

SYMULACJA DWÓCH MROWISK, "MRÓWKI"

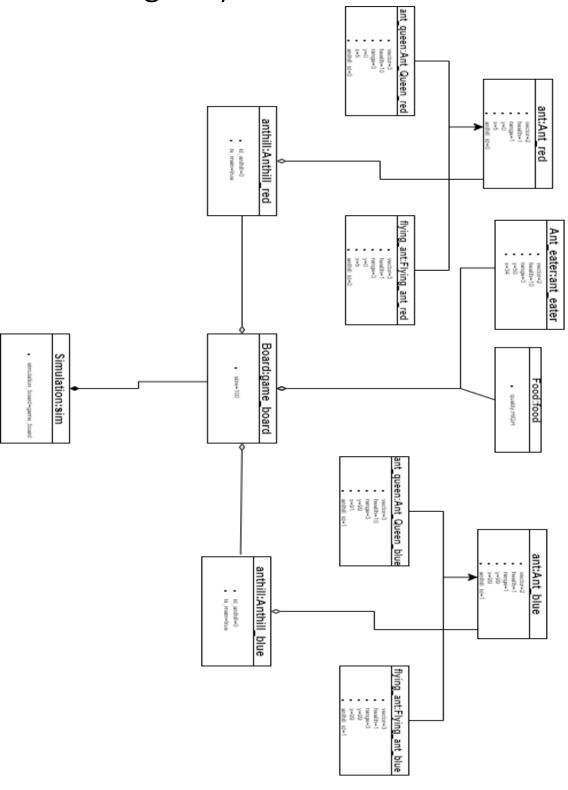
## Szymon Adamczak Maria Kranz

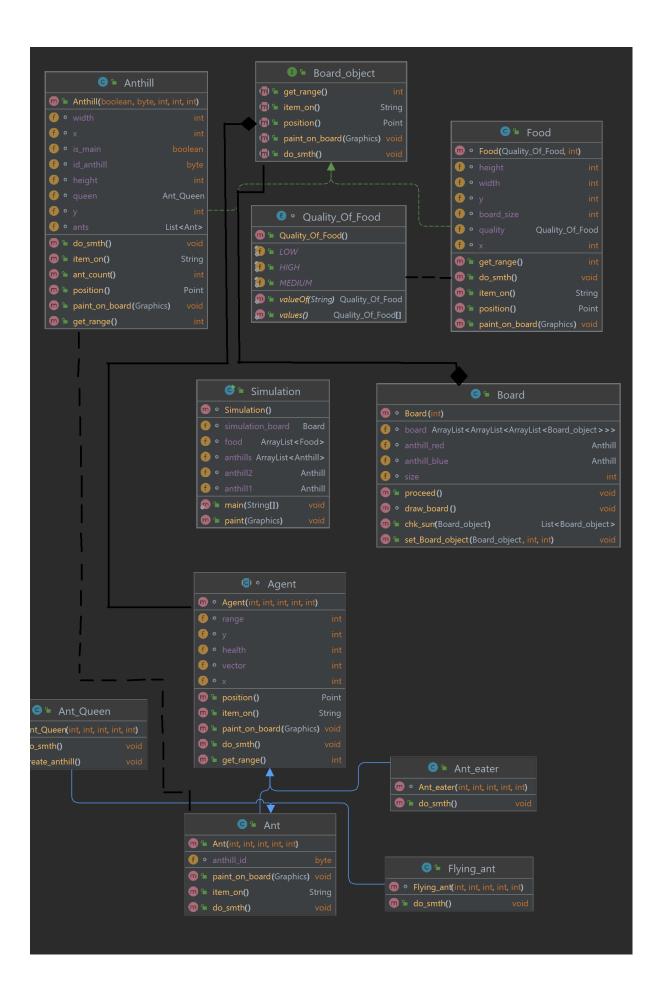
Projekt na kurs "Programowanie Obiektowe"

# Opis symulacji

Na samym początku symulacji generowana jest plansza z dwoma mrowiskami (czerwonym [mrówki żółte oraz brązowe, królowa pomarańczowa] i niebieskim[mrówki niebieskie oraz ciemnofioletowe, królowa zielona]), mrówkojady (różowe) na losowych polach (ilość w zależności od wprowadzonych danych) oraz jedzenie również na losowych polach(czarne, jasnoszare i szare) (ilość jedzenia w zależności od wprowadzonych danych). W każdym mrowisku generuje się pewna ilość mrówek, które następnie są ustawiane na planszy. Mrówki chodzą w losowych kierunkach i w przypadku znalezienia w swoim zasięgu jedzenia wchodzą na pole z jedzeniem. W momencie, w którym mrówka wejdzie na pole z jedzeniem pojawiają się w jej mrowisku nowe mrówki (w zależności od jakości jedzenia: HIGH- pojawia się latająca mrówka, która ma większy zasięg od zwykłej mrówki oraz większą ilość "życia", która pozwala na większe szanse wygrania w przypadku kolizji z przeciwną mrówką, MEDIUM- w mrowisku pojawiają się dwie nowe mrówki, LOW- w mrowisku pojawia się jedna nowa mrówka) jedzenie zmienia swoją pozycję na losowo wygenerowaną nową pozycje i symulacja toczy się dalej. Jedzenie co kilka iteracji "starzeję się", oznacza to że zmienia swoją jakość z, bazowo początkowej, HIGH na MEDIUM lub z MEDIUM na LOW. Mrówki mogą wejść w interakcję z mrówkami z przeciwnego mrowiska, polega to na tzw. "walce" między nimi. W momencie, w którym dwie mrówki z przeciwnych mrowisk spotkają się na jednym polu losowana jest pewna wartość dla każdej z nich, zależąca od ilości mrówek w danym mrowisku (im więcej mrówek tym większa szansa na zwycięstwo) i "życia". W zależności która wylosowana wartość będzie większa to ta mrówka wygra spotkanie. Na planszy są również mrówkojady, utrudniające przetrwanie mrówek w początkowych stadiach symulacji albowiem ich szansa na wygranie z mrówką jest pięciokrotnością bazowej liczby mrówek w mrowisku. W przypadku pokonania mrówkojada przez mrówkę w mrowisku pojawią się 3 nowe mrówki. W momencie, w którym w z danego mrowiska nie będzie mrówek na planszy, pojawia się królowa a mrowisko znika (królowa może pojawić się tylko raz). Królowa nie wchodzi w kolizję z przeciwnymi mrówkami jej jedynym zadaniem jest znaleźć jedzenie i założyć nowe mrowisko. W momencie, w którym królowa znajdzie się na tym samym polu co jedzenie (jakiejkolwiek jakości) tworzy nowe mrowisko, które wypuszcza bazową ilość mrówek, i znika. Symulacja kończy się gdy po pojawieniu się królowej zginą wszystkie mrówki z nowo wygenerowanego mrowiska.

