.7 Dokumentacja pliku klasa_sasiedzi.h

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

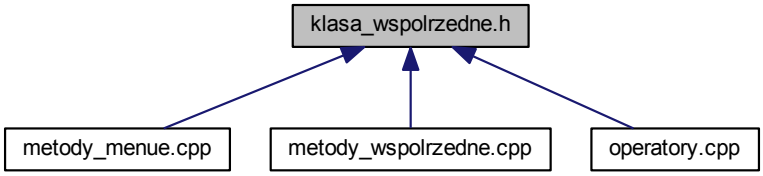


Komponenty

- class sasiedzi

5.8 Dokumentacja pliku klasa_wspolrzedne.h

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

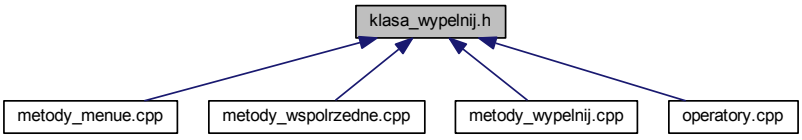


Komponenty

- class wspolrzedne

5.9 Dokumentacja pliku klasa_wypelnij.h

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

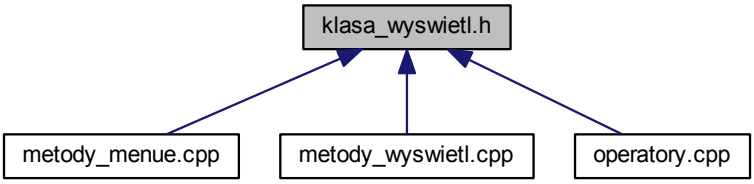


Komponenty

- class wypelnij

5.10 Dokumentacja pliku klasa_wyswietl.h

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



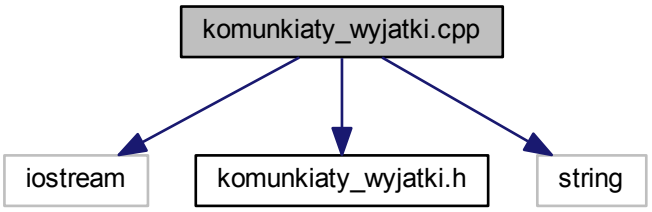
Komponenty

- class wyswietl

5.11 Dokumentacja pliku komunkiaty_wyjatki.cpp

```
#include <iostream>
#include "komunkiaty_wyjatki.h"
#include <string>
```

Wykres zależności załączania dla komunkiaty_wyjatki.cpp:



Funkcje

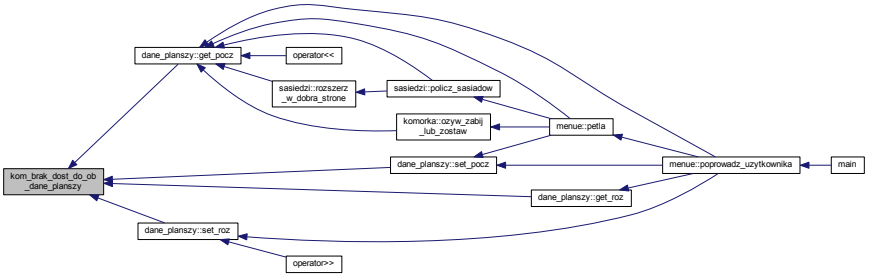
- void wyswietl_komunikat (std::string str)
- std::string kom_brak_dost_do_ob_dane_planszy ()
- std::string kom_brak_ob_element_planszy ()
- std::string kom_brak_ob_wypelnij ()
- std::string kom_wsp_spoza_planszy ()
- std::string kom_brak_wsp_y ()
- std::string kom_brak_ob_wyswietl ()
- std::string kom_brak_pliku ()
- std::string kom_brak_ob_element_listy_sasiadow ()

5.11.1 Dokumentacja funkcji

5.11.1.1 kom_brak_dost_do_ob_dane_planszy()

```
std::string kom_brak_dost_do_ob_dane_planszy ( )
```

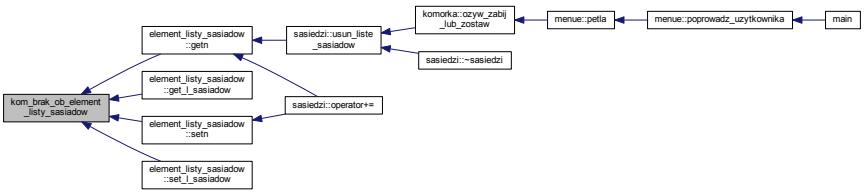
Funkcja zwraca napis informujacy o wyjatku w klasie dane_planszy. Napis informuje o braku dostepu do obiektu klasy dane_planszy. Oto graf wywoływań tej funkcji:



5.11.1.2 kom_brak_ob_element_listy_sasiadow()

```
std::string kom_brak_ob_element_listy_sasiadow ( )
```

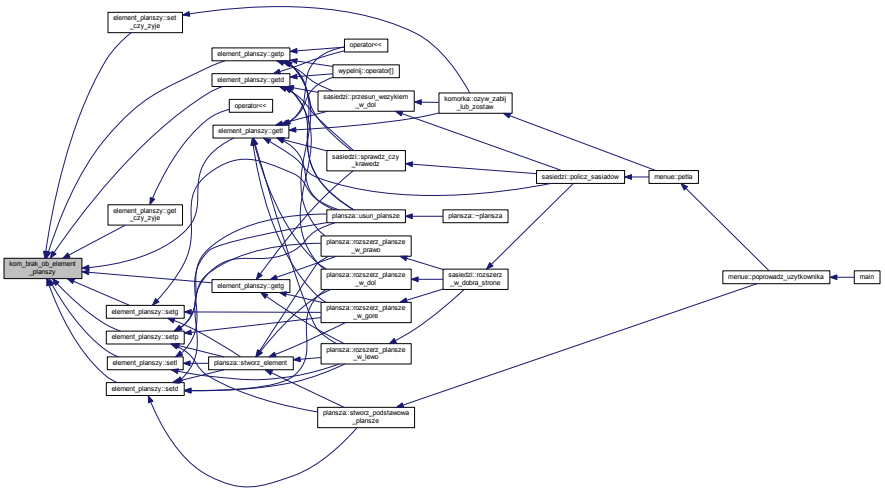
Funkcja zwraca napis informujacy o wyjatku w klasie element_listy_sasiadow. Napis informuje o braku dostepu do obiektu klasy element_listy_sasiadow. Oto graf wywoływań tej funkcji:



