

Selekcja Naturalna

1.0

Wygenerowano za pomocą Doxygen 1.14.0

Rozdział 1

Indeks przestrzeni nazw

1.1 Lista pakietów

Oto lista pakietów wraz z ich krótkimi opisami (jeśli są dostępne):

[proj.selekcjanatur](#) ??

Rozdział 2

Indeks hierarchiczny

2.1 Hierarchia klas

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

proj.selekcjanatur.AppController	??
Application	
proj.selekcjanatur.App	??
proj.selekcjanatur.Czlowiek	??
proj.selekcjanatur.Kobieta	??
proj.selekcjanatur.Mezczyzna	??
proj.selekcjanatur.Gen	??
proj.selekcjanatur.GenPredkoscChodzenia	??
proj.selekcjanatur.GenWzrok	??
proj.selekcjanatur.GenZapotrzebowanieEnergetyczne	??
proj.selekcjanatur.InterfejsSymulacji	??
proj.selekcjanatur.Symulacja	??
proj.selekcjanatur.SymulacjaPlik	??
proj.selekcjanatur.Jedzenie	??
proj.selekcjanatur.StartController	??

Rozdział 3

Indeks klas

3.1 Lista klas

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

proj.selekcjanatur.App	Główna klasa aplikacji dziedzicząca po Application z JavaFX	??
proj.selekcjanatur.AppController	Główny kontroler zarządzający interfejsem i logiką symulacji	??
proj.selekcjanatur.Czlowiek	Abstrakcyjna klasa reprezentująca człowieka w symulacji ewolucyjnej	??
proj.selekcjanatur.Gen	Interfejs reprezentujący gen w symulacji ewolucyjnej	??
proj.selekcjanatur.GenPredkoscChodzenia	Klasa implementująca gen prędkości chodzenia	??
proj.selekcjanatur.GenWzrok	Klasa implementująca gen zasięgu wzroku	??
proj.selekcjanatur.GenZapotrzebowanieEnergetyczne	Klasa implementująca gen zapotrzebowania energetycznego	??
proj.selekcjanatur.InterfejsSymulacji	Główny interfejs systemu symulacji ewolucyjnej	??
proj.selekcjanatur.Jedzenie	Klasa modelująca pożywienie w symulacji ewolucyjnej	??
proj.selekcjanatur.Kobieta	Specjalizacja klasy Czlowiek reprezentująca kobietę	??
proj.selekcjanatur.Mezczyzna	Specjalizacja klasy Czlowiek reprezentująca mężczyznę	??
proj.selekcjanatur.StartController	Kontroler GUI dla ekranu konfiguracji symulacji	??
proj.selekcjanatur.Symulacja	Implementacja symulacji ewolucyjnej z dziedziczeniem cech	??
proj.selekcjanatur.SymulacjaPlik	Klasa odtwarzająca symulację na podstawie zapisanego dziennika zdarzeń	??

Rozdział 4

Indeks plików

4.1 Lista plików

Tutaj znajduje się lista wszystkich plików wraz z ich krótkimi opisami:

src/main/java/ module-info.java	
Deklaracja modułu Java dla aplikacji Symulacja Selekcji Naturalnej	??
src/main/java/proj/selekcjanatur/ App.java	
Główna klasa aplikacji odpowiedzialna za uruchomienie symulacji	??
src/main/java/proj/selekcjanatur/ AppController.java	
Kontroler głównego interfejsu symulacji ewolucyjnej	??
src/main/java/proj/selekcjanatur/ Czlowiek.java	
Abstrakcyjna klasa bazowa reprezentująca człowieka w symulacji	??
src/main/java/proj/selekcjanatur/ Gen.java	
Interfejs definiujący podstawowe zachowania genów w symulacji	??
src/main/java/proj/selekcjanatur/ GenPredkoscChodzenia.java	
Implementacja genu odpowiedzialnego za prędkość poruszania się	??
src/main/java/proj/selekcjanatur/ GenWzrok.java	
Implementacja genu odpowiedzialnego za zasięg wzroku	??
src/main/java/proj/selekcjanatur/ GenZapotrzebowanieEnergetyczne.java	
Implementacja genu określającego metabolizm organizmu	??
src/main/java/proj/selekcjanatur/ InterfejsSymulacji.java	
Interfejs definiujący podstawowe operacje symulacji ewolucyjnej	??
src/main/java/proj/selekcjanatur/ Jedzenie.java	
Klasa reprezentująca źródło pożywienia w symulacji	??
src/main/java/proj/selekcjanatur/ Kobieta.java	
Klasa reprezentująca kobietę w symulacji ewolucyjnej	??
src/main/java/proj/selekcjanatur/ Mezcyzna.java	
Klasa reprezentująca mężczyznę w symulacji ewolucyjnej	??
src/main/java/proj/selekcjanatur/ StartController.java	
Kontroler ekranu startowego symulacji	??
src/main/java/proj/selekcjanatur/ Symulacja.java	
Główna klasa implementująca logikę symulacji ewolucyjnej	??
src/main/java/proj/selekcjanatur/ SymulacjaPlik.java	
Implementacja odtwarzania symulacji z pliku dziennika zdarzeń	??
src/main/resources/proj/selekcjanatur/ grid-view.fxml	??
src/main/resources/proj/selekcjanatur/ start-view.fxml	??

Rozdział 5

Dokumentacja przestrzeni nazw

5.1 Pakiet proj.selekcjanatur

Komponenty

- class [App](#)
Główna klasa aplikacji dziedzicząca po Application z JavaFX.
- class [AppController](#)
Główny kontroler zarządzający interfejsem i logiką symulacji.
- class [Czlowiek](#)
Abstrakcyjna klasa reprezentująca człowieka w symulacji ewolucyjnej.
- interface [Gen](#)
Interfejs reprezentujący gen w symulacji ewolucyjnej.
- class [GenPredkoscChodzenia](#)
Klasa implementująca gen prędkości chodzenia.
- class [GenWzrok](#)
Klasa implementująca gen zasięgu wzroku.
- class [GenZapotrzebowanieEnergetyczne](#)
Klasa implementująca gen zapotrzebowania energetycznego.
- interface [InterfejsSymulacji](#)
Główny interfejs systemu symulacji ewolucyjnej.
- class [Jedzenie](#)
Klasa modelująca pożywienie w symulacji ewolucyjnej.
- class [Kobieta](#)
Specjalizacja klasy [Czlowiek](#) reprezentująca kobietę
- class [Meczyzna](#)
Specjalizacja klasy [Czlowiek](#) reprezentująca mężczyznę
- class [StartController](#)
Kontroler GUI dla ekranu konfiguracji symulacji.
- class [Symulacja](#)
Implementacja symulacji ewolucyjnej z dziedziczeniem cech.
- class [SymulacjaPlik](#)
Klasa odtwarzająca symulację na podstawie zapisanego dziennika zdarzeń

Rozdział 6

Dokumentacja klas

6.1 Dokumentacja klasy proj.selekcjanatur.App

Główna klasa aplikacji dziedzicząca po Application z JavaFX.

Diagram dziedziczenia dla proj.selekcjanatur.App

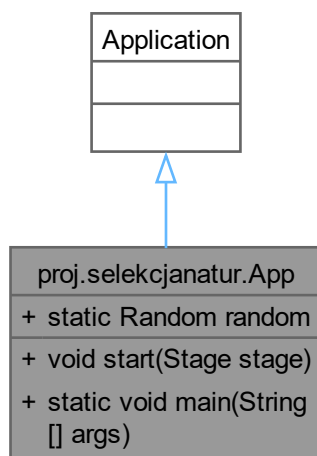
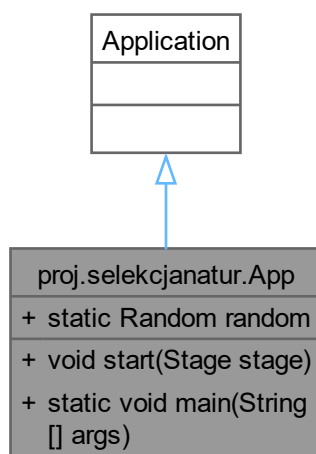


Diagram współpracy dla proj.selekcjanatur.App:



Metody publiczne

- void `start` (Stage stage) throws Exception
Metoda startująca aplikację JavaFX.

Statyczne metody publiczne

- static void `main` (String[] args)
Główna metoda uruchamiająca aplikację.

Statyczne atrybuty publiczne

- static Random `random` = new Random()
Współdzielony generator liczb pseudolosowych.

6.1.1 Opis szczegółowy

Główna klasa aplikacji dziedzicząca po `Application` z JavaFX.

Klasa odpowiedzialna za inicjalizację interfejsu użytkownika oraz uruchomienie symulacji.

6.1.2 Dokumentacja funkcji składowych

6.1.2.1 main()

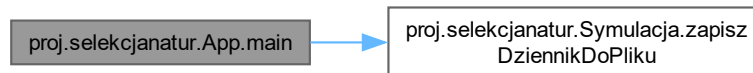
```
void proj.selekcjanatur.App.main (  
    String[] args) [static]
```

Główna metoda uruchamiająca aplikację.

Parametry

<code>args</code>	Argumenty wiersza poleceń (niewykorzystywane).
-------------------	--

Po zamknięciu aplikacji, jeśli nie zapisano do pliku, automatycznie zapisuje dziennik zdarzeń. Oto graf wywołań dla tej funkcji:



6.1.2.2 start()

```
void proj.selekcjanatur.App.start (
    Stage stage) throws Exception
```

Metoda startująca aplikację JavaFX.

Parametry

<code>stage</code>	Główny kontener okna aplikacji.
--------------------	---------------------------------

Wyjątki

<code>Exception</code>	Może wyrzucić wyjątek przy problemach z ładowaniem FXML.
------------------------	--

Ładuje plik FXML z interfejsem użytkownika, ustawia scenę i tytuł okna, a następnie je wyświetla.

6.1.3 Dokumentacja atrybutów składowych

6.1.3.1 random

```
Random proj.selekcjanatur.App.random = new Random() [static]
```

Współdzielony generator liczb pseudolosowych.

Używany w całej aplikacji do generowania losowych wartości.

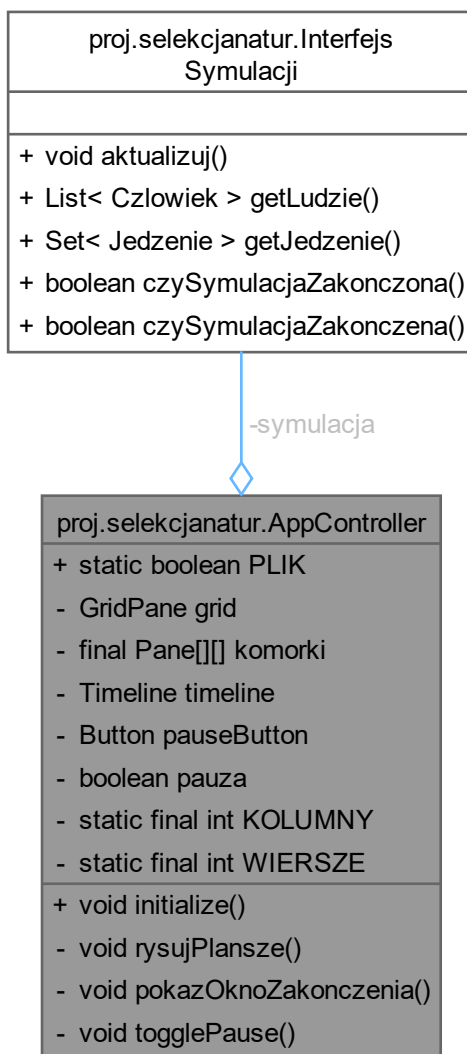
Dokumentacja dla tej klasy została wygenerowana z pliku:

- [src/main/java/proj/selekcjanatur/App.java](#)

6.2 Dokumentacja klasy proj.selekcjanatur.AppController

Główny kontroler zarządzający interfejsem i logiką symulacji.

Diagram współpracy dla proj.selekcjanatur.AppController:



Metody publiczne

- void `initialize()`
Metoda inicjalizująca kontroler.

Statyczne atrybuty publiczne

- static boolean `PLIK` = false
Flaga określająca czy symulacja działa w trybie odtwarzania z pliku.

Metody prywatne

- void `rysujPlansze ()`
Aktualizuje wygląd planszy na podstawie stanu symulacji.
- void `pokazOknoZakonczenia ()`
Wyświetla okno dialogowe po zakończeniu symulacji.
- void `togglePause ()`
Obsługuje wciśnięcie przycisku pauzy.

Atrybuty prywatne

- GridPane `grid`
Kontener GridPane przechowujący siatkę komórek.
- final Pane[] [] `komorki` = new Pane[WIERSZE][KOLUMNY]
Dwuwymiarowa tablica przechowująca panele komórek.
- InterfejsSymulacji `symulacja`
Referencja do obiektu symulacji.
- Timeline `timeline`
Timeline kontrolująca cykl symulacji.
- Button `pauseButton`
Przycisk do wstrzymywania/wznawiania symulacji.
- boolean `pauza` = false
Flaga określająca czy symulacja jest wstrzymana.

Statyczne atrybuty prywatne

- static final int `KOLUMNY` = `Symulacja.szerokosc`
Szerokość planszy w komórkach (pobrana z klasy `Symulacja`)
- static final int `WIERSZE` = `Symulacja.wysokosc`
Wysokość planszy w komórkach (pobrana z klasy `Symulacja`)

6.2.1 Opis szczegółowy

Główny kontroler zarządzający interfejsem i logiką symulacji.

Klasa odpowiedzialna za:

- Inicjalizację siatki komórek
- Obsługę symulacji
- Wyświetlanie symulacji
- Obsługę zdarzeń kończących symulację

6.2.2 Dokumentacja funkcji składowych

6.2.2.1 initialize()

```
void proj.selekcjanatur.AppController.initialize ()
```

Metoda inicjalizująca kontroler.

Wykonywana automatycznie po załadowaniu widoku FXML. Tworzy siatkę komórek, inicjalizuje symulację i uruchamia timer.

6.2.2.2 pokazOknoZakonczenia()

```
void proj.selekcjanatur.AppController.pokazOknoZakonczenia () [private]
```

Wyświetla okno dialogowe po zakończeniu symulacji.

W zależności od trybu (plik/live) wyświetla odpowiedni komunikat i w trybie live zapisuje dziennik zdarzeń do pliku. Oto graf wywołań dla tej funkcji:



6.2.2.3 rysujPlansze()

```
void proj.selekcjanatur.AppController.rysujPlansze () [private]
```

Aktualizuje wygląd planszy na podstawie stanu symulacji.

Metoda:

1. Resetuje wszystkie komórki do koloru białego
2. Rysuje jedzenie na ciemnozielono
3. Rysuje ludzi (mężczyzn na niebiesko, kobiety na różowo, dzieci z gradientem)

6.2.2.4 togglePause()

```
void proj.selekcjanatur.AppController.togglePause () [private]
```

Obsługuje wciśnięcie przycisku pauzy.

Zmienia stan symulacji (pauza/wznów) i aktualizuje tekst przycisku

6.2.3 Dokumentacja atrybutów składowych

6.2.3.1 grid

```
GridPane proj.selekcjanatur.AppController.grid [private]
```

Kontener GridPane przechowujący siatkę komórek.

6.2.3.2 KOLUMNY

```
final int proj.selekcjanatur.AppController.KOLUMNY = Symulacja.szerekosc [static], [private]
```

Szerokość planszy w komórkach (pobrana z klasy [Symulacja](#))

6.2.3.3 komorki

```
final Pane [][] proj.selekcjanatur.AppController.komorki = new Pane[WIERSZE][KOLUMNY] [private]
```

Dwuwymiarowa tablica przechowująca panele komórek.