Diagramy klas oraz obiektów





# Opis scenariuszy

#### 1. scenario 1.txt

Scenariusz ten przewiduje plansze 25x25, na której bazowo z mrowiska wychodzi 10 mrówek, jedzenia na planszy jest również 10 oraz na planszy pojawi się 8 mrówkojadów. Raczej optymalna symulacja.

#### 2. scenario 2.txt

Scenariusz ten przewiduje plansze 25x25, na której bazowo z mrowiska wychodzi 5 mrówek, jedzenia na planszy jest 4 oraz na planszy pojawi się 10 mrówkojadów. W tym scenariuszu, może się zdarzyć, że mrowiska nie osiągną swojego pełnego potencjału ponieważ duża ilość mrówkojadów utrudni rozwój.

#### 3. scenario\_3.txt

Scenariusz ten przewiduje plansze 50x50, na której bazowo z mrowiska wychodzi 30 mrówek, jedzenia na planszy jest 15 oraz na planszy pojawi się 20 mrówkojadów. Bazowa duża ilość mrówek i duża ilość jedzenia pozwala na dosyć prędki rozwój mrowiska, jednakże w przypadku gdy jedno z mrowisk zostanie zdominowane "druga szansa" jaką jest królowa rzadko pozwala na odbicie się mrowiska co oznacza porażkę.

#### 4. scenario\_4.txt

Scenariusz ten przewiduje plansze 5x5, na której bazowo z mrowiska wychodzi 2 mrówek, jedzenia na planszy jest 1 oraz na planszy pojawi się 1 mrówkojad. W tym scenariuszu zwykle dochodzi do zakończenia symulacji w ciągu 30 sekund ponieważ mały rozmiar planszy powoduje, że praktycznie w każdej iteracji dochodzi do kolizji mrówki z mrówką co bardzo przyśpiesza zakończenie całej symulacji.

SEARCH: Q Search	×

<b>Packages</b>	

Description

agents

Package

env

PACKAGE: DESCRIPTION | RELATED PACKAGES | CLASSES AND INTERFACES

SEARCH: Q Search

X

# Package agents

package agents

# Classes

Class	Description
Agent	Klasa abstrakcyjna, po ktorej dziedzicza wszyscy agenci bioracy udział w symulacji
Ant	Klasa odpowiadajaca za agenta-mrowke
Ant_eater	Klasa odpowiadajaca za agenta-mrowkojada
Ant_Queen	Klasa odpowiadajaca za agenta-krolowa mrowek
Flying_ant	Klasa odpowiadajaca za agenta-latajaca mrowke

Package agents

# **Class Agent**

java.lang.Object<sup>™</sup> agents.Agent

All Implemented Interfaces:

Board\_object

Direct Known Subclasses:

Ant, Ant\_eater

public abstract class Agent extends Object<sup>™</sup> implements Board\_object

Klasa abstrakcyjna, po ktorej dziedzicza wszyscy agenci bioracy udzial w symulacji

# Field Summary

Fields		
Modifier and Type	Field	Description
(package private) final int	health	"zdrowie" agenta (zwieksza szanse przezycia po wejsciu na pole z wrogim agentem)
(package private) static Random <sup>©</sup>	<sup>™</sup> rand	zmienna sluzaca do losowania koordynatow na ktore przesunie sie agent
(package private) final int	range	zasieg poruszania sie i widzenia agentow
int	x	Pozycja agenta x
(package private) final int	x_move_range	zasieg mozliwego poruszania (rozmiar planszy) w osi x
int	у	Pozycja agenta y
(package private) final int	y_move_range	zasieg mozliwego poruszania (rozmiar planszy) w osi y

## **Constructor Summary**

Constructors

Description Constructor

Agent(int x, int y, int range, int health, int x\_move\_range, int y\_move\_range)

SEARCH: Q Search

(package privace, rimat int

# **Constructor Summary**

#### Constructors

Description Constructor

Agent(int x, int y, int range, int health, int x\_move\_range, int y\_move\_range)

# **Method Summary**

All Methods	nstance Methods	Concrete Methods	
<b>Modifier and Type</b>	Method	1	Description
void	<pre>do_smth(int x</pre>	-	Metoda z interfejsu, ktora w przypadku klasy Agent i klas dziedziczacych po niej odpowiada za poruszanie sie po planszy (jedynie za zmiane parametrow x oraz y faktyczne poruszanie odbywa sie w klasie Board)
int	<pre>get_range()</pre>	]	Metoda zwraca zasieg agenta
int	<pre>getHealth()</pre>	]	Metoda zwracajaca wartosc pola health
Point™	<pre>position()</pre>	]	Metoda zwraca pozycje agenta w postaci Point