5.11.1.3 kom_brak_ob_element_planszy()

```
std::string kom_brak_ob_element_planszy ( )
```

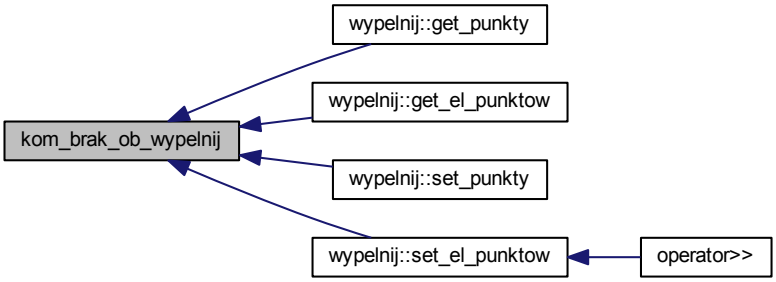
Funkcja zwraca napis informujacy o wyjatku w klasie element_planszy. Napis informuje o braku dostepu do obiektu klasy element_planszy. Oto graf wywoływań tej funkcji:



5.11.1.4 kom_brak_ob_wypelnij()

```
std::string kom_brak_ob_wypelnij ( )
```

Funkcja zwraca napis informujacy o wyjatku w klasie wypelnij. Napis informuje o braku dostepu do obiektu klasy wypelnij. Oto graf wywoływań tej funkcji:



5.11.1.5 kom_brak_ob_wyswietl()

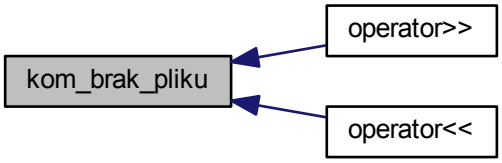
```
std::string kom_brak_ob_wyswietl ( )
```

Funkcja zwraca napis informujacy o wyjatku w klasie wyswietl. Napis informuje o braku dostepu do obiektu klasy wyswietl. Oto graf wywoływań tej funkcji:

5.11.1.6 kom_brak_pliku()

```
std::string kom_brak_pliku ( )
```

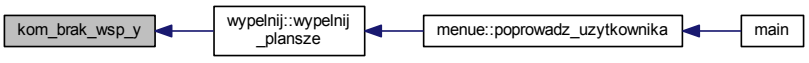
Funkcja zwraca napis informujacy o wyjatku w przeciazonym operatorze. Napis informuje o braku dostepu do pliku o podanej nazwie. Oto graf wywoływań tej funkcji:



5.11.1.7 kom_brak_wsp_y()

```
std::string kom_brak_wsp_y ( )
```

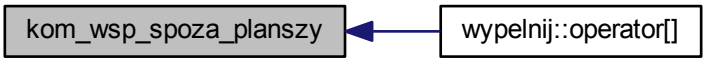
Funkcja zrwaca napis informujacy o wyjatku w klasie wypelnij. Napis informuje o braku drugiej wspolrzednej komorki do ozywienia. Oto graf wywoływań tej funkcji:



5.11.1.8 kom_wsp_spoza_planszy()

```
std::string kom_wsp_spoza_planszy ( )
```

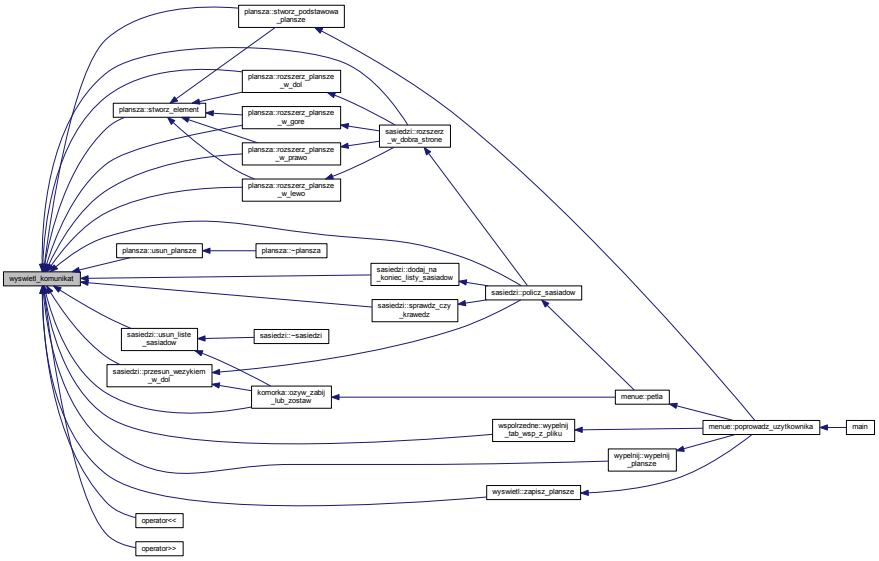
Funkcja zwraca napis informujacy o wyjatku w klasie wypelnij. Napis informuje o tym, ze wspolrzedna komorki do ozywienia wskazuje poza plansze. Wyjatek jest wyrzucony w metodzie wczytujacej dane z pliku(przy recznym wpisywaniu jest to od razu sprawdzane). Oto graf wywoływań tej funkcji:



5.11.1.9 wyswietl_komunikat()

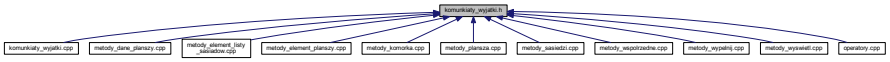
```
void wyswietl_komunikat (
            std::string str )
```