6.2.3.4 pauseButton

```
Button proj.selekcjanatur.AppController.pauseButton [private]
```

Przycisk do wstrzymywania/wznawiania symulacji.

6.2.3.5 pauza

```
boolean proj.selekcjanatur.AppController.pauza = false [private]
```

Flaga określająca czy symulacja jest wstrzymana.

6.2.3.6 PLIK

```
boolean proj.selekcjanatur.AppController.PLIK = false [static]
```

Flaga określająca czy symulacja działa w trybie odtwarzania z pliku.

6.2.3.7 symulacja

```
InterfejsSymulacji proj.selekcjanatur.AppController.symulacja [private]
```

Referencja do obiektu symulacji.

6.2.3.8 timeline

```
Timeline proj.selekcjanatur.AppController.timeline [private]
```

Timeline kontrolująca cykl symulacji.

6.2.3.9 WIERSZE

```
final int proj.selekcjanatur.AppController.WIERSZE = Symulacja.wysokosc [static], [private]
```

Wysokość planszy w komórkach (pobrana z klasy [Symulacja](#))

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [src/main/java/proj/selekcjanatur/AppController.java](#)

6.3 Dokumentacja klasy proj.selekcjanatur.Czlowiek

Abstrakcyjna klasa reprezentująca człowieka w symulacji ewolucyjnej.

Diagram dziedziczenia dla proj.selekcjanatur.Czlowiek



Diagram współpracy dla proj.selekcjanatur.Czlowiek:



Metody publiczne

- `Czlowiek` (`int x`, `int y`)
Konstruktor tworzący nowego człowieka.
- `boolean jestDzieckiem` ()
Sprawdza czy człowiek jest dzieckiem.
- `int zasiegWzroku` ()

- *Oblicza zasięg wzroku na podstawie genów.*
• int `predkosc` ()
- *Oblicza prędkość poruszania na podstawie genów.*
• float `zuzycieEnergii` ()
- *Oblicza zużycie energii na cykl.*
• void `aktualizuj` ()
- *Aktualizuje stan człowieka w każdym cyklu symulacji.*
• boolean `powinnienSieRuszyć` (int klatki)
- *Decyduje czy człowiek powinien się poruszyć w danej klatce.*
• abstract boolean `mozeRozmnazac` ()
- *Sprawdza możliwość rozmnożenia.*
• void `zjedz` (`Jedzenie` jedzenie)
- *Zjada podane jedzenie.*
• abstract boolean `czyMezyczna` ()
- *Sprawdza płeć człowieka.*
• boolean `czyUmieraZeStarosci` ()
- *Sprawdza czy człowiek umiera ze starości.*
• `Czlowiek rozmnazajZ` (`Czlowiek` inny)
- *Rozmnaża się z innym człowiekiem.*
• String `toString` ()
- *Reprezentacja tekstowa człowieka.*

Atrybuty publiczne

- int `x`
Pozycja x na planszy.
- int `y`
Pozycja y na planszy.
- final int `id`
Unikalny identyfikator człowieka.

Atrybuty chronione

- boolean `zywy` = true
Flaga określająca czy człowiek jest żywy.
- float `poziomGlodu` = 100
Poziom głodu (0-100)
- int `czasOdRozmnazania` = 20
Liczba cykli od ostatniego rozmnożenia.
- `Gen[] geny` = new `Gen`[3]
Tablica genów człowieka (wzrok, prędkość, metabolizm)
- int `wiek` = 0
Wiek człowieka w cyklach symulacji.

Statyczne atrybuty prywatne

- static int `globalneId` = 0
Licznik globalny do generowania unikalnych ID.

6.3.1 Opis szczegółowy

Abstrakcyjna klasa reprezentująca człowieka w symulacji ewolucyjnej.

Klasa zawiera podstawowe mechanizmy:

- Śledzenie stanu życiowego
- Zarządzanie głodem i energią
- Mechanizmy rozmnażania
- Dziedziczenie genów
- Starzenie się i śmierć

6.3.2 Dokumentacja konstruktora i destruktora

6.3.2.1 Czlowiek()

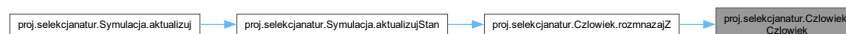
```
proj.selekcjanatur.Czlowiek.Czlowiek (
    int x,
    int y)
```

Konstruktor tworzący nowego człowieka.

Parametry

x	Pozycja startowa x
y	Pozycja startowa y

Oto graf wywołań tej funkcji:



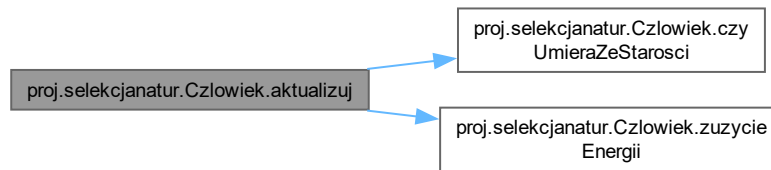
6.3.3 Dokumentacja funkcji składowych

6.3.3.1 aktualizuj()

```
void proj.selekcjanatur.Czlowiek.aktualizuj ()
```

Aktualizuje stan człowieka w każdym cyklu symulacji.

Zwiększa wiek, zmniejsza poziom energii, sprawdza warunki śmierci z głodu i starości Oto graf wywołań dla tej funkcji:



6.3.3.2 czyMezyczna()

```
abstract boolean proj.selekcjanatur.Czlowiek.czyMezyczna () [abstract]
```

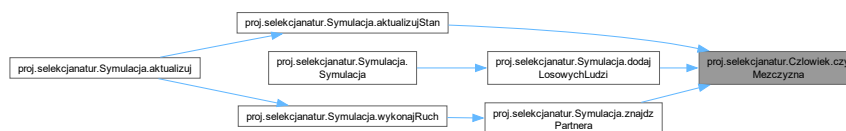
Sprawdza płeć człowieka.

Zwraca

true jeśli to mężczyzna

Reimplementowana w [proj.selekcjanatur.Kobieta](#) i [proj.selekcjanatur.Mezyczna](#).

Oto graf wywołań tej funkcji:



6.3.3.3 czyUmieraZeStarosci()

```
boolean proj.selekcjanatur.Czlowiek.czyUmieraZeStarosci ()
```

Sprawdza czy człowiek umiera ze starości.

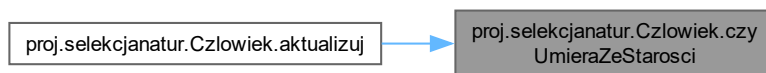
Zwraca

true jeśli umiera

Szansa rośnie liniowo od wieku 70 cykli:

- 70 cykli: 0% szansy
- 100 cykli: 30% szansy
- 170 cykli: 100% szansy

Oto graf wywołań tej funkcji:

**6.3.3.4 jestDzieckiem()**

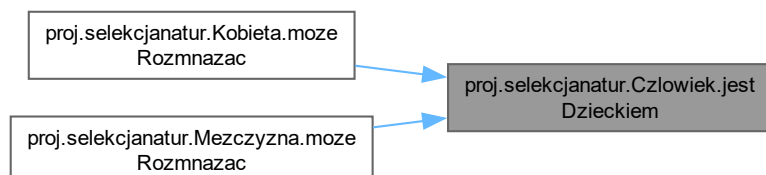
```
boolean proj.selekcjanatur.Czlowiek.jestDzieckiem ()
```

Sprawdza czy człowiek jest dzieckiem.

Zwraca

true jeśli wiek < 18 cykli

Oto graf wywołań tej funkcji:



6.3.3.5 mozeRozmnazac()

```
abstract boolean proj.selekcjanatur.Czlowiek.mozeRozmnazac () [abstract]
```

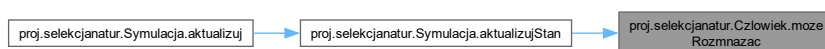
Sprawdza możliwość rozmnożenia.

Zwraca

true jeśli może się rozmnożyć

Reimplementowana w [proj.selekcjanatur.Kobieta](#) i [proj.selekcjanatur.Mezczyzna](#).

Oto graf wywoływań tej funkcji:



6.3.3.6 powinnySieRuszyć()

```
boolean proj.selekcjanatur.Czlowiek.powinnySieRuszyć (
    int klatki)
```

Decyduje czy człowiek powinien się poruszyć w danej klatce.

Parametry

<i>klatki</i>	Numer klatki symulacji
---------------	------------------------

Zwraca

true jeśli powinien się poruszyć

Oto graf wywołań dla tej funkcji:



6.3.3.7 predkosc()

```
int proj.selekcjanatur.Czlowiek.predkosc ()
```

Oblicza prędkość poruszania na podstawie genów.

Zwraca

Prędkość (1-3)

Oto graf wywołań tej funkcji:



6.3.3.8 rozmnazajZ()

```
Czlowiek proj.selekcjanatur.Czlowiek.rozmnazajZ (
    Czlowiek inny)
```

Rozmnaża się z innym człowiekiem.

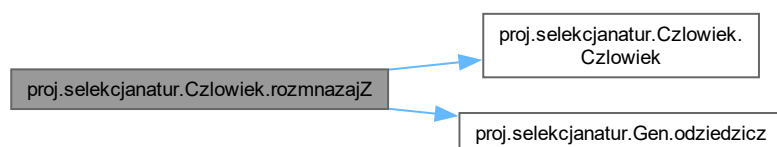
Parametry

<i>inny</i>	Drugi rodzic
-------------	--------------

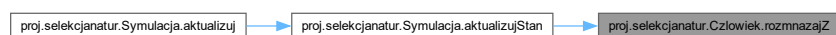
Zwraca

Nowy człowiek (dziecko)

Dziecko dziedziczy losowo geny od rodziców, powstaje w pozycji pierwszego rodzica Oto graf wywołań dla tej funkcji:



Oto graf wywołań tej funkcji:



6.3.3.9 toString()

```
String proj.selekcjanatur.Czlowiek.toString ()
```

Reprezentacja tekstowa człowieka.

Zwraca

String w formacie "Klasa@ID"

6.3.3.10 zasiegWzroku()

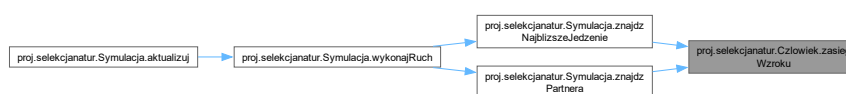
```
int proj.selekcjanatur.Czlowiek.zasiegWzroku ()
```

Oblicza zasięg wzroku na podstawie genów.

Zwraca

Zasięg wzroku w komórkach

Oto graf wywoływań tej funkcji:

**6.3.3.11 zjedz()**

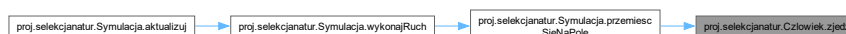
```
void proj.selekcjanatur.Czlowiek.zjedz (
    Jedzenie jedzenie)
```

Zjada podane jedzenie.

Parametry

<code>Jedzenie</code>	Obiekt jedzenia do spożycia
-----------------------	-----------------------------

Oto graf wywoływań tej funkcji:



6.3.3.12 zuzycieEnergii()

```
float proj.selekcjanatur.Czlowiek.zuzycieEnergii ()
```

Oblicza zużycie energii na cykl.

Zwraca

Wartość zużycia energii

Oto graf wywoływań tej funkcji:



6.3.4 Dokumentacja atrybutów składowych

6.3.4.1 czasOdRozmnazania

```
int proj.selekcjanatur.Czlowiek.czasOdRozmnazania = 20 [protected]
```

Liczba cykli od ostatniego rozmnożenia.

6.3.4.2 geny

```
Gen [] proj.selekcjanatur.Czlowiek.geny = new Gen[3] [protected]
```

Tablica genów człowieka (wzrok, prędkość, metabolizm)

6.3.4.3 globalneId

```
int proj.selekcjanatur.Czlowiek.globalneId = 0 [static], [private]
```

Licznik globalny do generowania unikalnych ID.

6.3.4.4 id

```
final int proj.selekcjanatur.Czlowiek.id
```

Unikalny identyfikator człowieka.

6.3.4.5 poziomGłodu

```
float proj.selekcjanatur.Czlowiek.poziomGłodu = 100 [protected]
```

Poziom głodu (0-100)

6.3.4.6 wiek

```
int proj.selekcjanatur.Czlowiek.wiek = 0 [protected]
```

Wiek człowieka w cyklach symulacji.

6.3.4.7 x

```
int proj.selekcjanatur.Czlowiek.x
```

Pozycja x na planszy.

6.3.4.8 y

```
int proj.selekcjanatur.Czlowiek.y
```

Pozycja y na planszy.

6.3.4.9 zywy

```
boolean proj.selekcjanatur.Czlowiek.zywy = true [protected]
```

Flaga określająca czy człowiek jest żywy.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [src/main/java/proj/selekcjanatur/Czlowiek.java](#)

6.4 Dokumentacja interfejsu proj.selekcjanatur.Gen

Interfejs reprezentujący gen w symulacji ewolucyjnej.

Diagram dziedziczenia dla proj.selekcjanatur.Gen

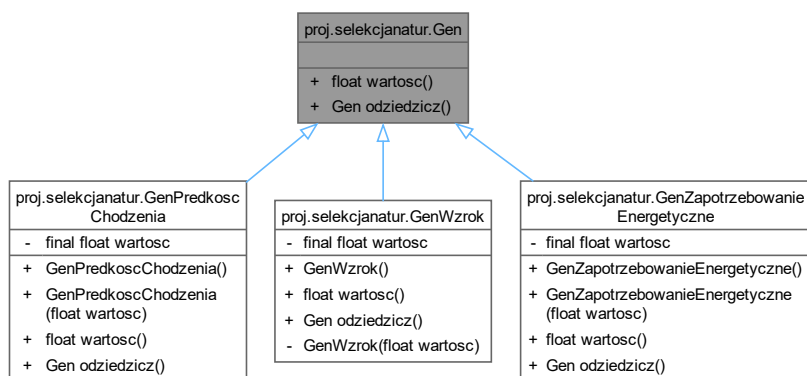
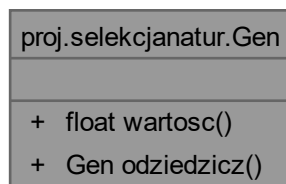


Diagram współpracy dla proj.selekcjanatur.Gen:



Metody publiczne

- float [wartosc](#) ()
Pobiera aktualną wartość cechy genetycznej.
- [Gen odziedzicz](#) ()
Tworzy nowy gen poprzez dziedziczenie z mutacją

6.4.1 Opis szczegółowy

Interfejs reprezentujący gen w symulacji ewolucyjnej.

Definiuje podstawowe operacje jakie musi implementować każdy gen:

- Dostęp do swojej wartości
- Mechanizm dziedziczenia z możliwością mutacji

6.4.2 Dokumentacja funkcji składowych

6.4.2.1 odziedzicz()

`Gen proj.selekcjanatur.Gen.odziedzicz ()`

Tworzy nowy gen poprzez dziedziczenie z mutacją

Zwraca

Nowa instancja genu potomnego

Metoda powinna:

1. Skopiować podstawowe cechy genu rodzica
2. Zastosować losową mutację (zmianę wartości)
3. Zwrócić nową instancję genu

Przykład:

```
Gen oryginalny = new GenPredkoscChodzenia();
Gen potomny = oryginalny.odziedzicz(); // nowa instancja z mutacją
```

Implementowany w [proj.selekcjanatur.GenPredkoscChodzenia](#), [proj.selekcjanatur.GenWzrok](#) i [proj.selekcjanatur.GenZapotrzebowani](#)

Oto graf wywoływań tej funkcji:



6.4.2.2 wartosc()

`float proj.selekcjanatur.Gen.wartosc ()`

Pobiera aktualną wartość cechy genetycznej.