# Methods inherited from class java.lang.Object

clone<sup>1</sup>, equals<sup>1</sup>, finalize<sup>1</sup>, getClass<sup>1</sup>, hashCode<sup>1</sup>, notify<sup>1</sup>, notifyAll<sup>1</sup>, toString<sup>1</sup>, wait<sup>1</sup>, wait<sup>1</sup>

#### Field Details

X

public int x

Pozycja agenta x

#### rand

static Random<sup>™</sup> rand

zmienna sluzaca do losowania koordynatow na ktore przesunie sie agent

SEARCH: Q Search

## rand

static Random<sup>™</sup> rand

zmienna sluzaca do losowania koordynatow na ktore przesunie sie agent

## У

public int y

Pozycja agenta y

## range

final int range

zasieg poruszania sie i widzenia agentow

### health

final int health

"zdrowie" agenta (zwieksza szanse przezycia po wejsciu na pole z wrogim agentem)

# x\_move\_range

final int x\_move\_range

zasieg mozliwego poruszania (rozmiar planszy) w osi x

# y\_move\_range

final int y\_move\_range

zasieg mozliwego poruszania (rozmiar planszy) w osi y

# **Constructor Details**

Agent

SEARCH: Q Search

# **Constructor Details**

## Agent

```
Agent(int x,
      int y,
     int range,
     int health,
     int x_move_range,
      int y_move_range)
```

#### **Method Details**

#### do\_smth

```
public void do_smth(int x,
                   int y)
```

Metoda z interfejsu, ktora w przypadku klasy Agent i klas dziedziczacych po niej odpowiada za poruszanie sie po planszy (jedynie za zmiane parametrow x oraz y faktyczne poruszanie odbywa sie w klasie Board)

#### Specified by:

do\_smth in interface Board\_object

#### **Parameters:**

- x pozycja x ewentualnego jedzenia uzywana tylko w klasie Ant i dziedziczacych po niej klasach
- y pozycja y ewentualnego jedzenia uzywana tylko w klasie Ant i dziedziczacych po niej klasach

#### get\_range

```
public int get_range()
```

Metoda zwraca zasieg agenta

#### **Specified by:**

get\_range in interface Board\_object

#### **Returns:**

wartosc pola range

SEARCH: Q Search

Metoda z interfejsu, ktora w przypadku klasy Agent i klas dziedziczacych po niej odpowiada za poruszanie sie po planszy (jedynie za zmiane parametrow x oraz y faktyczne poruszanie odbywa sie w klasie Board)

Specified by:

do\_smth in interface Board\_object

**Parameters:** 

x - pozycja x ewentualnego jedzenia uzywana tylko w klasie Ant i dziedziczacych po niej klasach

y - pozycja y ewentualnego jedzenia uzywana tylko w klasie Ant i dziedziczacych po niej klasach

#### get\_range

public int get\_range()

Metoda zwraca zasieg agenta

Specified by:

get\_range in interface Board\_object

**Returns:** 

wartosc pola range

#### position

public Point<sup>™</sup> position()

Metoda zwraca pozycje agenta w postaci Point

**Specified by:** 

position in interface Board\_object

**Returns:** 

zwraca typ Point wskazujacy na pozycje agenta na plnaszy

### getHealth

public int getHealth()

Metoda zwracajaca wartosc pola health

**Returns:** 

wartosc pola health

SEARCH: Q Search

Package agents

#### **Class Ant**

java.lang.Object<sup>™</sup> agents.Agent agents.Ant

**All Implemented Interfaces:** 

Board\_object

**Direct Known Subclasses:** 

Ant\_Queen, Flying\_ant

public class Ant extends Agent

Klasa odpowiadajaca za agenta-mrowke

## Field Summary

#### Fields

Field **Modifier and Type Description** 

anthill\_id id mrowiska do ktorego dana mrowka nalezy final byte

Fields inherited from class agents. Agent

health, rand, range, x, x\_move\_range, y, y\_move\_range

### **Constructor Summary**

#### Constructors

Description Constructor

Description

Ant(int x, int y, int range, int health, int x\_move\_range, int y\_move\_range, byte anthill\_id)

## **Method Summary**

**All Methods** 

**Instance Methods** 

**Concrete Methods** 

**Modifier and Type** Method

SEARCH: Q Search

# **Method Summary**

All Methods	Instance Methods   Concrete Met	nods
Modifier and Type	e Method	Description
void	<pre>do_smth(int x, int y)</pre>	rozszerzenie metody z klasy Agent z interfejsu Board_object.
Methods inher	rited from class agents.Agent	

get\_range, getHealth, position

## Methods inherited from class java.lang.Object

clone<sup>1</sup>, equals<sup>1</sup>, finalize<sup>1</sup>, getClass<sup>1</sup>, hashCode<sup>1</sup>, notify<sup>1</sup>, notifyAll<sup>1</sup>, toString<sup>1</sup>, wait<sup>1</sup>, wait<sup>1</sup>

### Field Details

#### anthill\_id

public final byte anthill\_id

id mrowiska do ktorego dana mrowka nalezy

## **Constructor Details**

## Ant

```
public Ant(int x,
           int y,
           int range,
          int health,
          int x_move_range,
           int y_move_range,
           byte anthill_id)
```