Funkcja wyswietla stringa o podanej nazwie. Jest uzywana do wyswietlania ewentualnych wyjatkow. Wyswietla inny komunikat w zaleznosci od tego, jaki wyjatek zostal wyrzucony. Oto graf wywoływań tej funkcji:



5.12 Dokumentacja pliku komunkiaty_wyjatki.h

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Funkcje

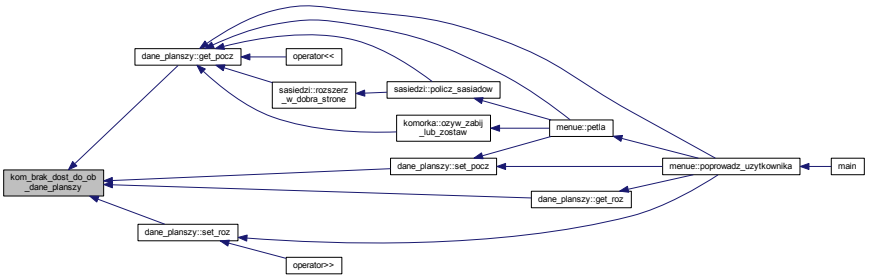
- void wyswietl_komunikat (std::string str)
- std::string kom_brak_dost_do_ob_dane_planszy ()
- std::string kom_brak_ob_element_planszy ()
- std::string kom_brak_ob_wypelnij ()
- std::string kom_wsp_spoza_planszy ()
- std::string kom_brak_wsp_y ()
- std::string kom_brak_ob_wyswietl ()
- std::string kom_brak_pliku ()
- std::string kom_brak_ob_element_listy_sasiadow ()

5.12.1 Dokumentacja funkcji

5.12.1.1 kom_brak_dost_do_ob_dane_planszy()

```
std::string kom_brak_dost_do_ob_dane_planszy ( )
```

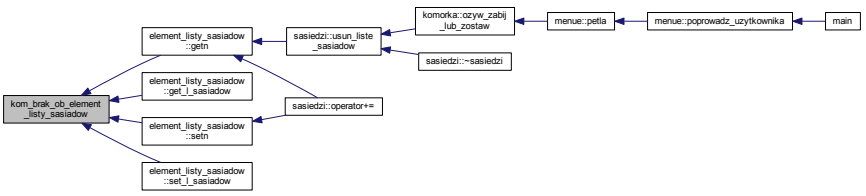
Funkcja zwraca napis informujacy o wyjatku w klasie dane_planszy. Napis informuje o braku dostepu do obiektu klasy dane_planszy. Oto graf wywoływań tej funkcji:



5.12.1.2 kom_brak_ob_element_listy_sasiadow()

```
std::string kom_brak_ob_element_listy_sasiadow ( )
```

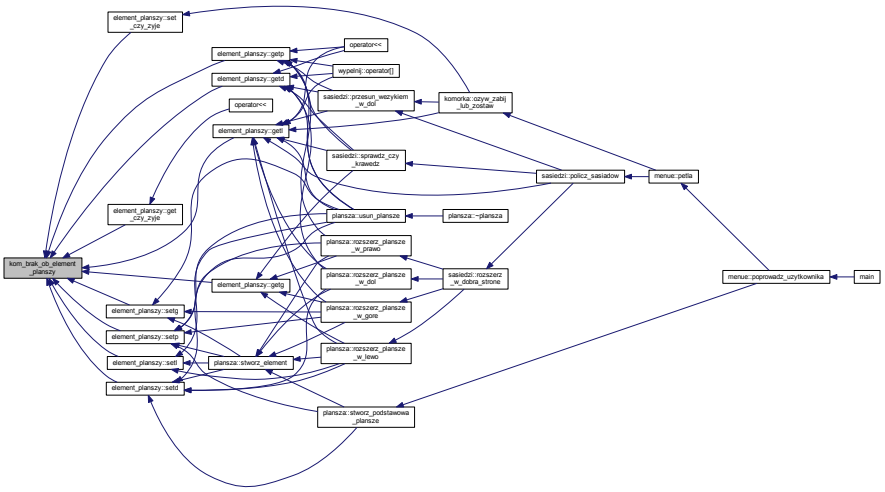
Funkcja zwraca napis informujacy o wyjatku w klasie element_listy_sasiadow. Napis informuje o braku dostepu do obiektu klasy element_listy_sasiadow. Oto graf wywoływań tej funkcji:



5.12.1.3 kom_brak_ob_element_planszy()

```
std::string kom_brak_ob_element_planszy ( )
```

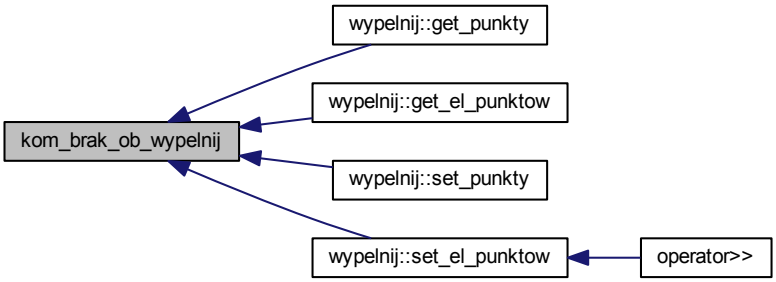
Funkcja zwraca napis informujacy o wyjatku w klasie element_planszy. Napis informuje o braku dostepu do obiektu klasy element_planszy. Oto graf wywoływań tej funkcji:



5.12.1.4 kom_brak_ob_wypelnij()

```
std::string kom_brak_ob_wypelnij ( )
```

Funkcja zwraca napis informujacy o wyjatku w klasie wypelnij. Napis informuje o braku dostepu do obiektu klasy wypelnij. Oto graf wywoływań tej funkcji:



5.12.1.5 kom_brak_ob_wyswietl()

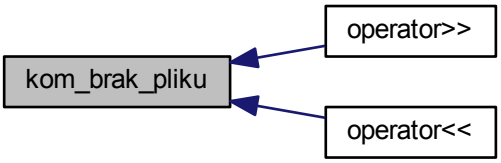
```
std::string kom_brak_ob_wyswietl ( )
```