Zwraca

Wartość liczbowa genu (zależna od implementacji)

Wartość może być stała lub zmieniać się w czasie, w zależności od konkretnej implementacji genu.

Przykład:

```
Gen wzrok = new GenWzrok();
float zasieg = wzrok.wartosc(); // np. 2.5f
```

Implementowany w [proj.selekcjanatur.GenPredkoscChodzenia](#), [proj.selekcjanatur.GenWzrok](#) i [proj.selekcjanatur.GenZapotrzebowani](#)

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- `src/main/java/proj/selekcjanatur/Gen.java`

6.5 Dokumentacja klasy proj.selekcjanatur.GenPredkoscChodzenia

Klasa implementująca gen prędkości chodzenia.

Diagram dziedziczenia dla proj.selekcjanatur.GenPredkoscChodzenia

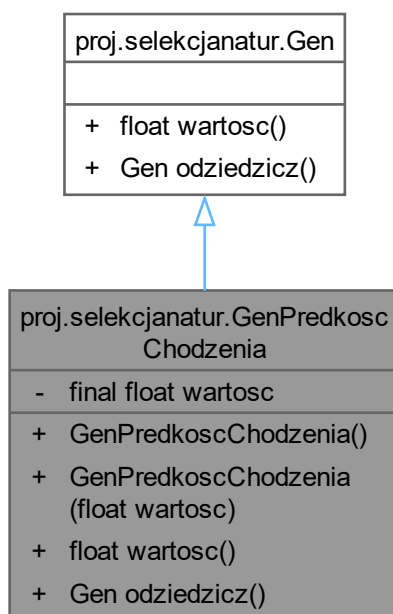
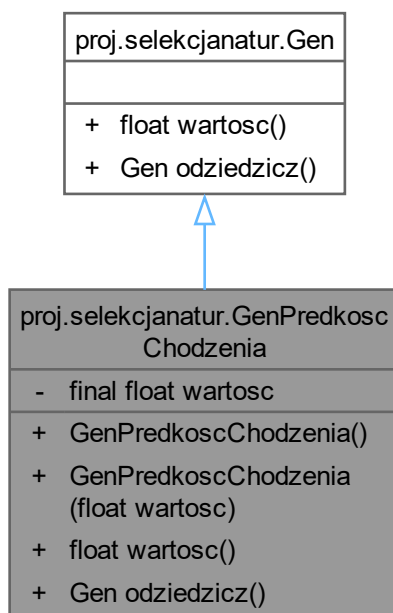


Diagram współpracy dla proj.selekcjanatur.GenPredkoscChodzenia:



Metody publiczne

- `GenPredkoscChodzenia ()`
Konstruktor losujący początkową wartość genu.
- `GenPredkoscChodzenia (float wartosc)`
Konstruktor z określeniem konkretnej wartości.
- `float wartosc ()`
Zwraca aktualną wartość genu.
- `Gen odziedzicz ()`
Tworzy nowy gen poprzez dziedziczenie z mutacją

Atrybuty prywatne

- `final float wartosc`
Przechowuje wartość genu prędkości.

6.5.1 Opis szczegółowy

Klasa implementująca gen prędkości chodzenia.

Gen określa z jaką częstotliwością człowiek porusza się w symulacji. Wartość genu jest z zakresu 1.0-3.0 gdzie:

- 1.0 = najwolniejszy ruch (co 3 klatki)
- 2.0 = średnia prędkość (co 2 klatki)
- 3.0 = najszybszy ruch (co klatkę)

Dziedziczenie odbywa się z możliwością mutacji ± 0.75 , przy czym wartość jest zawsze ograniczana do zakresu 1-3.

6.5.2 Dokumentacja konstruktora i destruktora

6.5.2.1 GenPredkoscChodzenia() [1/2]

```
proj.selekcjanatur.GenPredkoscChodzenia.GenPredkoscChodzenia ()
```

Konstruktor losujący początkową wartość genu.

Generuje wartość całkowitą z przedziału 1-3 Oto graf wywoływań tej funkcji:



6.5.2.2 GenPredkoscChodzenia() [2/2]

```
proj.selekcjanatur.GenPredkoscChodzenia.GenPredkoscChodzenia (
    float wartosc)
```

Konstruktor z określeniem konkretnej wartości.

Parametry

<u>wartosc</u>	Początkowa wartość genu (zostanie ograniczona do 1-3)
----------------	---

6.5.3 Dokumentacja funkcji składowych

6.5.3.1 odziedzicz()

```
Gen proj.selekcjanatur.GenPredkoscChodzenia.odziedzicz ()
```

Tworzy nowy gen poprzez dziedziczenie z mutacją

Zwraca

Nowa instancja [GenPredkoscChodzenia](#)

Algorytm działania:

1. Losuje mutację z zakresu ± 0.75
2. Dodaje mutację do obecnej wartości
3. Ogranicza wynik do zakresu 1.0-3.0

4. Zwraca nowy gen z zmutowaną wartością

Przykład:

```
Gen gen = new GenPredkoscChodzenia(2.0f);  
Gen potomek = gen.odziedzicz(); // np. 2.3
```

Implementuje [proj.selekcjanatur.Gen](#).

Oto graf wywołań dla tej funkcji:



6.5.3.2 wartosc()

```
float proj.selekcjanatur.GenPredkoscChodzenia.wartosc ()
```

Zwraca aktualną wartość genu.

Zwraca

Wartość prędkości w zakresie 1.0-3.0

Implementuje [proj.selekcjanatur.Gen](#).

6.5.4 Dokumentacja atrybutów składowych

6.5.4.1 wartosc

```
final float proj.selekcjanatur.GenPredkoscChodzenia.wartosc [private]
```

Przechowuje wartość genu prędkości.

Wartość z przedziału 1.0-3.0

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `src/main/java/proj/selekcjanatur/GenPredkoscChodzenia.java`

6.6 Dokumentacja klasy proj.selekcjanatur.GenWzrok

Klasa implementująca gen zasięgu wzroku.

Diagram dziedziczenia dla proj.selekcjanatur.GenWzrok

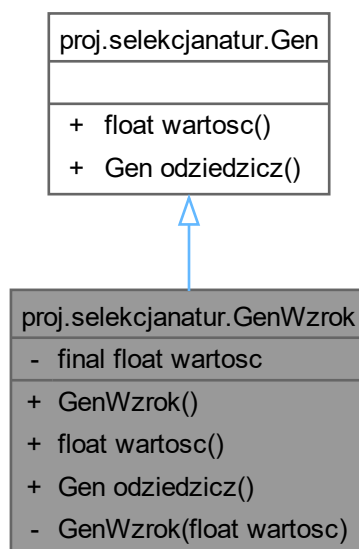
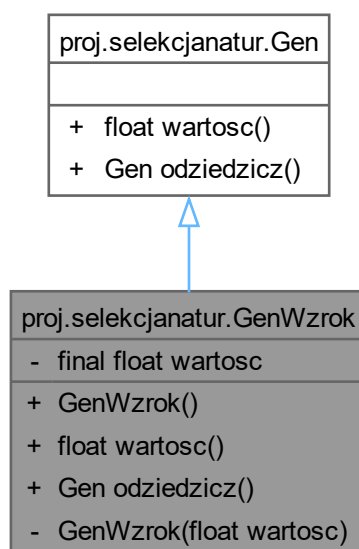


Diagram współpracy dla proj.selekcjanatur.GenWzrok:



Metody publiczne

- `GenWzrok ()`
Konstruktor losujący początkową wartość genu.
- `float wartosc ()`
Zwraca aktualną wartość genu.
- `Gen odziedzicz ()`
Tworzy nowy gen poprzez dziedziczenie z mutacją

Metody prywatne

- `GenWzrok (float wartosc)`
Prywatny konstruktor do tworzenia genów potomnych.

Atrybuty prywatne

- `final float wartosc`
Przechowuje wartość genu zasięgu wzroku.

6.6.1 Opis szczegółowy

Klasa implementująca gen zasięgu wzroku.

`Gen` określa jak daleko człowiek widzi w symulacji. Wartość genu jest z zakresu 1.0-10.0 gdzie:

- 1.0 = minimalny zasięg wzroku (1 komórka)
- 10.0 = maksymalny zasięg wzroku (10 komórek)

Dziedziczenie odbywa się z możliwością mutacji ± 1.0 , przy czym wartość jest zawsze ograniczana do zakresu 1-10. Konstruktor prywatny jest używany tylko przy dziedziczeniu.

6.6.2 Dokumentacja konstruktora i destruktora**6.6.2.1 GenWzrok() [1/2]**

```
proj.selekcjanatur.GenWzrok.GenWzrok ()
```

Konstruktor losujący początkową wartość genu.

Generuje wartość całkowitą z przedziału 1-10 Oto graf wywołań tej funkcji:

**6.6.2.2 GenWzrok() [2/2]**

```
proj.selekcjanatur.GenWzrok.GenWzrok (
    float wartosc) [private]
```

Prywatny konstruktor do tworzenia genów potomnych.

Parametry

<code>wartosc</code>	Określona wartość genu (zostanie ograniczona do 1-10)
----------------------	---

6.6.3 Dokumentacja funkcji składowych

6.6.3.1 odziedzicz()

```
Gen proj.selekcjanatur.GenWzrok.odziedzicz ()
```

Tworzy nowy gen poprzez dziedziczenie z mutacją

Zwraca

Nowa instancja [GenWzrok](#)

Algorytm działania:

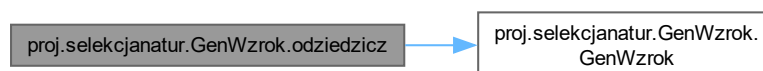
1. Losuje mutację z zakresu ± 1.0
2. Dodaje mutację do obecnej wartości
3. Ogranicza wynik do zakresu 1.0-10.0
4. Zwraca nowy gen z zmutowaną wartością

Przykład:

```
Gen gen = new GenWzrok();  
Gen potomek = gen.odziedzicz(); // np. 5.3
```

Implementuje [proj.selekcjanatur.Gen](#).

Oto graf wywołań dla tej funkcji:



6.6.3.2 wartosc()

```
float proj.selekcjanatur.GenWzrok.wartosc ()
```

Zwraca aktualną wartość genu.

Zwraca

Wartość zasięgu wzroku w zakresie 1.0-10.0

Implementuje [proj.selekcjanatur.Gen](#).

6.6.4 Dokumentacja atrybutów składowych

6.6.4.1 wartosc

```
final float proj.selekcjanatur.GenWzrok.wartosc [private]
```

Przechowuje wartość genu zasięgu wzroku.

Wartość z przedziału 1.0-10.0

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `src/main/java/proj/selekcjanatur/GenWzrok.java`

6.7 Dokumentacja klasy proj.selekcjanatur.GenZapotrzebowanieEnergetyczne

Klasa implementująca gen zapotrzebowania energetycznego.

Diagram dziedziczenia dla proj.selekcjanatur.GenZapotrzebowanieEnergetyczne

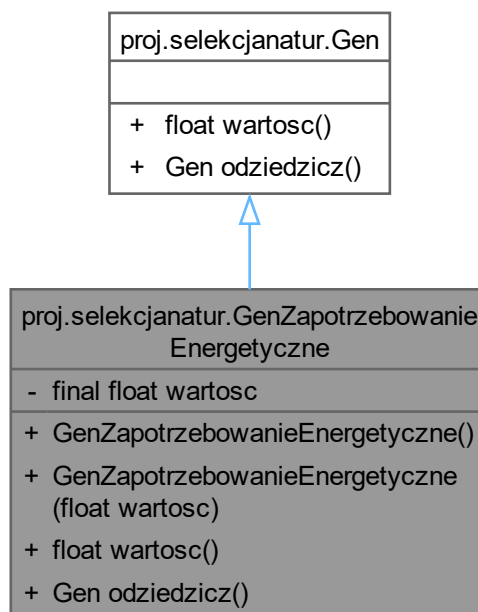
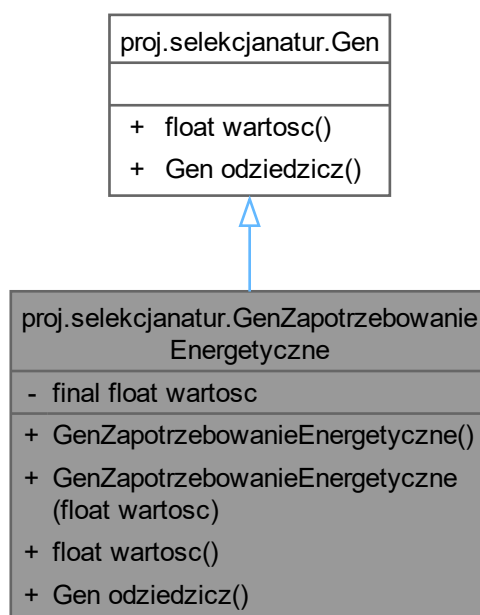


Diagram współpracy dla proj.selekcjanatur.GenZapotrzebowanieEnergetyczne:



Metody publiczne

- [GenZapotrzebowanieEnergetyczne \(\)](#)
Konstruktor losujący początkową wartość genu.
- [GenZapotrzebowanieEnergetyczne \(float wartosc\)](#)
Konstruktor z określeniem konkretnej wartości.
- float [wartosc \(\)](#)
Zwraca aktualną wartość genu.
- [Gen odziedzicz \(\)](#)
Tworzy nowy gen poprzez dziedziczenie z mutacją

Atrybuty prywatne

- final float [wartosc](#)
Przechowuje wartość genu metabolizmu.

6.7.1 Opis szczegółowy

Klasa implementująca gen zapotrzebowania energetycznego.

[Gen](#) określa ile energii człowiek traci w każdym cyklu symulacji. Wartość genu reprezentuje procentowy ubytek energii na cykl:

- Niższe wartości: organizm oszczędniejszy (dłużej przeżyje bez jedzenia)
- Wyższe wartości: szybsze zużycie energii (większa potrzeba jedzenia)

Dziedziczenie odbywa się z asymetryczną mutacją (-5 do +7), przy czym minimalna wartość wynosi 5 (organizm nie może być całkowicie bierny).

6.7.2 Dokumentacja konstruktora i destruktora

6.7.2.1 `GenZapotrzebowanieEnergetyczne()` [1/2]

```
proj.selekcjanatur.GenZapotrzebowanieEnergetyczne.GenZapotrzebowanieEnergetyczne ()
```

Konstruktor losujący początkową wartość genu.

Generuje wartość zmiennoprzecinkową z przedziału 5-20 Oto graf wywołań tej funkcji:



6.7.2.2 `GenZapotrzebowanieEnergetyczne()` [2/2]

```
proj.selekcjanatur.GenZapotrzebowanieEnergetyczne.GenZapotrzebowanieEnergetyczne (
    float wartosc)
```

Konstruktor z określeniem konkretnej wartości.