## **Method Details**

SEARCH: Q Search

# anthill\_id

public final byte anthill\_id id mrowiska do ktorego dana mrowka nalezy

### **Constructor Details**

#### Ant

```
public Ant(int x,
           int y,
           int range,
           int health,
           int x_move_range,
           int y_move_range,
           byte anthill_id)
```

#### **Method Details**

#### do\_smth

```
public void do_smth(int x,
                      int y)
rozszerzenie metody z klasy Agent z interfejsu Board_object. Metda na wejsciu dostaje ewentualna pozycje jedzenia w zasiegu jej widoku (pole "range), jezeli przyjela wartosci -1,-1 oznacza to, ze nie ma w poblizu jedzenia
Specified by:
do_smth in interface Board_object
Overrides:
do_smth in class Agent
Parameters:
x - ewemtualna pozycja x jedzenia
y - ewentualna pozycja y jedzenia
```

SEARCH: Q Search

Package agents

# **Class Flying\_ant**

java.lang.Object<sup>™</sup> agents.Agent agents.Ant agents.Flying\_ant

**All Implemented Interfaces:** 

Board\_object

public class Flying\_ant extends Ant

Klasa odpowiadajaca za agenta-latajaca mrowke

# Field Summary

Fields inherited from class agents.Ant

anthill\_id

Fields inherited from class agents. Agent

health, rand, range, x, x\_move\_range, y, y\_move\_range

# **Constructor Summary**

Constructors

Description Constructor

Flying\_ant(int x, int y, int range, int health, int x\_move\_range, int y\_move\_range, byte anthill\_id)

## **Method Summary**

**All Methods Concrete Methods Instance Methods** 

Methods inherited from class agents Agent

**Modifier and Type Method Description** zadnych zmian w porownaniu z klasa nadrzedna do\_smth(int x, int y) void

X

All Methods Instance Methods Concrete Methods

Method

void do\_smth(int x, int y) zadnych zmian w porownaniu z klasa nadrzedna

Description

Methods inherited from class agents. Agent

get\_range, getHealth, position

Methods inherited from class java.lang.Object

clone<sup>17</sup>, equals<sup>17</sup>, finalize<sup>17</sup>, getClass<sup>17</sup>, hashCode<sup>17</sup>, notify<sup>17</sup>, notifyAll<sup>17</sup>, toString<sup>17</sup>, wait<sup>17</sup>, wait<sup>17</sup>

#### **Constructor Details**

**Modifier and Type** 

#### Flying\_ant

#### **Method Details**

#### do\_smth

```
public void do_smth(int x, int y)

zadnych zmian w porownaniu z klasa nadrzedna

Specified by:
do_smth in interface Board_object

Overrides:
do_smth in class Ant

Parameters:
X - pozycja x
Y - pozycja y
```

SEARCH: Q Search

Package agents

# **Class Ant\_Queen**

java.lang.Object<sup>™</sup> agents.Agent agents.Ant agents.Ant\_Queen

#### **All Implemented Interfaces:**

Board\_object

public class Ant\_Queen extends Ant

Klasa odpowiadajaca za agenta-krolowa mrowek

# Field Summary

### Fields inherited from class agents.Ant

anthill\_id

### Fields inherited from class agents. Agent

health, rand, range, x, x\_move\_range, y, y\_move\_range

## **Constructor Summary**

Constructors

Description Constructor

**Description** 

Ant\_Queen(int x, int y, int range, int health, int x\_move\_range, int y\_move\_range, byte anthill\_id)

# **Method Summary**

**Modifier and Type** 

All Methods	Instance Methods	Concrete Methods	

zadnych zmian w porownaniu z klasa nadrzedna do\_smth(int x, int y) void

Methods inherited from class agents Agent

**Method** 

#### Methods inherited from class agents. Agent

get\_range, getHealth, position

## Methods inherited from class java.lang.Object

clone<sup>1</sup>, equals<sup>1</sup>, finalize<sup>1</sup>, getClass<sup>1</sup>, hashCode<sup>1</sup>, notify<sup>1</sup>, notifyAll<sup>1</sup>, toString<sup>1</sup>, wait<sup>1</sup>, wait<sup>1</sup>

#### **Constructor Details**

### Ant\_Queen

```
public Ant_Queen(int x,
                 int y,
                 int range,
                 int health,
                 int x_move_range,
                 int y_move_range,
                 byte anthill_id)
```

#### **Method Details**

### do\_smth

y - pozycja y

```
public void do_smth(int x,
                     int y)
zadnych zmian w porownaniu z klasa nadrzedna
Specified by:
do_smth in interface Board_object
Overrides:
do_smth in class Ant
Parameters:
x - pozycja x
```

Package agents

# **Class Ant\_eater**

java.lang.Object<sup>™</sup> agents.Agent agents.Ant\_eater

**All Implemented Interfaces:** 

Board\_object

public class Ant\_eater extends Agent

Klasa odpowiadajaca za agenta-mrowkojada

## Field Summary

#### Fields inherited from class agents. Agent

health, rand, range, x, x\_move\_range, y, y\_move\_range

### **Constructor Summary**

**Constructors** 

Constructor **Description** 

Ant\_eater(int x, int y, int range, int health, int x\_move\_range, int y\_move\_range)

## **Method Summary**

**Instance Methods Concrete Methods All Methods** 

**Modifier and Type Description Method** zadnych zmian w por $\tilde{A}^3$ wnaniu z klasa nadrzedna do\_smth(int x, int y) void