Funkcja zwraca napis informujacy o wyjatku w klasie wyswietl. Napis informuje o braku dostepu do obiektu klasy wyswietl. Oto graf wywoływań tej funkcji:

5.12.1.6 kom_brak_pliku()

```
std::string kom_brak_pliku ( )
```

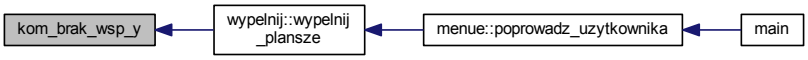
Funkcja zwraca napis informujacy o wyjatku w przeciazonym operatorze. Napis informuje o braku dostepu do pliku o podanej nazwie. Oto graf wywoływań tej funkcji:



5.12.1.7 kom_brak_wsp_y()

```
std::string kom_brak_wsp_y ( )
```

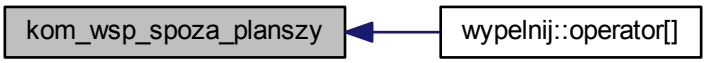
Funkcja zrwaca napis informujacy o wyjatku w klasie wypelnij. Napis informuje o braku drugiej wspolrzednej komorki do ozywienia. Oto graf wywoływań tej funkcji:



5.12.1.8 kom_wsp_spoza_planszy()

```
std::string kom_wsp_spoza_planszy ( )
```

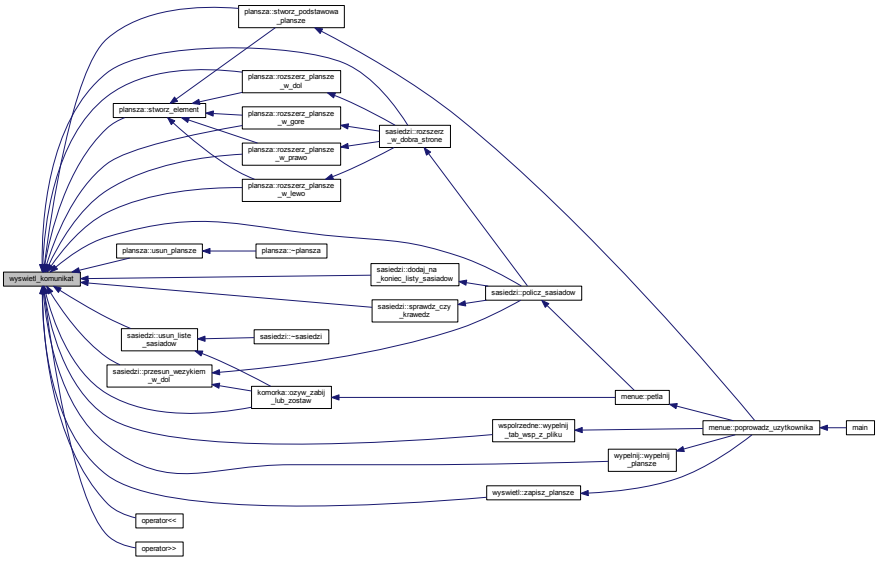
Funkcja zwraca napis informujacy o wyjatku w klasie wypelnij. Napis informuje o tym, ze wspolrzedna komorki do ozywienia wskazuje poza plansze. Wyjatek jest wyrzucony w metodzie wczytujacej dane z pliku(przy recznym wpisywaniu jest to od razu sprawdzane). Oto graf wywoływań tej funkcji:



5.12.1.9 wyswietl_komunikat()

```
void wyswietl_komunikat (
            std::string str )
```

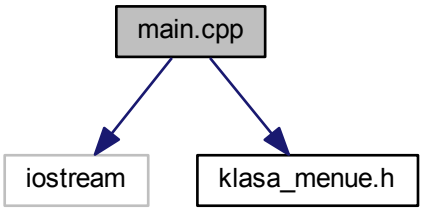
Funkcja wyswietla stringa o podanej nazwie. Jest uzywana do wyswietlania ewentualnych wyjatkow. Wyswietla inny komunikat w zaleznosci od tego, jaki wyjatek zostal wyrzucony. Oto graf wywoływań tej funkcji:



5.13 Dokumentacja pliku main.cpp

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

Wykres zależności załączania dla main.cpp:



Funkcje

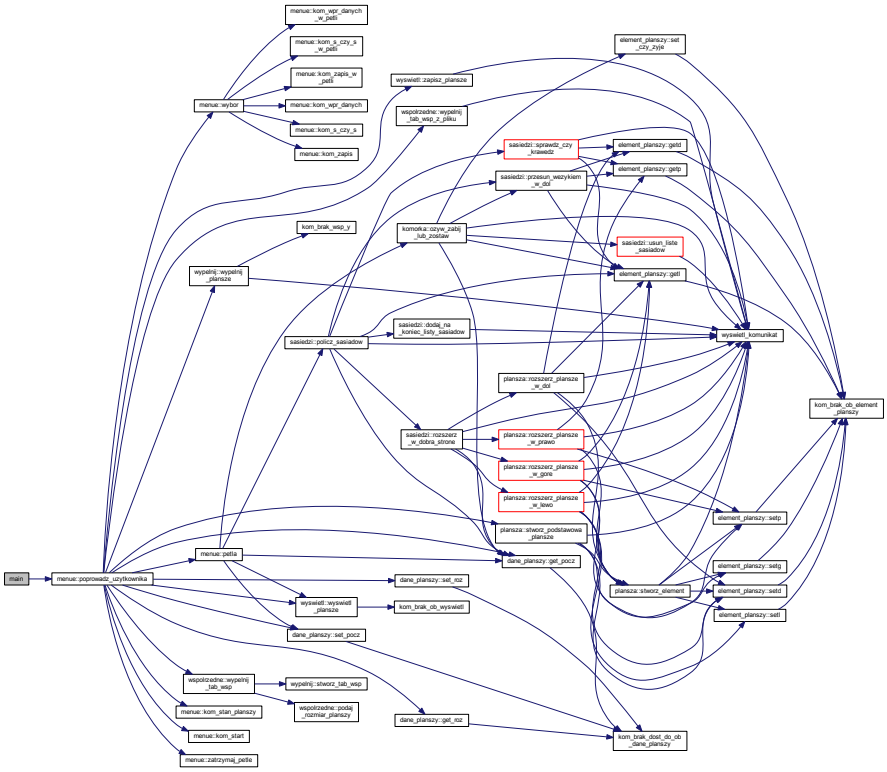
- int main ()