Parametry

<code>wartosc</code>	Początkowa wartość genu (minimalna wartość to 5)
----------------------	--

6.7.3 Dokumentacja funkcji składowych

6.7.3.1 `odziedzicz()`

```
Gen proj.selekcjanatur.GenZapotrzebowanieEnergetyczne.odziedzicz ()
```

Tworzy nowy gen poprzez dziedziczenie z mutacją

Zwraca

Nowa instancja `GenZapotrzebowanieEnergetyczne`

Algorytm działania:

1. Losuje mutację z zakresu -5 do +7 (asymetrycznie)
2. Dodaje mutację do obecnej wartości
3. Gwarantuje minimalną wartość 5
4. Zwraca nowy gen z zmutowaną wartością

Nota

Mutacja jest celowo asymetryczna - większa szansa na zwiększenie metabolizmu niż na jego zmniejszenie, co symuluje ewolucyjną tendencję do zwiększania aktywności organizmów.

Implementuje [proj.selekcjanatur.Gen](#).

Oto graf wywołań dla tej funkcji:

**6.7.3.2 wartosc()**

```
float proj.selekcjanatur.GenZapotrzebowanieEnergetyczne.wartosc ()
```

Zwraca aktualną wartość genu.

Zwraca

Wartość zapotrzebowania energetycznego

Implementuje [proj.selekcjanatur.Gen](#).

6.7.4 Dokumentacja atrybutów składowych**6.7.4.1 wartosc**

```
final float proj.selekcjanatur.GenZapotrzebowanieEnergetyczne.wartosc [private]
```

Przechowuje wartość genu metabolizmu.

Wartość określa procentowy ubytek energii na cykl symulacji

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [src/main/java/proj/selekcjanatur/GenZapotrzebowanieEnergetyczne.java](#)

6.8 Dokumentacja interfejsu proj.selekcjanatur.InterfejsSymulacji

Główny interfejs systemu symulacji ewolucyjnej.

Diagram dziedziczenia dla proj.selekcjanatur.InterfejsSymulacji

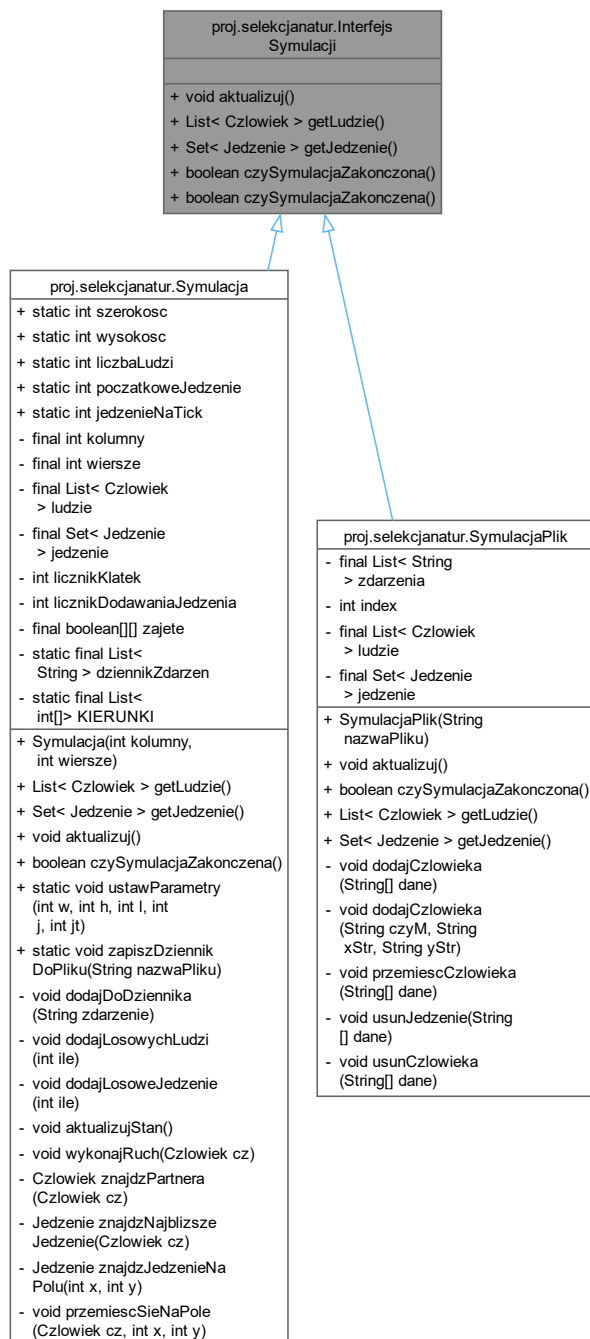
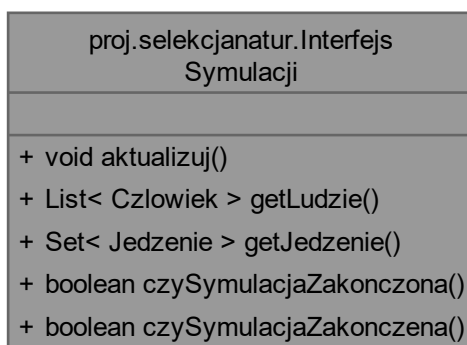


Diagram współpracy dla proj.selekcjanatur.InterfejsSymulacji:



Metody publiczne

- void [aktualizuj](#) ()
Aktualizuje stan symulacji o jeden krok.
- List< [Czlowiek](#) > [getLudzie](#) ()
Pobiera listę wszystkich ludzi w symulacji.
- Set< [Jedzenie](#) > [getJedzenie](#) ()
Pobiera zbiór dostępnego jedzenia.
- boolean [czySymulacjaZakonczone](#) ()
Sprawdza warunki zakończenia symulacji.
- boolean [czySymulacjaZakonczena](#) ()

6.8.1 Opis szczegółowy

Główny interfejs systemu symulacji ewolucyjnej.

Definiuje kontrakt dla wszystkich implementacji symulacji, zarówno działających w czasie rzeczywistym jak i odtwarzających z zapisanych danych. Interfejs jest implementowany przez:

- Klasę [Symulacja](#) (podstawowa implementacja)
- Klasę [SymulacjaPlik](#) (odtworzenie z dziennika zdarzeń)

6.8.2 Dokumentacja funkcji składowych

6.8.2.1 aktualizuj()

```
void proj.selekcjanatur.InterfejsSymulacji.aktualizuj ()
```

Aktualizuje stan symulacji o jeden krok.

Metoda powinna:

- Wykonać wszystkie obliczenia dla bieżącego cyklu
- Zaktualizować stan wszystkich obiektów
- Sprawdzić warunki zakończenia symulacji

Implementowany w [proj.selekcjanatur.Symulacja](#) i [proj.selekcjanatur.SymulacjaPlik](#).

6.8.2.2 czySymulacjaZakonczona()

```
boolean proj.selekcjanatur.InterfejsSymulacji.czySymulacjaZakonczona ()
```

Implementowany w [proj.selekcjanatur.Symulacja](#).

6.8.2.3 czySymulacjaZakonczona()

```
boolean proj.selekcjanatur.InterfejsSymulacji.czySymulacjaZakonczona ()
```

Sprawdza warunki zakończenia symulacji.

Zwraca

`true` jeśli symulacja powinna się zakończyć

Typowe warunki zakończenia:

- Wyczerpanie się populacji (brak żywych ludzi)
- Osiągnięcie maksymalnej liczby cykli
- Odtworzenie wszystkich zdarzeń z pliku (w trybie odtwarzania)

Implementowany w [proj.selekcjanatur.SymulacjaPlik](#).

6.8.2.4 getJedzenie()

```
Set< Jedzenie > proj.selekcjanatur.InterfejsSymulacji.getJedzenie ()
```

Pobiera zbiór dostępnego jedzenia.

Zwraca

Nieuporządkowany zbiór obiektów [Jedzenie](#)

Zwraca tylko jedzenie obecne w bieżącym cyklu symulacji. [Jedzenie](#) zjedzone w poprzednich cyklach nie jest uwzględniane.

Implementowany w [proj.selekcjanatur.Symulacja](#) i [proj.selekcjanatur.SymulacjaPlik](#).

6.8.2.5 getLudzie()

```
List< Czlowiek > proj.selekcjanatur.InterfejsSymulacji.getLudzie ()
```

Pobiera listę wszystkich ludzi w symulacji.

Zwraca

Lista obiektów [Czlowiek](#)

Zwraca zarówno żywych jak i martwych uczestników symulacji. Kolejność na liście może mieć znaczenie dla renderowania.

Implementowany w [proj.selekcjanatur.Symulacja](#) i [proj.selekcjanatur.SymulacjaPlik](#).

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- [src/main/java/proj/selekcjanatur/InterfejsSymulacji.java](#)

6.9 Dokumentacja klasy proj.selekcjanatur.Jedzenie

Klasa modelująca pożywienie w symulacji ewolucyjnej.

Diagram współpracy dla proj.selekcjanatur.Jedzenie:

proj.selekcjanatur.Jedzenie
+ int x
+ int y
+ int wartosc
+ Jedzenie(int x, int y)

Metody publiczne

- `Jedzenie` (int `x`, int `y`)
Konstruktor tworzący nowe pożywienie.

Atrybuty publiczne

- int `x`
Pozycja `x` na planszy.
- int `y`
Pozycja `y` na planszy.
- int `wartosc`
Wartość energetyczna pożywienia.

6.9.1 Opis szczegółowy

Klasa modelująca pożywienie w symulacji ewolucyjnej.

Reprezentuje pojedyncze źródło pożywienia na planszy symulacji. Określa pozycję (`x`,`y`) oraz wartość energetyczną, która jest losowana przy tworzeniu obiektu. Klasa jest niezmiennicza (immutable) poza modyfikacjami dokonywanymi przez symulację.

6.9.2 Dokumentacja konstruktora i destruktora**6.9.2.1 Jedzenie()**

```
proj.selekcjanatur.Jedzenie.Jedzenie (
    int x,
    int y)
```

Konstruktor tworzący nowe pożywienie.

Parametry

<code>x</code>	Pozycja pozioma na planszy
<code>y</code>	Pozycja pionowa na planszy

Wartość energetyczna jest losowana z przedziału 10-50. Pozycje powinny być zawarte w wymiarach planszy symulacji.

6.9.3 Dokumentacja atrybutów składowych**6.9.3.1 wartosc**

```
int proj.selekcjanatur.Jedzenie.wartosc
```

Wartość energetyczna pożywienia.

Losowana z przedziału 10-50 w momencie tworzenia obiektu. Im wyższa wartość, tym więcej energii zyskuje człowiek po zjedzeniu.

6.9.3.2 x

```
int proj.selekcjanatur.Jedzenie.x
```

Pozycja x na planszy.

Wartość z zakresu 0 do szerokości planszy - 1

6.9.3.3 y

```
int proj.selekcjanatur.Jedzenie.y
```

Pozycja y na planszy.

Wartość z zakresu 0 do wysokości planszy - 1

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [src/main/java/proj/selekcjanatur/Jedzenie.java](#)

6.10 Dokumentacja klasy proj.selekcjanatur.Kobieta

Specjalizacja klasy [Czlowiek](#) reprezentująca kobietę

Diagram dziedziczenia dla proj.selekcjanatur.Kobieta

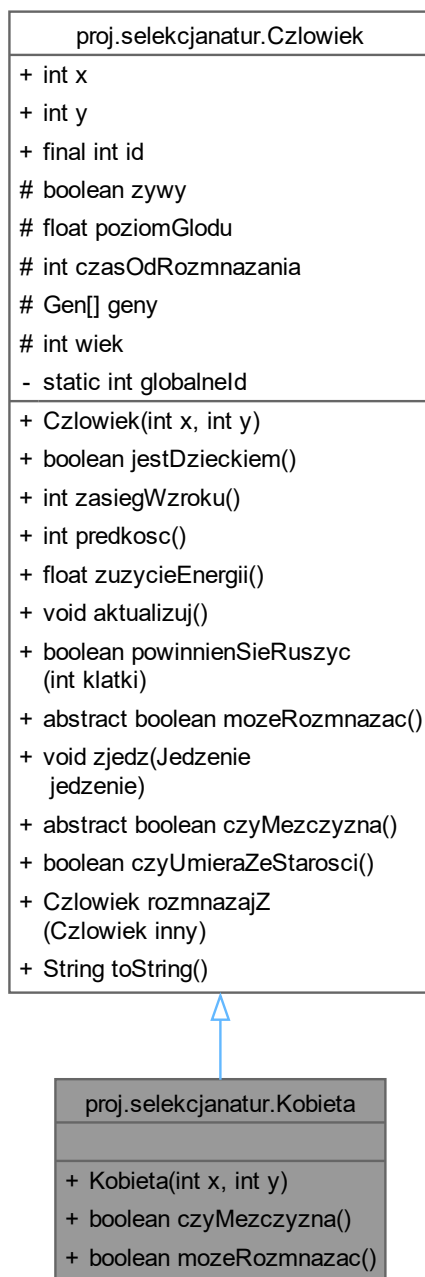
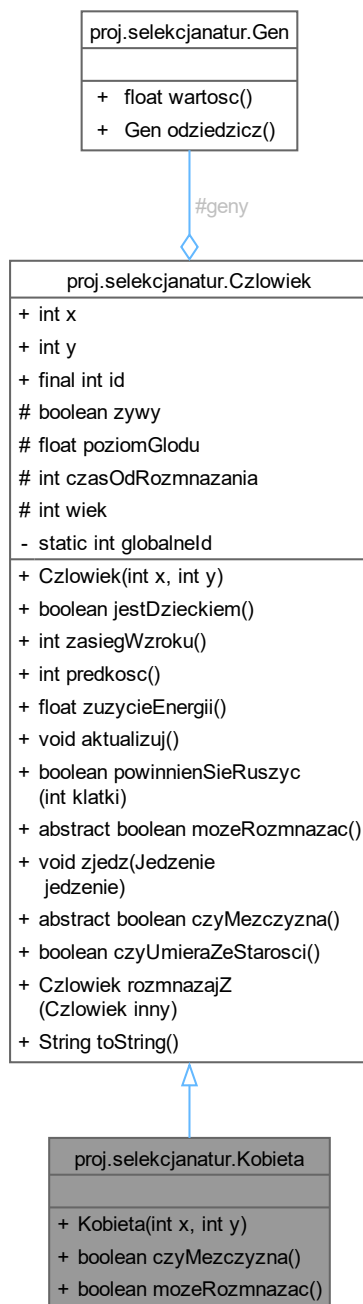


Diagram współpracy dla proj.selekcjanatur.Kobieta:



Metody publiczne

- **Kobieta** (int *x*, int *y*)
Konstruktor tworzący nową kobietę
- boolean **czyMezczyzna** ()
Sprawdza płeć człowieka.
- boolean **mozeRozmnazac** ()
Określa możliwość rozmnażania.

Metody publiczne dziedziczone z proj.selekcjanatur.Czlowiek

- **Czlowiek** (int x, int y)
Konstruktor tworzący nowego człowieka.
- boolean **jestDzieckiem** ()
Sprawdza czy człowiek jest dzieckiem.
- int **zasiegWzroku** ()
Oblicza zasięg wzroku na podstawie genów.
- int **predkosc** ()
Oblicza prędkość poruszania na podstawie genów.
- float **zuzycieEnergii** ()
Oblicza zużycie energii na cykl.
- void **aktualizuj** ()
Aktualizuje stan człowieka w każdym cyklu symulacji.
- boolean **powinnienSieRuszyć** (int klatki)
Decyduje czy człowiek powinien się poruszyć w danej klatce.
- void **zjedz** (Jedzenie jedzenie)
Zjada podane jedzenie.
- boolean **czyUmieraZeStarosci** ()
Sprawdza czy człowiek umiera ze starości.
- **Czlowiek rozmnazajZ** (Czlowiek inny)
Rozmnaża się z innym człowiekiem.
- String **toString** ()
Reprezentacja tekstowa człowieka.