Methods inherited from class agents. Agent

get\_range, getHealth, position

Methods inherited from class java.lang.Object 

✓

SEARCH: Q Search

## Methods inherited from class agents. Agent

get\_range, getHealth, position

# Methods inherited from class java.lang.Object

clone<sup>1</sup>, equals<sup>1</sup>, finalize<sup>1</sup>, getClass<sup>1</sup>, hashCode<sup>1</sup>, notify<sup>1</sup>, notifyAll<sup>1</sup>, toString<sup>1</sup>, wait<sup>1</sup>, wait<sup>1</sup>

#### **Constructor Details**

#### **Ant\_eater**

```
public Ant_eater(int x,
                 int y,
                 int range,
                 int health,
                 int x_move_range,
                 int y_move_range)
```

#### **Method Details**

#### do\_smth

```
public void do_smth(int x,
                   int y)
```

zadnych zmian w porównaniu z klasa nadrzedna

#### Specified by:

do\_smth in interface Board\_object

#### **Overrides:**

do\_smth in class Agent

#### **Parameters:**

x - pozycja x

y - pozycja y

PACKAGE: DESCRIPTION | RELATED PACKAGES | CLASSES AND INTERFACES

SEARCH: Q Search

X

# Package env

package env

All Classes and Interfaces	Interfaces (	Classes E	Enum Classes
Class	Description	1	
Anthill	Klasa odpov	wiadajaca za	a mrowisko
Board	Klasa odpov	wiadajaca za	a symulacje i cal
Board_object			
Food	Klasa odpov	wiadajaca za	a jedzenie
<b>JPanelEnding</b>	Klasa sluzao	ca do wyswi	etlenia panelu k
<b>JPanelSimulation</b>	Klasa sluzao	ca do przeds	stawienia graficz
<b>JPanelStatistics</b>	Klasa sluzao	ca do stworz	zenia wykresu.
Quality_Of_Food			
Simulation	Klasa odpov	wiadajaca za	a nadzor calej sy

SEARCH: Q Search

## Package env

# **Class Board**

java.lang.Object₫ env.Board

public class **Board** extends Object<sup>™</sup>

Klasa odpowiadajaca za symulacje i cala plansze

# Field Summary

Fields		
Modifier and Type	Field	Description
(package private) Anthill	anthill_blue	Niebieskie mrowisko
(package private) Anthill	anthill_red	Czerwone mrowisko
(package private) ArrayList d <arraylist d<arraylist="" d<board_object="">&gt;&gt;</arraylist>	board	Lista list list typu Board_object, traktowana przez cala symulacje jako plansza, na ktorej dzieje sie symulacja
(package private) final int	size	rozmiar planszy

# **Constructor Summary**

Constructors

Description Constructor

Board(int size)

# **Method Summary**

All Methods Insta	ince Methods Concrete Methods	
Modifier and Type	Method	Description
void	ant_collision(ArrayList⊡ <board_object> list, Simulation sim)</board_object>	Metoda spinajaca wszelkie mozliwe kolizje istotne dla symulacji
List <sup>™</sup> <board_object></board_object>	<pre>chk_surr(Board_object item)</pre>	Metoda sluzaca do sprawdzenia sasiadow znajdujacych sie w zasiegu jedzenia w celu znalezienia jedzenia
private void	<pre>get_rid(Anthill anthill)</pre>	Metoda poprawiajaca pewnego buga w kodzie polegajacego na tym, ze liczba mrowek na planszy i mrowek w liscie w mrowisku sie nie zgadza (mrowki nie znikaly samoistnie no prostu z jakiegos powodu nie znikaly z lsity w mrowisku)

SUMMARY: NESTED   FIELD   CONSTR   METHOD DETAIL: FIELD   CONSTR   METHOD	HOD SEARCH: Q Search
List <pre>coara_object&gt; cnk_surr(boara_object item)</pre>	Metoda siuzaca do sprawdzenia sasiadow znajdujacych sie w zasiegu jedzenia w ceiu znaiezienia jedzenia
private void <b>get rid(Anthill</b> anthill)	Metoda poprawiajaca pewnego buga w kodzie polegajacego na tym, ze liczba mrowek na planszy i mrowek w liscie w mrowisku sie nie zgadza

		riotoda orazaoa ao oprawazonia odoradow zirajaajaoyon oro w zaoroga jouzonia w oora zirarozionia jouzonia
private void	<pre>get_rid(Anthill anthill)</pre>	Metoda poprawiajaca pewnego buga w kodzie polegajacego na tym, ze liczba mrowek na planszy i mrowek w liscie w mrowisku sie nie zgadza (mrowki nie znikaly samoistnie po prostu z jakiegos powodu nie znikaly z lsity w mrowisku)
void	<pre>move_Board_object(Board_object thing, Point old_position)</pre>	Metoda wykonujaca faktyczny ruch po planszy zmieniajac pozycje obiektow na planszy w przypadku gdy ich pozycja sie zmienila
void	<pre>proceed(Simulation sim)</pre>	Metoda ktora przeprowadza zasadniczo cala symulacje, odpowiada za wywolanie metod odpowiadajacych za konkretne rzeczy w przebiegu symulacji t.j poruszanie sie agentow, starzenie jedzenia
void	<pre>release_queen(Simulation sim)</pre>	Metoda sluzaca do stworzenia krolowej w przypadku gdy liczba mrowek w mrowisku wynosi 0 oraz usuniecia mrowiska z planszy
void	<pre>set_Board_object(Board_object thing, int x, int y)</pre>	metoda sluzaca do wpisania obiektu do planszy

# 

clone<sup>1</sup>, equals<sup>1</sup>, finalize<sup>1</sup>, getClass<sup>1</sup>, hashCode<sup>1</sup>, notify<sup>1</sup>, notifyAll<sup>1</sup>, toString<sup>1</sup>, wait<sup>1</sup>, wait<sup>1</sup>