5.13.1 Dokumentacja funkcji

5.13.1.1 main()

```
int main ( )
```

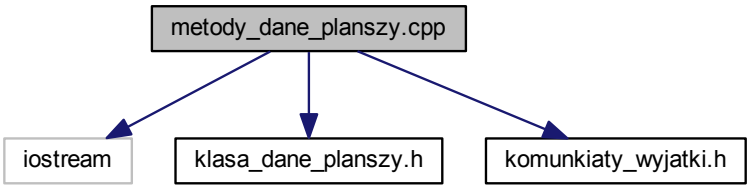
Oto graf wywołań dla tej funkcji:



5.14 Dokumentacja pliku metody_dane_planszy.cpp

```
#include <iostream>
#include "klasa_dane_planszy.h"
#include "komunkiaty_wyjatki.h"
```

Wykres zależności załączania dla metody_dane_planszy.cpp:

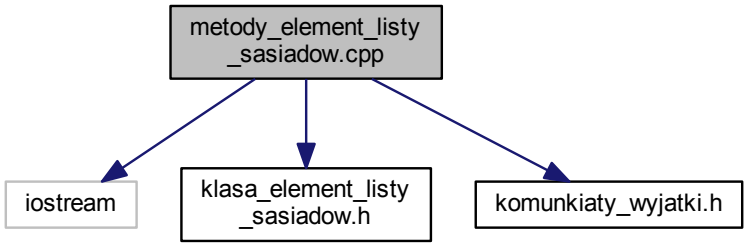


5.15 Dokumentacja pliku metody_element_listy_sasiadow.cpp

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

```
#include "komunkiaty_wyjatki.h"
```

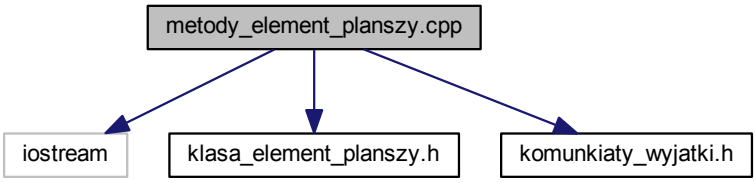
Wykres zależności załączania dla metody_element_listy_sasiadow.cpp:



5.16 Dokumentacja pliku metody_element_planszy.cpp

```
#include <iostream>
#include "klasa_element_planszy.h"
#include "komunkiaty_wyjatki.h"
```

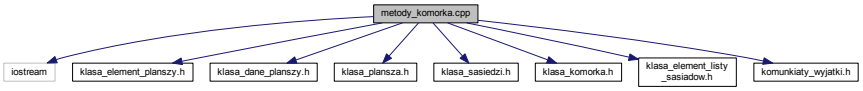
Wykres zależności załączania dla metody_element_planszy.cpp:



5.17 Dokumentacja pliku metody_komorka.cpp

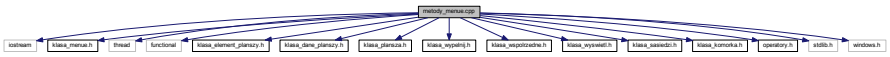
```
#include <iostream>
#include "klasa_element_planszy.h"
#include "klasa_dane_planszy.h"
#include "klasa_plansza.h"
#include "klasa_sasiedzi.h"
#include "klasa_komorka.h"
#include "klasa_element_listy_sasiadow.h"
```

```
#include "komunkiaty_wyjatki.h"
```
Wykres zależności załączania dla metody_komorka.cpp:



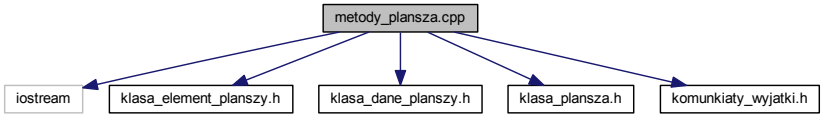
5.18 Dokumentacja pliku metody_menue.cpp

```
#include <iostream>
#include "klasa_menue.h"
#include <thread>
#include <functional>
#include "klasa_element_planszy.h"
#include "klasa_dane_planszy.h"
#include "klasa_plansza.h"
#include "klasa_wypelnij.h"
#include "klasa_wspolrzedne.h"
#include "klasa_wyswietl.h"
#include "klasa_sasiedzi.h"
#include "klasa_komorka.h"
#include "operatory.h"
#include <stdlib.h>
#include "windows.h"
```
Wykres zależności załączania dla metody_menue.cpp:



5.19 Dokumentacja pliku metody_plansza.cpp

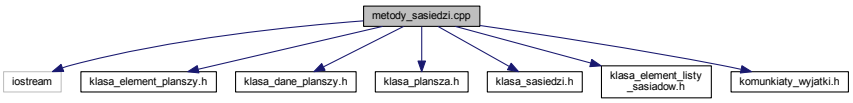
```
#include <iostream>
#include "klasa_element_planszy.h"
#include "klasa_dane_planszy.h"
#include "klasa_plansza.h"
#include "komunkiaty_wyjatki.h"
```
Wykres zależności załączania dla metody_plansza.cpp:



5.20 Dokumentacja pliku metody_sasiedzi.cpp

```
#include <iostream>
#include "klasa_element_planszy.h"
#include "klasa_dane_planszy.h"
#include "klasa_plansza.h"
#include "klasa_sasiedzi.h"
#include "klasa_element_listy_sasiadow.h"
#include "komunkiaty_wyjatki.h"
```

Wykres zależności załączania dla metody_sasiedzi.cpp:



5.21 Dokumentacja pliku metody_wspolrzedne.cpp

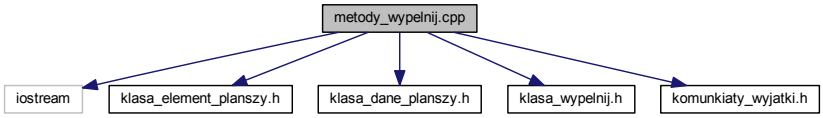
```
#include <iostream>
#include "klasa_dane_planszy.h"
#include "klasa_wypelnij.h"
#include "klasa_wspolrzedne.h"
#include <string>
#include <fstream>
#include "operatory.h"
#include "komunkiaty_wyjatki.h"
```

Wykres zależności załączania dla metody_wspolrzedne.cpp:

5.22 Dokumentacja pliku metody_wypelnij.cpp

```
#include <iostream>
#include "klasa_element_planszy.h"
#include "klasa_dane_planszy.h"
#include "klasa_wypelnij.h"
#include "komunkiaty_wyjatki.h"
```

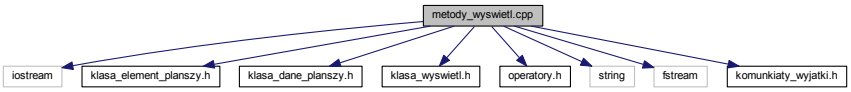
Wykres zależności załączania dla metody_wypelnij.cpp:



5.23 Dokumentacja pliku metody_wyswietl.cpp

```
#include <iostream>
#include "klasa_element_planszy.h"
#include "klasa_dane_planszy.h"
#include "klasa_wyswietl.h"
#include "operatory.h"
#include <string>
#include <fstream>
#include "komunkiaty_wyjatki.h"
```

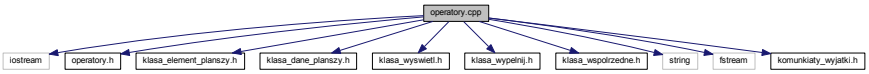
Wykres zależności załączania dla metody_wyswietl.cpp:



5.24 Dokumentacja pliku operatory.cpp

```
#include <iostream>
#include "operatory.h"
#include "klasa_element_planszy.h"
#include "klasa_dane_planszy.h"
#include "klasa_wyswietl.h"
#include "klasa_wypelnij.h"
#include "klasa_wspolrzedne.h"
#include <string>
#include <fstream>
#include "komunkiaty_wyjatki.h"
```

Wykres zależności załączania dla operatory.cpp:



Funkcje

- std::ostream & operator<< (std::ostream &s, element_planszy *ob)
- std::ostream & operator<< (std::ostream &s, wyswietl &ob)
- std::ifstream & operator>> (std::ifstream &plik, wspolrzedne &ob)
- std::ofstream & operator<< (std::ofstream &plik, wyswietl &ob)

5.24.1 Dokumentacja funkcji

5.24.1.1 operator<<() [1/3]

```
std::ostream& operator<< (
            std::ostream & s,
            element_planszy * ob )
```

Przeciazony operator strumieniowy << sluzy do wyswietlania stanu obiektu klasy element_planszy.

W zaleznosci od stanu (wartosci pola czy_zyje) elementu wskazywanego przez drugi argument operator "wrzuca" do strumienia ostream inne wartosci. " . " - gdy komorka nie zyje. " O " - gdy komorka zyje.

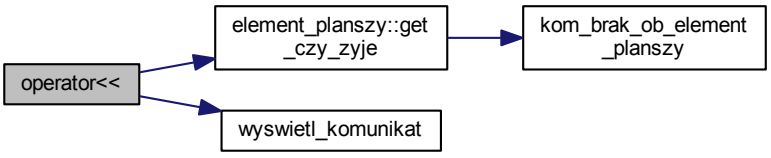
Parametry

s	strumien ostream, do ktorego beda "wrzucone" znaki do wyswietlenia.
ob	wskaznik na element, ktorego stan ma byc wyswietlony.

Zwraca

strumien ostream i to co w nim.

Oto graf wywołań dla tej funkcji:



5.24.1.2 operator<<() [2/3]

```
std::ostream& operator<< (
            std::ostream & s,
            wyswietl & ob )
```

Przeciazony operator strumieniowy << sluzy do wyswietlania aktualnego stanu planszy do gry w zycie.

Operator ustawia wskaznik pomocniczy na pierwszy element planszy(w obiekcie wyswietl). Przesuwa go po niej i uzywa przeciazonego operatora << dla elementu planszy do wyswietlenia jego stanu.

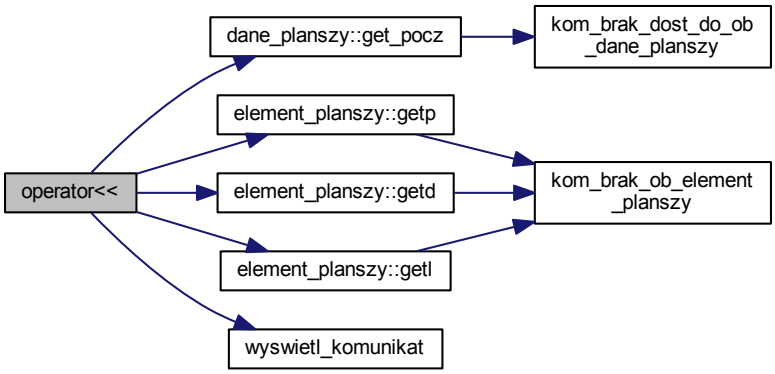
Parametry

s	strumien ostream, do ktorego beda "wrzucone" znaki do wyswietlenia.
ob	obiekt klasy wypelnij ktorego stan planszy pod wskaznikiem ppoczatkowy byc wyswietlony.

Zwraca

strumien ostream i to co w nim.

Oto graf wywołań dla tej funkcji:



5.24.1.3 operator<<() [3/3]

```
std::ofstream& operator<< (
            std::ofstream & plik,
            wyswietl & ob )
```

Przeciazony operator strumieniowy << sluzy do zapisania aktualnego stanu planszy do pliku.