Dodatkowe dziedziczone składowe

Atrybuty publiczne dziedziczone z proj.selekcjanatur.Czlowiek

- int x
Pozycja x na planszy.
- int y
Pozycja y na planszy.
- final int id
Unikalny identyfikator człowieka.

Atrybuty chronione dziedziczone z proj.selekcjanatur.Czlowiek

- boolean **zywy** = true
Flaga określająca czy człowiek jest żywy.
- float **poziomGlodu** = 100
Poziom głodu (0-100)
- int **czasOdRozmnazania** = 20
Liczba cykli od ostatniego rozmnożenia.
- **Gen[] geny** = new **Gen**[3]
Tablica genów człowieka (wzrok, prędkość, metabolizm)
- int **wiek** = 0
Wiek człowieka w cyklach symulacji.

6.10.1 Opis szczegółowy

Specjalizacja klasy [Czlowiek](#) reprezentująca kobietę

Klasa rozszerza podstawową funkcjonalność człowieka o:

- Specyficzne cechy reprodukcyjne
- Ograniczoną maksymalną prędkość chodzenia (1-2 zamiast 1-3)
- Logikę decydującą o możliwości rozmnażania

6.10.2 Dokumentacja konstruktora i destruktora

6.10.2.1 Kobieta()

```
proj.selekcjanatur.Kobieta.Kobieta (  
    int x,  
    int y)
```

Konstruktor tworzący nową kobietę

Parametry

x	Pozycja startowa x na planszy
y	Pozycja startowa y na planszy

Inicjalizuje kobietę z:

- Losową prędkością chodzenia z zakresu 1-2
- Standardowymi genami wzroku i metabolizmu
- Początkowym poziomem energii 100%

6.10.3 Dokumentacja funkcji składowych

6.10.3.1 czyMeczczyzna()

```
boolean proj.selekcjanatur.Kobieta.czyMeczczyzna ()
```

Sprawdza płeć człowieka.

Zwraca

Zawsze false (obiekt reprezentuje kobietę)

Reimplementowana z [proj.selekcjanatur.Czlowiek](#).

6.10.3.2 mozeRozmnazac()

```
boolean proj.selekcjanatur.Kobieta.mozeRozmnazac ()
```

Określa możliwość rozmnażania.

Zwraca

true jeśli spełnione są wszystkie warunki:

- Nie jest dzieckiem (wiek 18 cykli)
- Poziom głodu > 70%
- Od ostatniego rozmnażania minęło > 20 cykli

Kobiety mają bardziej restrykcyjne wymagania energetyczne do rozmnażania niż mężczyźni (70% vs 50% energii).

Reimplementowana z [proj.selekcjanatur.Czlowiek](#).

Oto graf wywołań dla tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z pliku:

- [src/main/java/proj/selekcjanatur/Kobieta.java](#)

6.11 Dokumentacja klasy proj.selekcjanatur.Mezczyzna

Specjalizacja klasy [Czlowiek](#) reprezentująca mężczyznę

Diagram dziedziczenia dla proj.selekcjanatur.Mezczyzna

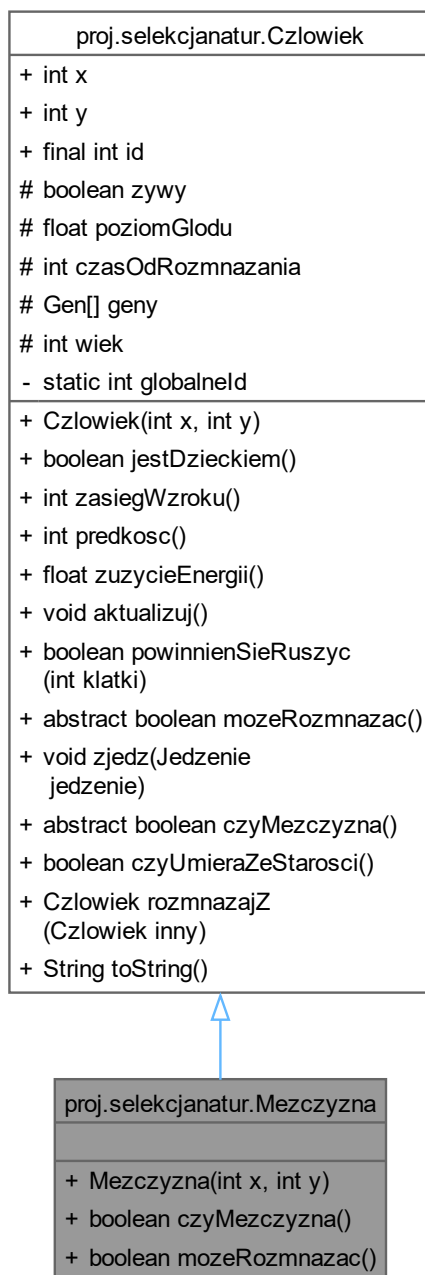
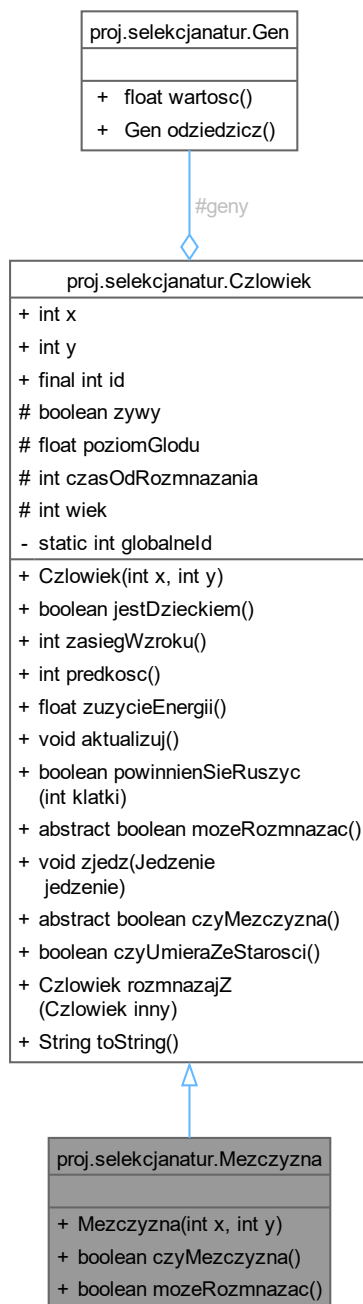


Diagram współpracy dla proj.selekcjanatur.Mezczyzna:



Metody publiczne

- **Mezczyzna** (int x, int y)
Konstruktor tworzący nowego mężczyznę
- boolean **czyMezczyzna** ()
Sprawdza płeć człowieka.
- boolean **mozeRozmnazac** ()
Określa możliwość rozmnażania.

Metody publiczne dziedziczone z [proj.selekcjanatur.Czlowiek](#)

- [Czlowiek](#) (int *x*, int *y*)
Konstruktor tworzący nowego człowieka.
- boolean [jestDzieckiem](#) ()
Sprawdza czy człowiek jest dzieckiem.
- int [zasiegWzroku](#) ()
Oblicza zasięg wzroku na podstawie genów.
- int [predkosc](#) ()
Oblicza prędkość poruszania na podstawie genów.
- float [zuzycieEnergii](#) ()
Oblicza zużycie energii na cykl.
- void [aktualizuj](#) ()
Aktualizuje stan człowieka w każdym cyklu symulacji.
- boolean [powinnienSieRuszy](#) (int *klatki*)
Decyduje czy człowiek powinien się poruszyć w danej klatce.
- void [zjedz](#) ([Jedzenie](#) *jedzenie*)
Zjada podane jedzenie.
- boolean [czyUmieraZeStarosci](#) ()
Sprawdza czy człowiek umiera ze starości.
- [Czlowiek rozmnazajZ](#) ([Czlowiek](#) *inny*)
Rozmnaża się z innym człowiekiem.
- String [toString](#) ()
Reprezentacja tekstowa człowieka.

Dodatkowe dziedziczone składowe

Atrybuty publiczne dziedziczone z [proj.selekcjanatur.Czlowiek](#)

- int *x*
Pozycja x na planszy.
- int *y*
Pozycja y na planszy.
- final int *id*
Unikalny identyfikator człowieka.

Atrybuty chronione dziedziczone z [proj.selekcjanatur.Czlowiek](#)

- boolean *zywy* = true
Flaga określająca czy człowiek jest żywy.
- float *poziomGlodu* = 100
Poziom głodu (0-100)
- int *czasOdRozmnazania* = 20
Liczba cykli od ostatniego rozmnożenia.
- [Gen](#)[] *geny* = new [Gen](#)[3]
Tablica genów człowieka (wzrok, prędkość, metabolizm)
- int *wiek* = 0
Wiek człowieka w cyklach symulacji.

6.11.1 Opis szczegółowy

Specjalizacja klasy [Czlowiek](#) reprezentująca mężczyznę

Klasa rozszerza podstawową funkcjonalność człowieka o:

- Męską płeć (zawsze zwraca true w [czyMezczyzna\(\)](#))
- Specyficzne warunki reprodukcyjne (mniejsze wymagania niż kobieta)
- Standardową prędkość chodzenia (pełen zakres 1-3)

6.11.2 Dokumentacja konstruktora i destruktor

6.11.2.1 Mezczyzna()

```
proj.selekcjanatur.Mezczyzna.Mezczyzna (  
    int x,  
    int y)
```

Konstruktor tworzący nowego mężczyznę

Parametry

x	Pozycja startowa x na planszy
y	Pozycja startowa y na planszy

Inicjalizuje mężczyznę z:

- Standardowymi genami (wzrok, prędkość, metabolizm)
- Pełnym zakresem możliwej prędkości (1-3)
- Początkowym poziomem energii 100%

6.11.3 Dokumentacja funkcji składowych

6.11.3.1 czyMezczyzna()

```
boolean proj.selekcjanatur.Mezczyzna.czyMezczyzna ()
```

Sprawdza płeć człowieka.

Zwraca

Zawsze true (obiekt reprezentuje mężczyznę)

Reimplementowana z [proj.selekcjanatur.Czlowiek](#).

6.11.3.2 mozeRozmnazac()

```
boolean proj.selekcjanatur.Mezczyzna.mozeRozmnazac ()
```

Określa możliwość rozmnażania.

Zwraca

true jeśli spełnione są wszystkie warunki:

- Nie jest dzieckiem (wiek 18 cykli)
- Poziom głodu > 60%
- Od ostatniego rozmnażania minęło > 10 cykli

Mężczyźni mają mniejsze wymagania energetyczne do rozmnażania niż kobiety (60% vs 70% energii) i krótszy wymagany czas między rozmnażaniem (10 vs 20 cykli).

Reimplementowana z [proj.selekcjanatur.Czlowiek](#).

Oto graf wywołań dla tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z pliku:

- [src/main/java/proj/selekcjanatur/Mezczyzna.java](#)

6.12 Dokumentacja klasy proj.selekcjanatur.StartController

Kontroler GUI dla ekranu konfiguracji symulacji.

Diagram współpracy dla proj.selekcjanatur.StartController:

proj.selekcjanatur.Start Controller
<ul style="list-style-type: none"> - Slider sliderWidth - Slider sliderHeight - Slider sliderPeople - Slider sliderFood - Slider sliderFoodPerTick - Label labelWidth - Label labelHeight - Label labelPeople - Label labelFood - Label labelFoodPerTick
<ul style="list-style-type: none"> + void initialize() - void bindSliderLabel (Slider slider, Label label) - void startSimulation() - void replayFromFile()

Metody publiczne

- void [initialize](#) ()
Inicjalizacja kontrolera.

Metody prywatne

- void [bindSliderLabel](#) (Slider slider, Label label)
Wiąże suwak z etykietą wyświetlającą jego wartość
- void [startSimulation](#) ()
Uruchamia nową symulację z wybranymi parametrami.
- void [replayFromFile](#) ()
Odtwarza symulację z pliku dziennika zdarzeń

Atrybuty prywatne

- Slider [sliderWidth](#)
Suwak do ustawiania szerokości planszy.
- Slider [sliderHeight](#)

- Suwak do ustawiania wysokości planszy.
- Slider [sliderPeople](#)
Suwak do ustawiania liczby ludzi.
- Slider [sliderFood](#)
Suwak do ustawiania początkowej ilości jedzenia.
- Slider [sliderFoodPerTick](#)
Suwak do ustawiania ilości nowego jedzenia na cykl.
- Label [labelWidth](#)
Etykieta wyświetlająca wartość szerokości.
- Label [labelHeight](#)
Etykieta wyświetlająca wartość wysokości.
- Label [labelPeople](#)
Etykieta wyświetlająca liczbę ludzi.
- Label [labelFood](#)
Etykieta wyświetlająca ilość jedzenia.
- Label [labelFoodPerTick](#)
Etykieta wyświetlająca ilość jedzenia na cykl.

6.12.1 Opis szczegółowy

Kontroler GUI dla ekranu konfiguracji symulacji.

Klasa odpowiada za:

- Obsługę interfejsu użytkownika ekranu startowego
- Walidację parametrów symulacji
- Uruchamianie nowej symulacji
- Odtwarzanie symulacji z pliku dziennika zdarzeń

6.12.2 Dokumentacja funkcji składowych

6.12.2.1 bindSliderLabel()

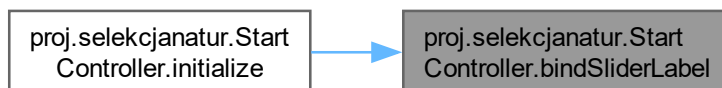
```
void proj.selekcjanatur.StartController.bindSliderLabel (
    Slider slider,
    Label label) [private]
```

Wiąże suwak z etykietą wyświetlającą jego wartość

Parametry

<i>slider</i>	Suwak do powiązania
<i>label</i>	Etykieta wyświetlająca wartość

Oto graf wywołań tej funkcji:



6.12.2.2 initialize()

```
void proj.selekcjanatur.StartController.initialize ()
```

Inicjalizacja kontrolera.

Wiąże suwaki z odpowiadającymi im etykietami, aby wartości były wyświetlane w czasie rzeczywistym. Oto graf wywołań dla tej funkcji:



6.12.2.3 replayFromFile()

```
void proj.selekcjanatur.StartController.replayFromFile () [private]
```

Odtwarza symulację z pliku dziennika zdarzeń

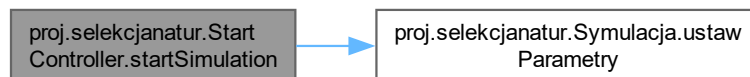
Wczytuje podstawowe parametry symulacji z pliku i przełącza na widok symulacji w trybie odtwarzania. Wyświetla komunikaty błędów jeśli plik jest nieprawidłowy.

6.12.2.4 startSimulation()

```
void proj.selekcjanatur.StartController.startSimulation () [private]
```

Uruchamia nową symulację z wybranymi parametrami.

Sprawdza poprawność parametrów i uruchamia symulację. Wyświetla komunikat błędu jeśli liczba obiektów przekracza liczbę dostępnych pól na planszy. Oto graf wywołań dla tej funkcji:



6.12.3 Dokumentacja atrybutów składowych

6.12.3.1 labelFood

```
Label proj.selekcjanatur.StartController.labelFood [private]
```

Etykieta wyświetlająca ilość jedzenia.

6.12.3.2 labelFoodPerTick

```
Label proj.selekcjanatur.StartController.labelFoodPerTick [private]
```

Etykieta wyświetlająca ilość jedzenia na cykl.