## Field Details

# board

ArrayList<sup>™</sup><ArrayList<sup>™</sup><Board\_object>>> board

Lista list list typu Board\_object, traktowana przez cala symulacje jako plansza, na ktorej dzieje sie symulacja

#### size

final int size

rozmiar planszy

# anthill\_red

Anthill anthill\_red

Czerwone mrowisko

# anthill\_blue

Anthill anthill\_blue

Niebieskie mrowisko

SEARCH: Q Search

### **Constructor Details**

## **Board**

Board(int size)

### **Method Details**

#### proceed

public void proceed(Simulation sim)

Metoda ktora przeprowadza zasadniczo cala symulacje, odpowiada za wywolanie metod odpowiadajacych za konkretne rzeczy w przebiegu symulacji t.j poruszanie sie agentow, starzenie jedzenia

#### **Parameters:**

sim -

#### chk\_surr

public List™<Board\_object> chk\_surr(Board\_object item)

Metoda sluzaca do sprawdzenia sasiadow znajdujacych sie w zasiegu jedzenia w celu znalezienia jedzenia

#### **Parameters:**

item - agent, ktorego sasiadow sprawdzamy

#### **Returns:**

zwraca liste obiektow znajdujacych sie w obrebie agenta

#### set\_Board\_object

public void set\_Board\_object(Board\_object thing,

int x,

int y)

metoda sluzaca do wpisania obiektu do planszy

#### **Parameters:**

thing - obiekt ktory wstawiamy do planszy

x - jego pozycja x

metoda sluzaca do wpisania obiektu do planszy

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

SEARCH: Q Search

# **Parameters:**

thing - obiekt ktory wstawiamy do planszy

x - jego pozycja x

y - jego pozycja y

# move\_Board\_object

public void move\_Board\_object(Board\_object thing, Point<sup>™</sup> old\_position)

Metoda wykonujaca faktyczny ruch po planszy zmieniajac pozycje obiektow na planszy w przypadku gdy ich pozycja sie zmienila

#### **Parameters:**

thing - obiekt, dla ktorego nastepuje przesuniecie

old\_position - punkt, ktory jest jego pozycja na planszy przed przesunieciem (w przypadku gdy pozycja obiektu nie zmienila swoich wartości przesuniecie nie nastepuje)

#### ant\_collision

public void ant\_collision(ArrayList™<Board\_object> list, Simulation sim)

Metoda spinajaca wszelkie mozliwe kolizje istotne dla symulacji

#### **Parameters:**

list - lista obiektow znajdujacych sie na danym polu

sim -

#### release\_queen

public void release\_queen(Simulation sim)

Metoda sluzaca do stworzenia krolowej w przypadku gdy liczba mrowek w mrowisku wynosi 0 oraz usuniecia mrowiska z planszy

#### **Parameters:**

sim -

#### get\_rid

SEARCH: Q Search

### move\_Board\_object

public void move\_Board\_object(Board\_object thing, Point<sup>™</sup> old\_position)

Metoda wykonujaca faktyczny ruch po planszy zmieniajac pozycje obiektow na planszy w przypadku gdy ich pozycja sie zmienila

#### **Parameters:**

thing - obiekt, dla ktorego nastepuje przesuniecie

old\_position - punkt, ktory jest jego pozycja na planszy przed przesunieciem (w przypadku gdy pozycja obiektu nie zmienila swoich wartości przesuniecie nie nastepuje)

### ant\_collision

public void ant\_collision(ArrayList™<Board\_object> list, Simulation sim)

Metoda spinajaca wszelkie mozliwe kolizje istotne dla symulacji

#### **Parameters:**

list - lista obiektow znajdujacych sie na danym polu

sim -

### release\_queen

public void release\_queen(Simulation sim)

Metoda sluzaca do stworzenia krolowej w przypadku gdy liczba mrowek w mrowisku wynosi 0 oraz usuniecia mrowiska z planszy

#### **Parameters:**

sim -

#### get\_rid

private void get\_rid(Anthill anthill)

Metoda poprawiajaca pewnego buga w kodzie polegajacego na tym, ze liczba mrowek na planszy i mrowek w liscie w mrowisku sie nie zgadza (mrowki nie znikaly samoistnie po prostu z jakiegos powodu nie znikaly z lsity w mrowisku)

#### **Parameters:**

anthill - mrowisko, ktorego sprawdzamy zgodność listy mrowek z tym co sie faktycznie znajduje na planszy

Package env

# **Class Anthill**

java.lang.Object₫ env.Anthill

All Implemented Interfaces:

Board\_object

public class **Anthill** extends Object<sup>™</sup> implements Board\_object

Klasa odpowiadajaca za mrowisko

# Field Summary

Fields		
Modifier and Type	Field	Description
(package private) final int	ant_number	bazowa ilosc mrowek, tzn.
private <b>List</b> <sup>™</sup> < <b>Ant</b> >	ants	Lista mrowek znajdujacych sie w danym mrowisku, tzn.
(package private) boolean	has_queen_appeared	sluzy do stwierdzenia czy krolowa zostala juz postawiona w poprzednich iteracjach na planszy
protected <b>ArrayList</b> <a href="#">Integer</a>	> history	Ilosc mrowek w kazdej petli
(package private) final byte	id_anthill	id mrowiska
(package private) boolean	is_main	wstepnie miala odpowiadac za stwierdzanie czy mrowisko jest mrowiskiem glownym, na te chwile odpowiada za stwierdzenie czy mrowisko istnieje czy nie
(package private) int	X	pozycja mrowiska oraz maksymalna wielkosc planszy
(package private) int	x_max	pozycja mrowiska oraz maksymalna wielkosc planszy
(package private) int	у	pozycja mrowiska oraz maksymalna wielkosc planszy
(package private) int	y_max	pozycja mrowiska oraz maksymalna wielkosc planszy