Jesli plik podany jako pierwszy argument istnieje oraz wskaznik ppoczatkowy obiektu podanego jako drugi argument na cos wskazuje, operator zapisuje w pliku aktualny stan planszy wskazanej przez wskaznik ppoczatkowy obiektu klasy wspolrzedne. Operator uzywa pomocniczego wskaznika ktorym przesuwa sie po planszy i uzywa przeciazonego operatora << dla kazdego elementu planszy. Jesli nie ma planszy, operator nic nie robi. Jesli nie istnieje podalny plik, operator wyrzuca wyjatek.

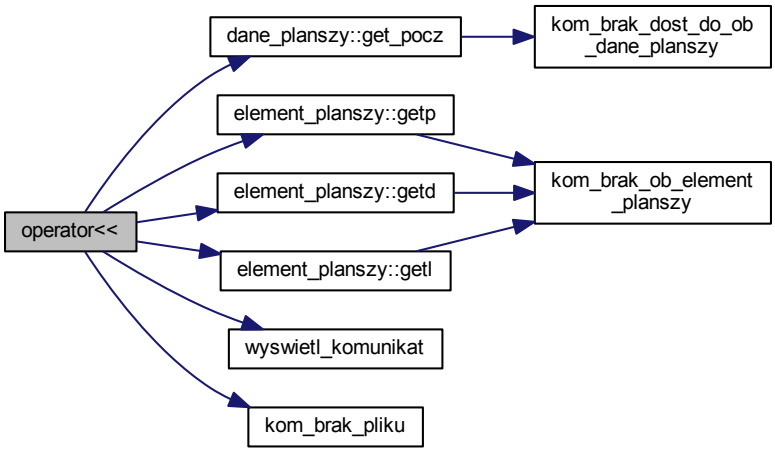
Parametry

plik	strumien ofstream, plik do ktorego zapisany bedzie aktualny stan planszy.
ob	obiekt klasy wyswietl, z ktorego ma zostac zapisany stan planszy(wskazywanej przez wskaznik ppoczatkowy).

Zwraca

strumien oftream.

Oto graf wywołań dla tej funkcji:



5.24.1.4 operator>>()

```
std::ifstream& operator>> (
            std::ifstream & plik,
            wspolrzedne & ob )
```

Przeciazony operator strumieniowy >> sluzy do wczytywania rozmiaru kwadrwtowej planszy poczatkowej oraz wspolrzednych komorek do ozywienia z pliku.

Jesli plik podany jako pierwszy argument istnieje, operator sczytuje rozmiar kwadratowej planszy podstawowej Nastepnie tworzy tablice z wspolrzednymi(pole punkty(wywolanie metody stworz_tab_wsp dla ob klasy wspolrzedne)). Natepnie dopoki w pliku istnieja jakies liczby operator sczytuje je do stworzonej tablicy. Operator wyzruca wyjatek gdy nie ma dostepu do pliku podanego jako pierwszy argument.

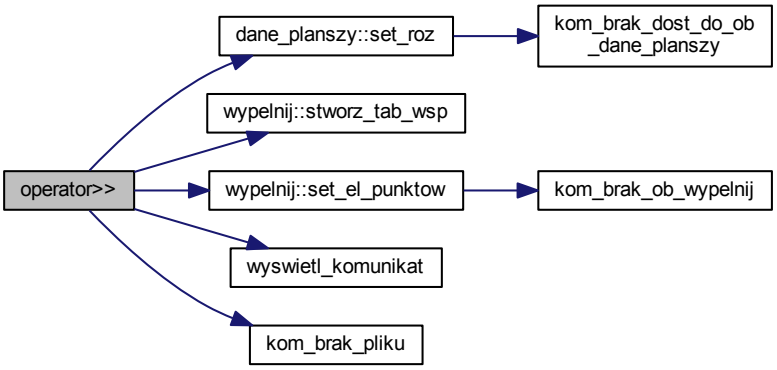
Parametry

plik	strumien ifstream, plik z ktorego beda wczytane dane.
ob	obiekt klasy wspolrzedne, do ktorego (do tablicy punkty) maja zostac wczytane wspolrzedne.

Zwraca

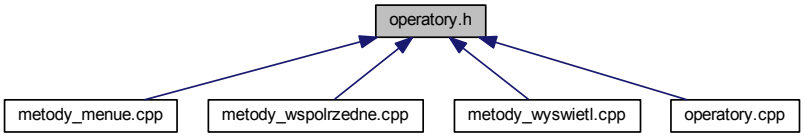
strumien iftream.

Oto graf wywołań dla tej funkcji:



5.25 Dokumentacja pliku operatory.h

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Funkcje

- std::ostream & operator<< (std::ostream &s, element_planszy *ob)
- std::ostream & operator<< (std::ostream &s, wyswietl &ob)
- std::ifstream & operator>> (std::ifstream &plik, wspolrzedne &ob)
- std::ofstream & operator<< (std::ofstream &plik, wyswietl &ob)

5.25.1 Dokumentacja funkcji

5.25.1.1 operator<<() [1/3]

```
std::ostream& operator<< (
            std::ostream & s,
            element_planszy * ob )
```

Przeciazony operator strumieniowy << sluzy do wyswietlania stanu obiektu klasy element_planszy.

W zaleznosci od stanu (wartosci pola czy_zyje) elementu wskazywanego przez drugi argument operator "wrzuca" do strumienia ostream inne wartosci. " . " - gdy komorka nie zyje. " O " - gdy komorka zyje.

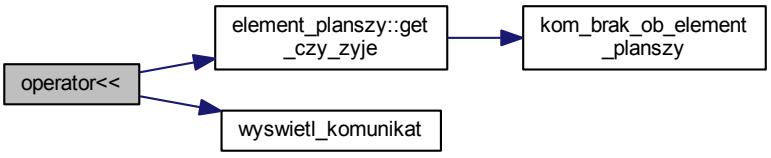
Parametry

s	strumien ostream, do ktorego beda "wrzucone" znaki do wyswietlenia.
ob	wskaznik na element, ktorego stan ma byc wyswietlony.

Zwraca

strumien ostream i to co w nim.

Oto graf wywołań dla tej funkcji:



5.25.1.2 operator<<() [2/3]

```
std::ostream& operator<< (
            std::ostream & s,
            wyswietl & ob )
```

Przeciazony operator strumieniowy << sluzy do wyswietlania aktualnego stanu planszy do gry w zycie.