6.12.3.3 labelHeight

```
Label proj.selekcjanatur.StartController.labelHeight [private]
```

Etykieta wyświetlająca wartość wysokości.

6.12.3.4 labelPeople

```
Label proj.selekcjanatur.StartController.labelPeople [private]
```

Etykieta wyświetlająca liczbę ludzi.

6.12.3.5 labelWidth

Label proj.selekcjanatur.StartController.labelWidth [private]

Etykieta wyświetlająca wartość szerokości.

6.12.3.6 sliderFood

Slider proj.selekcjanatur.StartController.sliderFood [private]

Suwak do ustawiania początkowej ilości jedzenia.

6.12.3.7 sliderFoodPerTick

Slider proj.selekcjanatur.StartController.sliderFoodPerTick [private]

Suwak do ustawiania ilości nowego jedzenia na cykl.

6.12.3.8 sliderHeight

Slider proj.selekcjanatur.StartController.sliderHeight [private]

Suwak do ustawiania wysokości planszy.

6.12.3.9 sliderPeople

Slider proj.selekcjanatur.StartController.sliderPeople [private]

Suwak do ustawiania liczby ludzi.

6.12.3.10 sliderWidth

Slider proj.selekcjanatur.StartController.sliderWidth [private]

Suwak do ustawiania szerokości planszy.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- src/main/java/proj/selekcjanatur/[StartController.java](#)

6.13 Dokumentacja klasy proj.selekcjanatur.Symulacja

Implementacja symulacji ewolucyjnej z dziedziczeniem cech.

Diagram dziedziczenia dla proj.selekcjanatur.Symulacja

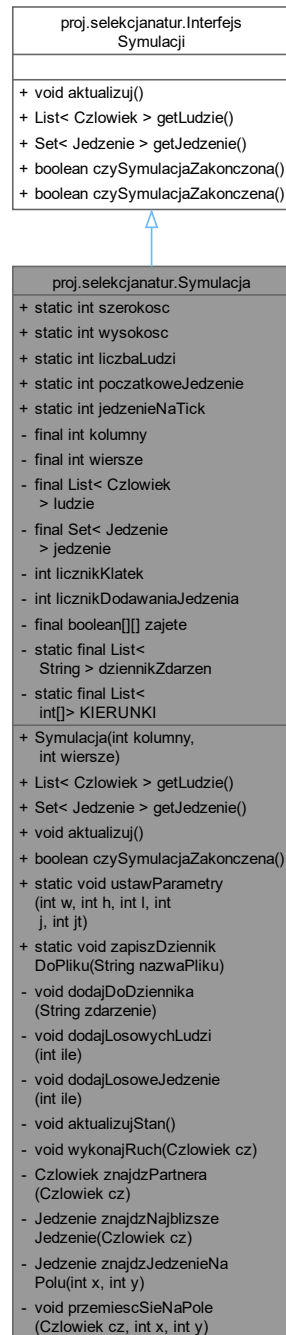
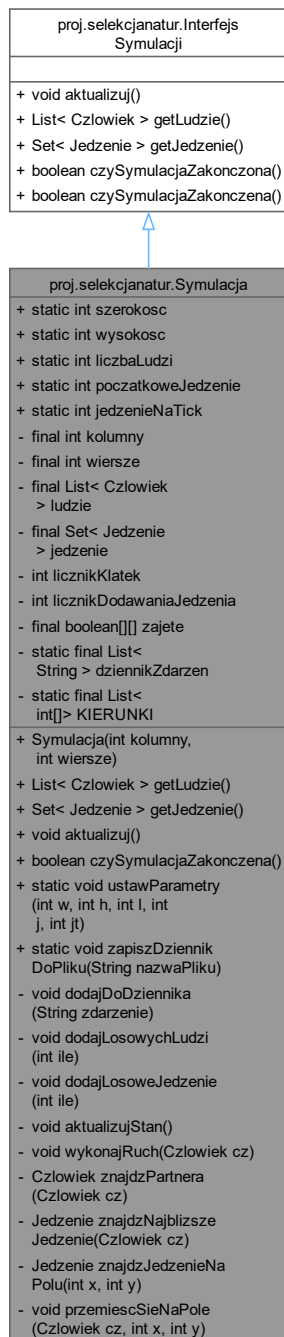


Diagram współpracy dla proj.selekcjanatur.Symulacja:



Metody publiczne

- **Symulacja** (int kolumny, int wiersze)
Konstruktor inicjalizujący symulację
- List< **Czlowiek** > **getLudzie** ()
Zwraca listę ludzi w symulacji.
- Set< **Jedzenie** > **getJedzenie** ()

- Zwraca zbiór jedzenia w symulacji.*
- void [aktualizuj](#) ()
Aktualizuje stan symulacji o jeden krok.
- boolean [czySymulacjaZakonczona](#) ()
Sprawdza czy symulacja powinna się zakończyć

Metody publiczne dziedziczone z [proj.selekcjanatur.InterfejsSymulacji](#)

- boolean [czySymulacjaZakonczona](#) ()
Sprawdza warunki zakończenia symulacji.

Statyczne metody publiczne

- static void [ustawParametry](#) (int w, int h, int l, int j, int jt)
Ustawia globalne parametry symulacji.
- static void [zapiszDziennikDoPliku](#) (String nazwaPliku)
Zapisuje dziennik zdarzeń do pliku.

Statyczne atrybuty publiczne

- static int [szerokosc](#)
Statyczne parametry symulacji - szerokość planszy.
- static int [wysokosc](#)
Statyczne parametry symulacji - wysokość planszy.
- static int [liczbaLudzi](#)
Statyczne parametry symulacji - początkowa liczba ludzi.
- static int [początkoweJedzenie](#)
Statyczne parametry symulacji - początkowa ilość jedzenia.
- static int [jedzenieNaTick](#)
Statyczne parametry symulacji - nowe jedzenie na cykl.

Metody prywatne

- void [dodajDoDziennika](#) (String zdarzenie)
Dodaje wpis do dziennika zdarzeń
- void [dodajLosowychLudzi](#) (int ile)
Dodaje losowych ludzi na planszę
- void [dodajLosoweJedzenie](#) (int ile)
Dodaje losowe jedzenie na planszę
- void [aktualizujStan](#) ()
Aktualizuje stan symulacji (co 3 klatki)
- void [wykonajRuch](#) (Człowiek cz)
Wykonuje ruch dla danego człowieka.
- [Człowiek](#) [znajdzPartnera](#) (Człowiek cz)
Znajduje partnera do rozmnażania w zasięgu wzroku.
- [Jedzenie](#) [znajdzNajblizszeJedzenie](#) (Człowiek cz)
Znajduje najbliższe jedzenie w zasięgu wzroku.
- [Jedzenie](#) [znajdzJedzenieNaPolu](#) (int x, int y)
Sprawdza czy na danym polu jest jedzenie.
- void [przemiescSieNaPole](#) (Człowiek cz, int x, int y)
Przenosi człowieka na nowe pole.

Atrybuty prywatne

- final int `kolumny`
Liczba kolumn planszy (szerokość)
- final int `wiersze`
Liczba wierszy planszy (wysokość)
- final List< Czlowiek > `ludzie` = new ArrayList<>()
Lista wszystkich ludzi w symulacji.
- final Set< Jedzenie > `jedzenie` = new HashSet<>()
Zbiór dostępnego jedzenia na planszy.
- int `licznikKlatek` = 0
Licznik klatek symulacji.
- int `licznikDodawaniaJedzenia` = 0
Licznik kontrolujący dodawanie nowego jedzenia.
- final boolean[][] `zajete`
Mapa zajętości pól na planszy.

Statyczne atrybuty prywatne

- static final List< String > `dziennikZdarzen` = new ArrayList<>()
Dziennik zdarzeń zapisywany do pliku.
- static final List< int[] > `KIERUNKI`
Lista możliwych kierunków ruchu.

6.13.1 Opis szczegółowy

Implementacja symulacji ewolucyjnej z dziedziczeniem cech.

Klasa zarządza całym przebiegiem symulacji, w tym:

- Inicjalizacją populacji i środowiska
- Mechaniką rozmnażania i dziedziczenia cech
- Ruchem i interakcjami między organizmami
- Systemem dziennika zdarzeń
- Warunkami zakończenia symulacji

6.13.2 Dokumentacja konstruktora i destruktora

6.13.2.1 Symulacja()

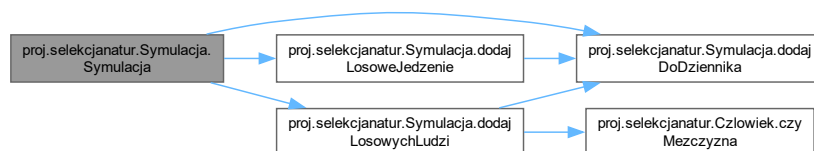
```
proj.selekcjanatur.Symulacja.Symulacja (  
    int kolumny,  
    int wiersze)
```

Konstruktor inicjalizujący symulację

Parametry

<i>kolumny</i>	Szerokość planszy
<i>wiersze</i>	Wysokość planszy

Oto graf wywołań dla tej funkcji:



6.13.3 Dokumentacja funkcji składowych

6.13.3.1 aktualizuj()

```
void proj.selekcjanatur.Symulacja.aktualizuj ()
```

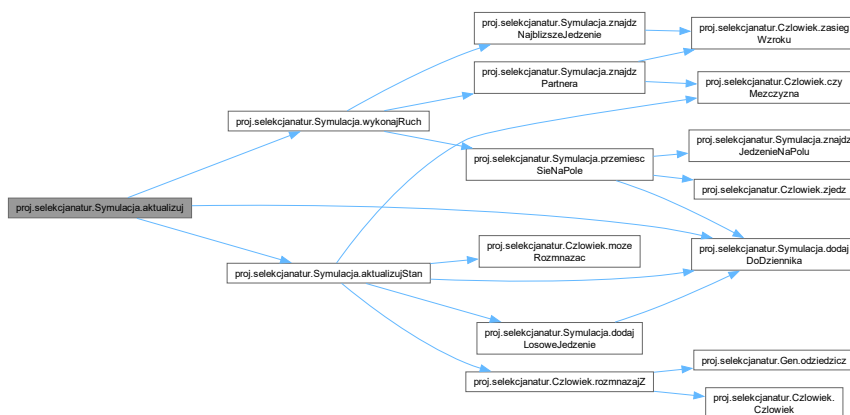
Aktualizuje stan symulacji o jeden krok.

Wykonuje:

- Aktualizację stanu co 3 klatki
- Ruch ludzi zgodnie z ich predyspozycjami
- Dodawanie nowego jedzenia
- Rozmnażanie i usuwanie martwych

Implementuje [proj.selekcjanatur.InterfejsSymulacji](#).

Oto graf wywołań dla tej funkcji:



6.13.3.2 aktualizujStan()

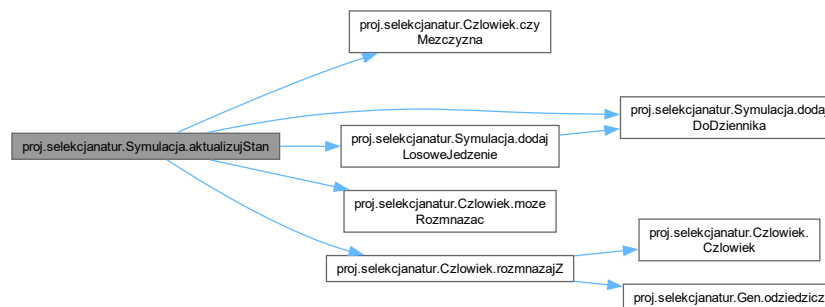
```
void proj.selekcjanatur.Symulacja.aktualizujStan () [private]
```

Aktualizuje stan symulacji (co 3 klatki)

Wykonuje:

- Dodawanie nowego jedzenia
- Aktualizację stanu ludzi
- Rozmnażanie i usuwanie martwych

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.13.3.3 czySymulacjaZakonczona()

```
boolean proj.selekcjanatur.Symulacja.czySymulacjaZakonczona ()
```

Sprawdza czy symulacja powinna się zakończyć

Zwraca

true jeśli nie ma żywych ludzi

Implementuje [proj.selekcjanatur.InterfejsSymulacji](#).

6.13.3.4 dodajDoDziennika()

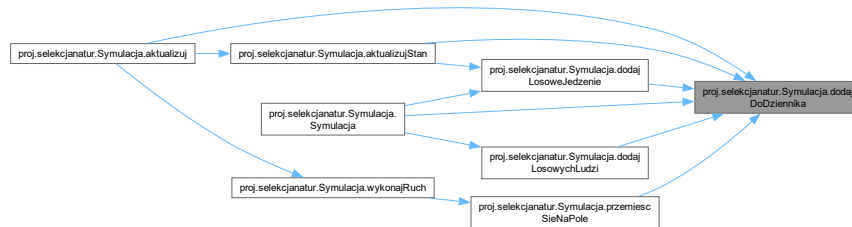
```
void proj.selekcjanatur.Symulacja.dodajDoDziennika (
    String zdarzenie) [private]
```

Dodaje wpis do dziennika zdarzeń

Parametry

	zdarzenie	Tekstowy opis zdarzenia
--	------------------	-------------------------

Oto graf wywołań tej funkcji:



6.13.3.5 dodajLosoweJedzenie()

```
void proj.selekcjanatur.Symulacja.dodajLosoweJedzenie (
    int ile) [private]
```

Dodaje losowe jedzenie na planszę

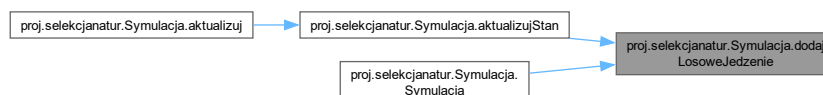
Parametry

	ile	Liczba jedzenia do dodania
--	------------	----------------------------

Oto graf wywołań dla tej funkcji:



Oto graf wywołań tej funkcji:



6.13.3.6 dodajLosowychLudzi()

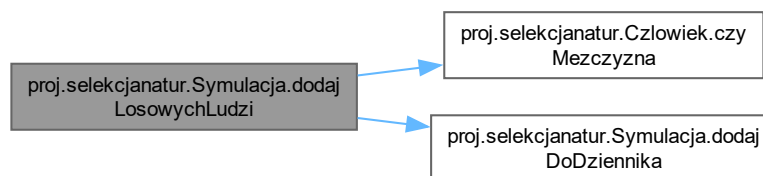
```
void proj.selekcjanatur.Symulacja.dodajLosowychLudzi (
    int ile) [private]
```

Dodaje losowych ludzi na planszę

Parametry

<code>ile</code>	Liczba ludzi do dodania
------------------	-------------------------

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

6.13.3.7 `getJedzenie()`

```
Set< Jedzenie > proj.selekcjanatur.Symulacja.getJedzenie ()
```

Zwraca zbiór jedzenia w symulacji.

Zwraca

Zbiór obiektów `Jedzenie`

Implementuje [proj.selekcjanatur.InterfejsSymulacji](#).

6.13.3.8 `getLudzie()`

```
List< Czlowiek > proj.selekcjanatur.Symulacja.getLudzie ()
```

Zwraca listę ludzi w symulacji.

Zwraca

Lista obiektów `Czlowiek`

Implementuje [proj.selekcjanatur.InterfejsSymulacji](#).