# **Constructor Summary**

SEARCH: Q Search

## **Constructor Summary**

#### Constructors

Description Constructor

Anthill(boolean is\_main, byte id\_anthill, int ant\_number, int x, int y, int x\_max, int y\_max)

# **Method Summary**

All Methods Inst	cance Methods Concrete Metho	ods
Modifier and Type	Method	Description
int	<pre>ant_count()</pre>	Metoda zwracajaca ilosc mrowek danego mrowiska na planszy
void	<pre>create_anthill(int x, int y)</pre>	metoda sprawdzajaca czy pojawila sie juz krolowa i jezeli nie tworzy nowe mrowisko razem z nowymi mrowkami
void	<pre>delete_ant(Ant ant)</pre>	usuwanie mrowki z listy mrowek mrowiska
void	<pre>do_smth(int x, int y)</pre>	metoda pozwalajaca na wykonanie np ruchu dla mrowki czy zestarzenia sie jedzenia
void	<pre>generate_ant()</pre>	Metoda tworzaca obiekt typu Ant i dodajaca go do listy mrowek mrowiska
void	<pre>generate_f_ant()</pre>	Metoda tworzca obiekt typu Flying_ant i dodajaca go do listy mrowek mrowiska
void	<pre>generate_queen()</pre>	
Board_object	<pre>get_ant(int s)</pre>	Metoda, ktora zwraca konkretna mrowke z mrowiska
int	<pre>get_range()</pre>	Metoda sluzaca do uzyskania zasiegu na jaki dany agent moze sie poruszac i/lub widziec
Point♂	<pre>position()</pre>	Metoda, ktora zwraca pozycje mrowiska

# 

clone<sup>1</sup>, equals<sup>1</sup>, finalize<sup>1</sup>, getClass<sup>1</sup>, hashCode<sup>1</sup>, notify<sup>1</sup>, notifyAll<sup>1</sup>, toString<sup>1</sup>, wait<sup>1</sup>, wait<sup>1</sup>

### Field Details

## ants

private List⊡<Ant> ants

Lists marval anciduis arch signy donom marviola, tan hadaarah na planaga

SEARCH: Q Search

# Field Details

#### ants

private List⊡<Ant> ants

Lista mrowek znajdujacych sie w danym mrowisku, tzn. bedacych na planszy

#### is\_main

boolean is\_main

wstepnie miała odpowiadac za stwierdzanie czy mrowisko jest mrowiskiem glownym, na te chwile odpowiada za stwierdzenie czy mrowisko istnieje czy nie

### id\_anthill

final byte id\_anthill

id mrowiska

# has\_queen\_appeared

boolean has\_queen\_appeared

sluzy do stwierdzenia czy krolowa zostala juz postawiona w poprzednich iteracjach na planszy

# ant\_number

final int ant\_number

bazowa ilosc mrowek, tzn. ilosc mrowek jaka powstanie po stworzeniu mrowiska

# X

int x

pozycja mrowiska oraz maksymalna wielkosc planszy

int y

SEARCH: Q Search

У

int y

pozycja mrowiska oraz maksymalna wielkosc planszy

x\_max

int x\_max

pozycja mrowiska oraz maksymalna wielkosc planszy

y\_max

int y\_max

pozycja mrowiska oraz maksymalna wielkosc planszy

history

protected ArrayList⊡<Integer⊡> history

Ilosc mrowek w kazdej petli

## **Constructor Details**

## **Anthill**

```
public Anthill(boolean is_main,
              byte id_anthill,
              int ant_number,
              int x,
              int y,
              int x_max,
              int y_max)
```

## **Constructor Details**

# **Anthill**

```
public Anthill(boolean is_main,
              byte id_anthill,
              int ant_number,
              int x,
              int y,
              int x_max,
              int y_max)
```

### **Method Details**

### do\_smth

```
public void do_smth(int x,
                   int y)
```

#### **Description copied from interface: Board\_object**

metoda pozwalajaca na wykonanie np ruchu dla mrowki czy zestarzenia sie jedzenia

#### Specified by:

do\_smth in interface Board\_object

# generate\_ant

```
public void generate_ant()
```

Metoda tworzaca obiekt typu Ant i dodajaca go do listy mrowek mrowiska

## generate\_f\_ant

public void generate\_f\_ant()

Metoda tworzca obiekt typu Flying\_ant i dodajaca go do listy mrowek mrowiska

### generate\_queen

SEARCH: Q Search

# generate\_f\_ant

public void generate\_f\_ant()

Metoda tworzca obiekt typu Flying\_ant i dodajaca go do listy mrowek mrowiska

# generate\_queen

public void generate\_queen()

# create\_anthill

public void create\_anthill(int x, int y)

metoda sprawdzajaca czy pojawila sie juz krolowa i jezeli nie tworzy nowe mrowisko razem z nowymi mrowkami

#### **Parameters:**

x - nowa pozycja x mrowiska

y - nowa pozycja y mrowiska

# delete\_ant

public void delete\_ant(Ant ant)

usuwanie mrowki z listy mrowek mrowiska

## **Parameters:**

ant - przekazwyany jest parametr, ktora mrowka ma zostac usunieta

# ant\_count

public int ant\_count()

Metoda zwracajaca ilosc mrowek danego mrowiska na planszy

#### **Returns:**

zwraca rozmiar listy mrowek dla mrowiska

## get\_ant

SEARCH: Q Search

public inc ant\_count()