Operator ustawia wskaznik pomocniczy na pierwszy element planszy(w obiekcie wyswietl). Przesuwa go po niej i uzywa przeciazonego operatora << dla elementu planszy do wyswietlenia jego stanu.

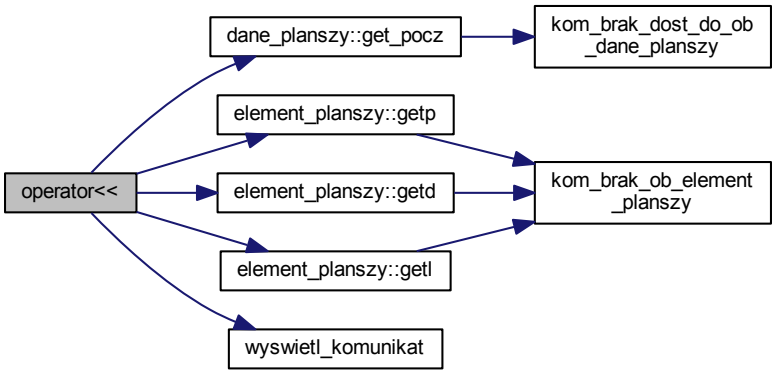
Parametry

s	strumien ostream, do ktorego beda "wrzucone" znaki do wyswietlenia.
ob	obiekt klasy wypelnij ktorego stan planszy pod wskaznikiem ppoczatkowy byc wyswietlony.

Zwraca

strumien ostream i to co w nim.

Oto graf wywołań dla tej funkcji:



5.25.1.3 operator<<() [3/3]

```
std::ofstream& operator<< (
            std::ofstream & plik,
            wyswietl & ob )
```

Przeciazony operator strumieniowy << sluzy do zapisania aktualnego stanu planszy do pliku.

Jesli plik podany jako pierwszy argument istnieje oraz wskaznik ppoczatkowy obiektu podanego jako drugi argument na cos wskazuje, operator zapisuje w pliku aktualny stan planszy wskazanej przez wskaznik ppoczatkowy obiektu klasy wspolrzedne. Operator uzywa pomocniczego wskaznika ktorym przesuwa sie po planszy i uzywa przeciazonego operatora << dla kazdego elementu planszy. Jesli nie ma planszy, operator nic nie robi. Jesli nie istnieje podalny plik, operator wyrzuca wyjatek.

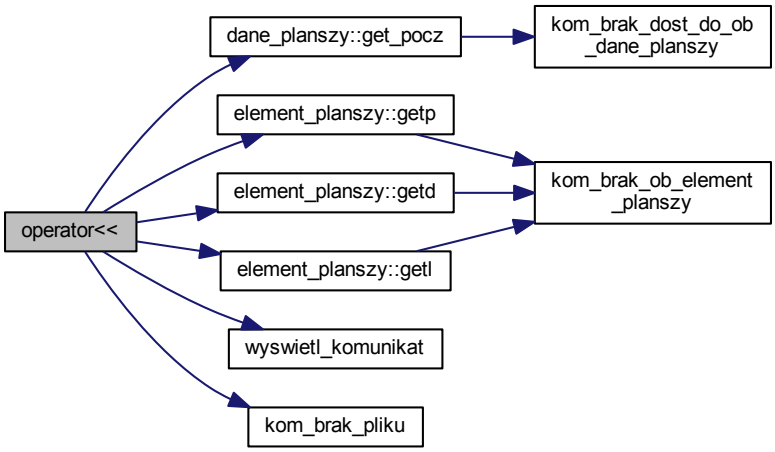
Parametry

plik	strumien ofstream, plik do ktorego zapisany bedzie aktualny stan planszy.
ob	obiekt klasy wyswietl, z ktorego ma zostac zapisany stan planszy(wskazywanej przez wskaznik ppoczatkowy).

Zwraca

strumien oftream.

Oto graf wywołań dla tej funkcji:



5.25.1.4 operator>>()

```
std::ifstream& operator>> (
            std::ifstream & plik,
            wspolrzedne & ob )
```

Przeciazony operator strumieniowy >> sluzy do wczytywania rozmiaru kwadrwtowej planszy poczatkowej oraz wspolrzednych komorek do ozywienia z pliku.

Jesli plik podany jako pierwszy argument istnieje, operator sczytuje rozmiar kwadratowej planszy podstawowej Nastepnie tworzy tablice z wspolrzednymi(pole punkty(wywolanie metody stworz_tab_wsp dla ob klasy wspolrzedne)). Natepnie dopoki w pliku istnieja jakies liczby operator sczytuje je do stworzonej tablicy. Operator wyzruca wyjatek gdy nie ma dostepu do pliku podanego jako pierwszy argument.

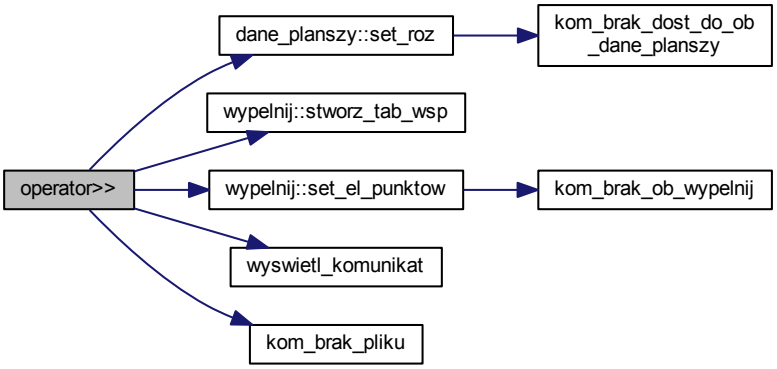
Parametry

plik	strumien ifstream, plik z ktorego beda wczytane dane.
ob	obiekt klasy wspolrzedne, do ktorego (do tablicy punkty) maja zostac wczytane wspolrzedne.

Zwraca

strumien iftream.

Oto graf wywołań dla tej funkcji:



Skorowidz