6.13.3.9 przemiescSieNaPole()

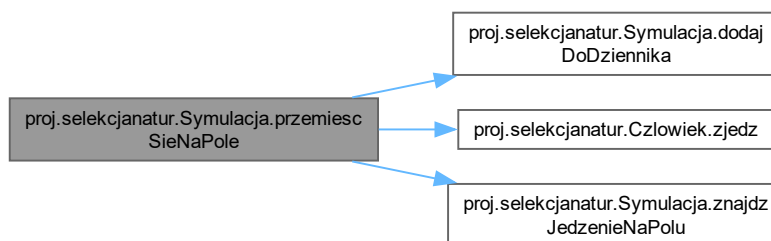
```
void proj.selekcjanatur.Symulacja.przemiescSieNaPole (
    Czlowiek cz,
    int x,
    int y) [private]
```

Przenosi człowieka na nowe pole.

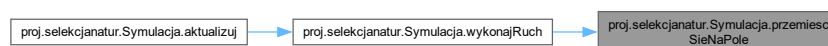
Parametry

cz	Człowiek do przeniesienia
x	Nowa pozycja x
y	Nowa pozycja y

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.13.3.10 ustawParametry()

```
void proj.selekcjanatur.Symulacja.ustawParametry (
    int w,
    int h,
    int l,
    int j,
    int jt) [static]
```

Ustawia globalne parametry symulacji.

Parametry

w	Szerokość planszy
h	Wysokość planszy
l	Początkowa liczba ludzi
j	Początkowa ilość jedzenia
j/t	Ilość nowego jedzenia dodawanego co cykl

Oto graf wywołań tej funkcji:



6.13.3.11 wykonajRuch()

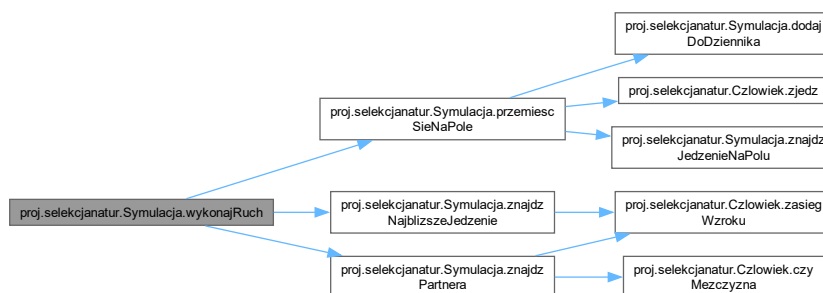
```
void proj.selekcjanatur.Symulacja.wykonajRuch (
    Czlowiek cz) [private]
```

Wykonuje ruch dla danego człowieka.

Parametry

cz	Człowiek, który ma się poruszyć
------	---------------------------------

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.13.3.12 zapiszDziennikDoPliku()

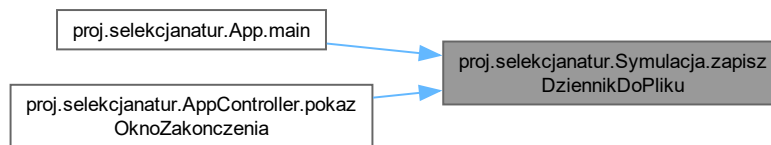
```
void proj.selekcjanatur.Symulacja.zapiszDziennikDoPliku (
    String nazwaPliku) [static]
```

Zapisuje dziennik zdarzeń do pliku.

Parametry

<code>nazwaPliku</code>	Nazwa pliku do zapisu
-------------------------	-----------------------

Oto graf wywoływań tej funkcji:



6.13.3.13 znajdzJedzenieNaPolu()

```
Jedzenie proj.selekcjanatur.Symulacja.znajdzJedzenieNaPolu (
    int x,
    int y) [private]
```

Sprawdza czy na danym polu jest jedzenie.

Parametry

<code>x</code>	Pozycja x
<code>y</code>	Pozycja y

Zwraca

Obiekt jedzenia lub null

Oto graf wywoływań tej funkcji:



6.13.3.14 znajdzNajblizszeJedzenie()

```
Jedzenie proj.selekcjanatur.Symulacja.znajdzNajblizszeJedzenie (
    Czlowiek cz) [private]
```

Znajduje najbliższe jedzenie w zasięgu wzroku.

Parametry

cz	Człowiek szukający jedzenia
----	-----------------------------

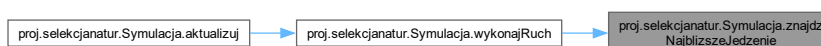
Zwraca

Znalezione jedzenie lub null

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.13.3.15 znajdźPartnera()

```
Człowiek proj.selekcjanatur.Symulacja.znajdźPartnera (
    Człowiek cz) [private]
```

Znajduje partnera do rozmnażania w zasięgu wzroku.

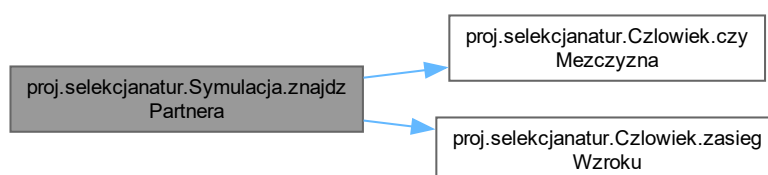
Parametry

cz	Człowiek szukający partnera
----	-----------------------------

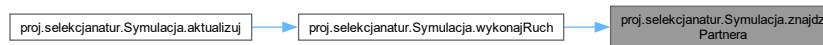
Zwraca

Znaleziony partner lub null

Oto graf wywołań dla tej funkcji:



Oto graf wywołań tej funkcji:



6.13.4 Dokumentacja atrybutów składowych

6.13.4.1 dziennikZdarzen

```
final List<String> proj.selekcjanatur.Symulacja.dziennikZdarzen = new ArrayList<>() [static], [private]
```

Dziennik zdarzeń zapisywany do pliku.

6.13.4.2 jedzenie

```
final Set<Jedzenie> proj.selekcjanatur.Symulacja.jedzenie = new HashSet<>() [private]
```

Zbiór dostępnego jedzenia na planszy.

6.13.4.3 jedzenieNaTick

```
int proj.selekcjanatur.Symulacja.jedzenieNaTick [static]
```

Statyczne parametry symulacji - nowe jedzenie na cykl.

6.13.4.4 KIERUNKI

```
final List<int[]> proj.selekcjanatur.Symulacja.KIERUNKI [static], [private]
```

Wartość początkowa:

```
= new ArrayList<>(Arrays.asList(
    new int[]{1, 0},
    new int[]{-1, 0},
    new int[]{0, 1},
    new int[]{0, -1},
    new int[]{1, 1},
    new int[]{-1, -1},
    new int[]{1, -1},
    new int[]{-1, 1}
))
```

Lista możliwych kierunków ruchu.

6.13.4.5 kolumny

```
final int proj.selekcjanatur.Symulacja.kolumny [private]
```

Liczba kolumn planszy (szerokość)

6.13.4.6 liczbaLudzi

```
int proj.selekcjanatur.Symulacja.liczbaLudzi [static]
```

Statyczne parametry symulacji - początkowa liczba ludzi.

6.13.4.7 licznikDodawaniaJedzenia

```
int proj.selekcjanatur.Symulacja.licznikDodawaniaJedzenia = 0 [private]
```

Licznik kontrolujący dodawanie nowego jedzenia.

6.13.4.8 licznikKlatek

```
int proj.selekcjanatur.Symulacja.licznikKlatek = 0 [private]
```

Licznik klatek symulacji.

6.13.4.9 ludzie

```
final List<Czlowiek> proj.selekcjanatur.Symulacja.ludzie = new ArrayList<>() [private]
```

Lista wszystkich ludzi w symulacji.

6.13.4.10 poczatkoweJedzenie

```
int proj.selekcjanatur.Symulacja.poczatkoweJedzenie [static]
```

Statyczne parametry symulacji - początkowa ilość jedzenia.

6.13.4.11 szerokosc

```
int proj.selekcjanatur.Symulacja.szerokosc [static]
```

Statyczne parametry symulacji - szerokość planszy.

6.13.4.12 wiersze

```
final int proj.selekcjanatur.Symulacja.wiersze [private]
```

Liczba wierszy planszy (wysokość)

6.13.4.13 wysokosc

```
int proj.selekcjanatur.Symulacja.wysokosc [static]
```

Statyczne parametry symulacji - wysokość planszy.

6.13.4.14 zajete

```
final boolean [][] proj.selekcjanatur.Symulacja.zajete [private]
```

Mapa zajętości pól na planszy.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [src/main/java/proj/selekcjanatur/Symulacja.java](#)

6.14 Dokumentacja klasy proj.selekcjanatur.SymulacjaPlik

Klasa odtwarzająca symulację na podstawie zapisanego dziennika zdarzeń

Diagram dziedziczenia dla proj.selekcjanatur.SymulacjaPlik

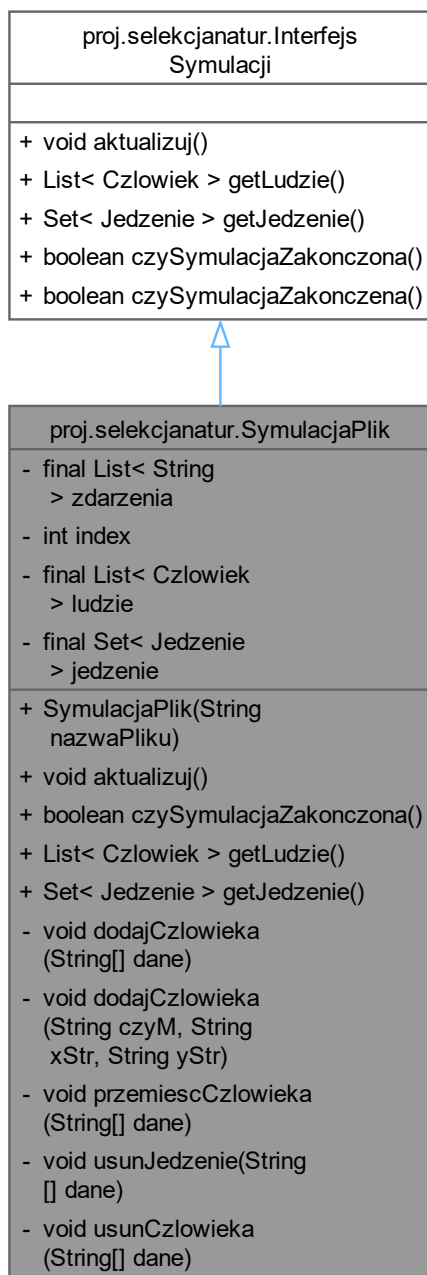
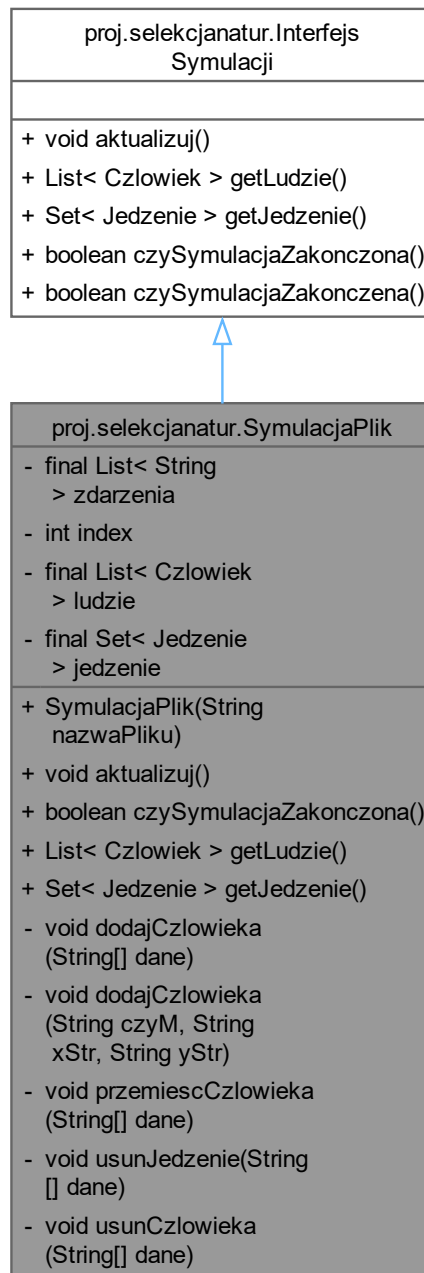


Diagram współpracy dla proj.selekcjanatur.SymulacjaPlik:



Metody publiczne

- [SymulacjaPlik](#) (String nazwaPliku) throws IOException
Konstruktor wczytujący dziennik zdarzeń z pliku.
- void [aktualizuj](#) ()
Aktualizuje stan symulacji o jeden krok.
- boolean [czySymulacjaZakonczone](#) ()

- *Sprawdza czy symulacja została w pełni odtworzona.*
- List< [Człowiek](#) > [getLudzie](#) ()
Zwraca listę ludzi w odtwarzanej symulacji.
- Set< [Jedzenie](#) > [getJedzenie](#) ()
Zwraca zbiór jedzenia w odtwarzanej symulacji.

Metody publiczne dziedziczone z [proj.selekcjanatur.InterfejsSymulacji](#)

- boolean [czySymulacjaZakonczona](#) ()

Metody prywatne

- void [dodajCzlowieka](#) (String[] dane)
Dodaje człowieka na podstawie danych z dziennika.
- void [dodajCzlowieka](#) (String czyM, String xStr, String yStr)
Dodaje dziecko powstałe w wyniku rozmnożenia.
- void [przemiescCzlowieka](#) (String[] dane)
Przenosi człowieka na nową pozycję
- void [usunJedzenie](#) (String[] dane)
Usuwa zjedzone jedzenie.
- void [usunCzlowieka](#) (String[] dane)
Usuwa martwego człowieka.

Atrybuty prywatne

- final List< String > [zdarzenia](#)
Lista wczytanych zdarzeń z pliku.
- int [index](#) = 0
Aktualny indeks przetwarzanego zdarzenia.
- final List< [Człowiek](#) > [ludzie](#) = new ArrayList<>()
Lista ludzi w odtwarzanej symulacji.
- final Set< [Jedzenie](#) > [jedzenie](#) = new HashSet<>()
Zbiór jedzenia w odtwarzanej symulacji.

6.14.1 Opis szczegółowy

Klasa odtwarzająca symulację na podstawie zapisanego dziennika zdarzeń

Implementuje ten sam interfejs co [Symulacja](#), ale zamiast obliczać nowe stany, odtwarza je sekwencyjnie z pliku. Pozwala to na odtworzenie przebiegu wcześniejszej symulacji krok po kroku.

6.14.2 Dokumentacja konstruktora i destruktor

6.14.2.1 SymulacjaPlik()

```
proj.selekcjanatur.SymulacjaPlik.SymulacjaPlik (
    String nazwaPliku) throws IOException
```

Konstruktor wczytujący dziennik zdarzeń z pliku.

Parametry

<code>nazwaPliku</code>	Ścieżka do pliku dziennika zdarzeń
-------------------------	------------------------------------

Wyjątki

<code>IOException</code>	Jeśli wystąpi błąd odczytu pliku
--------------------------	----------------------------------

6.14.3 Dokumentacja funkcji składowych**6.14.3.1 aktualizuj()**

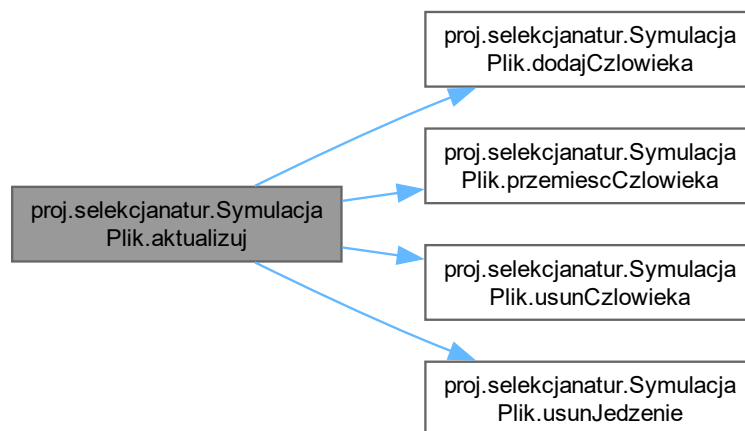
```
void proj.selekcjanatur.SymulacjaPlik.aktualizuj ()
```

Aktualizuje stan symulacji o jeden krok.

Przetwarza kolejne zdarzenia z dziennika aż do napotkania zdarzenia KLATKA, które oznacza koniec aktualnej klatki symulacji.

Implementuje [proj.selekcjanatur.InterfejsSymulacji](#).

Oto graf wywołań dla tej funkcji:

**6.14.3.2 czySymulacjaZakonczona()**

```
boolean proj.selekcjanatur.SymulacjaPlik.czySymulacjaZakonczona ()
```

Sprawdza czy symulacja została w pełni odtworzona.

Zwraca

true jeśli wszystkie zdarzenia zostały przetworzone

Implementuje [proj.selekcjanatur.InterfejsSymulacji](#).

6.14.3.3 dodajCzlowieka() [1/2]

```
void proj.selekcjanatur.SymulacjaPlik.dodajCzlowieka (
    String czyM,
    String xStr,
    String yStr) [private]
```

Dodaje dziecko powstałe w wyniku rozmnożenia.

Parametry

<code>czyM</code>	Płeć dziecka (true = męczyzna)
<code>xStr</code>	Pozycja x dziecka
<code>yStr</code>	Pozycja y dziecka

6.14.3.4 dodajCzlowieka() [2/2]

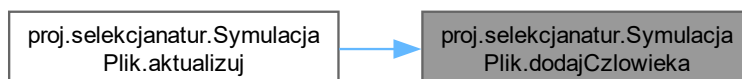
```
void proj.selekcjanatur.SymulacjaPlik.dodajCzlowieka (
    String[] dane) [private]
```

Dodaje człowieka na podstawie danych z dziennika.

Parametry

<code>dane</code>	Tablica parametrów zdarzenia DODANIE_CZLOWIEKA
-------------------	--

Oto graf wywoływań tej funkcji:

**6.14.3.5 getJedzenie()**

```
Set< Jedzenie > proj.selekcjanatur.SymulacjaPlik.getJedzenie ()
```

Zwraca zbiór jedzenia w odtwarzanej symulacji.

Zwraca

Zbiór obiektów `Jedzenie`

Implementuje `proj.selekcjanatur.InterfejsSymulacji`.

6.14.3.6 getLudzie()

```
List< Czlowiek > proj.selekcjanatur.SymulacjaPlik.getLudzie ()
```

Zwraca listę ludzi w odtwarzanej symulacji.

Zwraca

Lista obiektów [Czlowiek](#)

Implementuje [proj.selekcjanatur.InterfejsSymulacji](#).

6.14.3.7 przemiescCzlowieka()

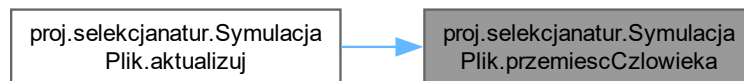
```
void proj.selekcjanatur.SymulacjaPlik.przemiescCzlowieka (
    String[] dane) [private]
```

Przenosi człowieka na nową pozycję

Parametry

<i>dane</i>	Tablica parametrów zdarzenia PRZEMIESZCZENIE
-------------	--

Oto graf wywoływań tej funkcji:



6.14.3.8 usunCzlowieka()

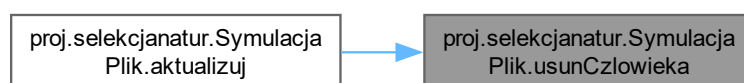
```
void proj.selekcjanatur.SymulacjaPlik.usunCzlowieka (
    String[] dane) [private]
```

Usuwa martwego człowieka.

Parametry

<i>dane</i>	Tablica parametrów zdarzenia SMIERC
-------------	-------------------------------------

Oto graf wywoływań tej funkcji:



6.14.3.9 usunJedzenie()

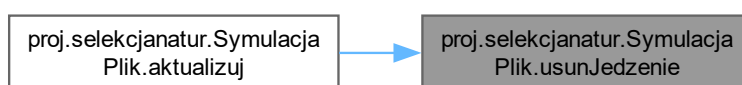
```
void proj.selekcjanatur.SymulacjaPlik.usunJedzenie (
    String[] dane) [private]
```

Usuwa zjedzone jedzenie.

Parametry

<code>dane</code>	Tablica parametrów zdarzenia ZJEDZONO
-------------------	---------------------------------------

Oto graf wywoływań tej funkcji:



6.14.4 Dokumentacja atrybutów składowych

6.14.4.1 index

```
int proj.selekcjanatur.SymulacjaPlik.index = 0 [private]
```

Aktualny indeks przetwarzanego zdarzenia.

6.14.4.2 jedzenie

```
final Set<Jedzenie> proj.selekcjanatur.SymulacjaPlik.jedzenie = new HashSet<>() [private]
```

Zbiór jedzenia w odtwarzanej symulacji.

6.14.4.3 ludzie

```
final List<Czlowiek> proj.selekcjanatur.SymulacjaPlik.ludzie = new ArrayList<>() [private]
```

Lista ludzi w odtwarzanej symulacji.

6.14.4.4 zdarzenia

```
final List<String> proj.selekcjanatur.SymulacjaPlik.zdarzenia [private]
```

Lista wczytanych zdarzeń z pliku.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [src/main/java/proj/selekcjanatur/SymulacjaPlik.java](#)

Rozdział 7

Dokumentacja plików

7.1 Dokumentacja pliku src/main/java/module-info.java

Deklaracja modułu Java dla aplikacji Symulacja Selekcji Naturalnej.

7.1.1 Opis szczegółowy

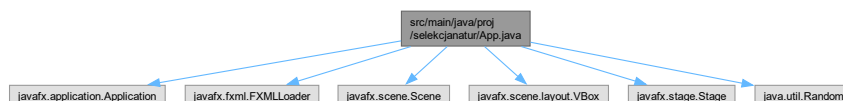
Deklaracja modułu Java dla aplikacji Symulacja Selekcji Naturalnej.

7.2 Dokumentacja pliku src/main/java/proj/selekcjanatur/App.java

Główna klasa aplikacji odpowiedzialna za uruchomienie symulacji.

```
import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
import java.util.Random;
```

Wykres zależności załączania dla App.java:



Komponenty

- class [proj.selekcjanatur.App](#)

Główna klasa aplikacji dziedzicząca po Application z JavaFX.

Pakiety

- package [proj.selekcjanatur](#)

7.2.1 Opis szczegółowy

Główna klasa aplikacji odpowiedzialna za uruchomienie symulacji.

Autor

Michał Kowalewski

Mateusz Sobański

Krystian Szydłowski

Data

20.06.2025

7.3 Dokumentacja pliku

src/main/java/proj/selekcjanatur/AppController.java

Kontroler głównego interfejsu symulacji ewolucyjnej.

```
import javafx.animation.KeyFrame;
import javafx.animation.Timeline;
import javafx.application.Platform;
import javafx.fxml.FXML;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.layout.*;
import javafx.util.Duration;
```

Wykres zależności załączania dla AppController.java:



Komponenty

- class [proj.selekcjanatur.AppController](#)
Główny kontroler zarządzający interfejsem i logiką symulacji.

Pakiety

- package [proj.selekcjanatur](#)

7.3.1 Opis szczegółowy

Kontroler głównego interfejsu symulacji ewolucyjnej.

7.4 Dokumentacja pliku src/main/java/proj/selekcjanatur/Czlowiek.java

Abstrakcyjna klasa bazowa reprezentująca człowieka w symulacji.

Komponenty

- class [proj.selekcjanatur.Czlowiek](#)
Abstrakcyjna klasa reprezentująca człowieka w symulacji ewolucyjnej.

Pakiety

- package [proj.selekcjanatur](#)

7.4.1 Opis szczegółowy

Abstrakcyjna klasa bazowa reprezentująca człowieka w symulacji.

7.5 Dokumentacja pliku src/main/java/proj/selekcjanatur/Gen.java

Interfejs definiujący podstawowe zachowania genów w symulacji.

Komponenty

- interface [proj.selekcjanatur.Gen](#)
Interfejs reprezentujący gen w symulacji ewolucyjnej.

Pakiety

- package [proj.selekcjanatur](#)

7.5.1 Opis szczegółowy

Interfejs definiujący podstawowe zachowania genów w symulacji.

7.6 Dokumentacja pliku src/main/java/proj/selekcjanatur/GenPredkoscChodzenia.java

Implementacja genu odpowiedzialnego za prędkość poruszania się

Komponenty

- class [proj.selekcjanatur.GenPredkoscChodzenia](#)
Klasa implementująca gen prędkości chodzenia.

Pakiety

- package [proj.selekcjanatur](#)

7.6.1 Opis szczegółowy

Implementacja genu odpowiedzialnego za prędkość poruszania się

7.7 Dokumentacja pliku src/main/java/proj/selekcjanatur/GenWzrok.java

Implementacja genu odpowiedzialnego za zasięg wzroku.

Komponenty

- class [proj.selekcjanatur.GenWzrok](#)
Klasa implementująca gen zasięgu wzroku.

Pakiety

- package [proj.selekcjanatur](#)

7.7.1 Opis szczegółowy

Implementacja genu odpowiedzialnego za zasięg wzroku.

7.8 Dokumentacja pliku src/main/java/proj/selekcjanatur/GenZapotrzebowanieEnergetyczne.java

Implementacja genu określającego metabolizm organizmu.

Komponenty

- class [proj.selekcjanatur.GenZapotrzebowanieEnergetyczne](#)
Klasa implementująca gen zapotrzebowania energetycznego.

Pakiety

- package [proj.selekcjanatur](#)

7.8.1 Opis szczegółowy

Implementacja genu określającego metabolizm organizmu.

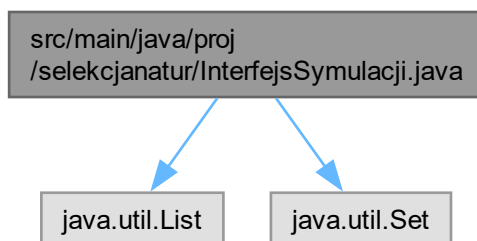
7.9 Dokumentacja pliku

src/main/java/proj/selekcjanatur/InterfejsSymulacji.java

Interfejs definiujący podstawowe operacje symulacji ewolucyjnej.

```
import java.util.List;  
import java.util.Set;
```

Wykres zależności załączania dla InterfejsSymulacji.java:



Komponenty

- interface [proj.selekcjanatur.InterfejsSymulacji](#)
Główny interfejs systemu symulacji ewolucyjnej.

Pakiety

- package [proj.selekcjanatur](#)

7.9.1 Opis szczegółowy

Interfejs definiujący podstawowe operacje symulacji ewolucyjnej.

7.10 Dokumentacja pliku src/main/java/proj/selekcjanatur/Jedzenie.java

Klasa reprezentująca źródło pożywienia w symulacji.

Komponenty

- class [proj.selekcjanatur.Jedzenie](#)
Klasa modelująca pożywienie w symulacji ewolucyjnej.

Pakiety

- package [proj.selekcjanatur](#)

7.10.1 Opis szczegółowy

Klasa reprezentująca źródło pożywienia w symulacji.

7.11 Dokumentacja pliku src/main/java/proj/selekcjanatur/Kobieta.java

Klasa reprezentująca kobietę w symulacji ewolucyjnej.

Komponenty

- class [proj.selekcjanatur.Kobieta](#)
Specjalizacja klasy [Czlowiek](#) reprezentująca kobietę

Pakiety

- package [proj.selekcjanatur](#)

7.11.1 Opis szczegółowy

Klasa reprezentująca kobietę w symulacji ewolucyjnej.

7.12 Dokumentacja pliku src/main/java/proj/selekcjanatur/Mezczyzna.java

Klasa reprezentująca mężczyznę w symulacji ewolucyjnej.

Komponenty

- class [proj.selekcjanatur.Mezczyzna](#)
Specjalizacja klasy [Czlowiek](#) reprezentująca mężczyznę

Pakiety

- package [proj.selekcjanatur](#)

7.12.1 Opis szczegółowy

Klasa reprezentująca mężczyznę w symulacji ewolucyjnej.

7.13 Dokumentacja pliku src/main/java/proj/selekcjanatur/StartController.java

Kontroler ekranu startowego symulacji.

```
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Label;
import javafx.scene.control.Slider;
import javafx.scene.layout.BorderPane;
import javafx.stage.Stage;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.nio.file.Files;
import java.nio.file.Path;
```

Wykres zależności załączania dla StartController.java:



Komponenty

- class [proj.selekcjanatur.StartController](#)
Kontroler GUI dla ekranu konfiguracji symulacji.

Pakiety

- package [proj.selekcjanatur](#)

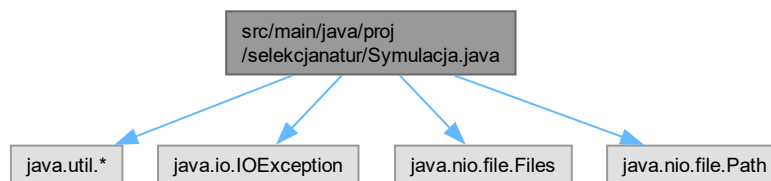
7.13.1 Opis szczegółowy

Kontroler ekranu startowego symulacji.

7.14 Dokumentacja pliku src/main/java/proj/selekcjanatur/Symulacja.java

Główna klasa implementująca logikę symulacji ewolucyjnej.

```
import java.util.*;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
Wykres zależności załączania dla Symulacja.java:
```



Komponenty

- class [proj.selekcjanatur.Symulacja](#)
Implementacja symulacji ewolucyjnej z dziedziczeniem cech.

Pakiety

- package [proj.selekcjanatur](#)

7.14.1 Opis szczegółowy

Główna klasa implementująca logikę symulacji ewolucyjnej.

7.15 Dokumentacja pliku src/main/java/proj/selekcjanatur/SymulacjaPlik.java

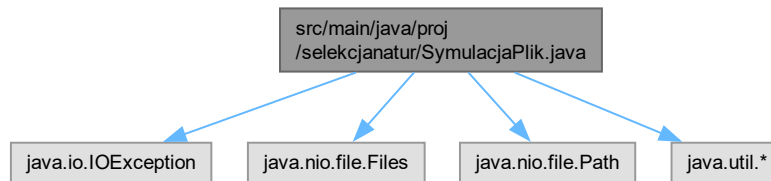
Implementacja odtwarzania symulacji z pliku dziennika zdarzeń

```
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
```



```
import java.util.*;
```

Wykres zależności załączania dla SymulacjaPlik.java:



Komponenty

- class [proj.selekcjanatur.SymulacjaPlik](#)
Klasa odtwarzająca symulację na podstawie zapisanego dziennika zdarzeń

Pakiety

- package [proj.selekcjanatur](#)

7.15.1 Opis szczegółowy

Implementacja odtwarzania symulacji z pliku dziennika zdarzeń

7.16 Dokumentacja pliku src/main/resources/proj/selekcjanatur/grid-view.fxml

7.17 Dokumentacja pliku src/main/resources/proj/selekcjanatur/start-view.fxml