Metoda zwracajaca ilosc mrowek danego mrowiska na planszy

**Returns:** 

zwraca rozmiar listy mrowek dla mrowiska

#### get\_ant

public Board\_object get\_ant(int s)

Metoda, ktora zwraca konkretna mrowke z mrowiska

**Parameters:** 

s - pozycja mrowki w liscie

**Returns:** 

zwraca mrowke znajdujaca sie na pozycji s

#### get\_range

public int get\_range()

**Description copied from interface: Board\_object** 

Metoda sluzaca do uzyskania zasiegu na jaki dany agent moze sie poruszac i/lub widziec

Specified by:

get\_range in interface Board\_object

**Returns:** 

### position

public Point<sup>™</sup> position()

Metoda, ktora zwraca pozycje mrowiska

Specified by:

position in interface Board\_object

**Returns:** 

zwraca punkt, w ktorym znajduje sie mrowka

SEARCH: Q Search

# **Method Summary**

All Methods In	stance Methods	ods
<b>Modifier and Type</b>	Method	Description
void	<pre>do_smth(int x, int y)</pre>	metoda pozwalajaca na wykonanie np ruchu dla mrowki czy zestarzenia sie jedzenia
int	<pre>get_range()</pre>	Metoda sluzaca do uzyskania zasiegu na jaki dany agent moze sie poruszac i/lub widziec
Point♂	<pre>position()</pre>	Metoda zwracajaca pozycje obiektu na planszy

# **Method Details**

## do\_smth

void do\_smth(int x, int y)

metoda pozwalajaca na wykonanie np ruchu dla mrowki czy zestarzenia sie jedzenia

**Parameters:** 

# get\_range

int get\_range()

Metoda sluzaca do uzyskania zasiegu na jaki dany agent moze sie poruszac i/lub widziec

**Returns:** 

## position

Point<sup>™</sup> position()

Metoda zwracajaca pozycje obiektu na planszy

**Returns:** 

#### Package env

### **Class Food**

java.lang.Object₫ env.Food

All Implemented Interfaces:

Board\_object

public class **Food** extends Object<sup>™</sup> implements Board\_object

Klasa odpowiadajaca za jedzenie

# Field Summary

Fields		
Modifier and Type	Field	Description
(package private) int	board_size	pozycja jedzenia
private int	iteration	sluzy do postarzania jedzenia
(package private) Quality_Of_Food	quality	Jakosc jedzenia
(package private) int	x	pozycja jedzenia
(package private) int	у	pozycja jedzenia

# **Constructor Summary**

Constructors

Description Constructor

Food(Quality\_Of\_Food quality, int board\_size)

# **Method Summary**

**All Methods Instance Methods Concrete Methods** 

Modifier and Type

Method

Description

## **Method Summary**

All Methods Inst	All Methods Instance Methods Concrete Methods		
<b>Modifier and Type</b>	Method	Description	
void	<pre>do_smth(int x, int y)</pre>	Metoda sluzaca do starzenia jedzenia	
int	<pre>get_range()</pre>	Metoda sluzaca do uzyskania zasiegu na jaki dany agent moze sie poruszac i/lub widziec	
Point♂	<pre>position()</pre>	Metoda zwracajaca pozycje obiektu na planszy	

## Methods inherited from class java.lang.Object

clone<sup>13</sup>, equals<sup>13</sup>, finalize<sup>13</sup>, getClass<sup>13</sup>, hashCode<sup>13</sup>, notify<sup>13</sup>, notifyAll<sup>13</sup>, toString<sup>13</sup>, wait<sup>13</sup>, wait<sup>13</sup>

## Field Details

Jakosc jedzenia

Quality\_Of\_Food quality

## iteration

quality

private int iteration

sluzy do postarzania jedzenia

int x

pozycja jedzenia

int y

pozycja jedzenia

get range

## do\_smth

public void do\_smth(int x, int y)

Metoda sluzaca do starzenia jedzenia

Specified by:

do\_smth in interface Board\_object

**Parameters:** 

## get\_range

public int get\_range()

**Description copied from interface: Board\_object** 

Metoda sluzaca do uzyskania zasiegu na jaki dany agent moze sie poruszac i/lub widziec

Specified by:

get\_range in interface Board\_object

**Returns:** 

## position

public Point<sup>™</sup> position()

Description copied from interface: Board\_object

Metoda zwracajaca pozycje obiektu na planszy

Specified by:

position in interface Board\_object

**Returns:** 

## **Class JPanelEnding**

java.lang.Object<sup>™</sup> java.awt.Component<sup>™</sup> java.awt.Ċontainer™ javax.swing.JComponent<sup>™</sup> javax.swing.JPanel □ env.JPanelEnding

#### **All Implemented Interfaces:**

ImageObserver<sup>™</sup>, MenuContainer<sup>™</sup>, Serializable<sup>™</sup>, Accessible<sup>™</sup>

public class JPanelEnding extends JPanel □

Klasa sluzaca do wyswietlenia panelu koncowego

See Also:

Serialized Form

## **Nested Class Summary**

Nested classes/interfaces inherited from class javax.swing.JPanel

JPanel.AccessibleJPanel<sup>™</sup>

Nested classes/interfaces inherited from class javax.swing.JComponent

JComponent.AccessibleJComponent<sup>™</sup>

Nested classes/interfaces inherited from class java.awt.Container

Container.AccessibleAWTContainer<sup>™</sup>

Nested classes/interfaces inherited from class java.awt.Component

Component.AccessibleAWTComponent.BaselineResizeBehavior™, Component.BltBufferStrategy™, Component.FlipBufferStrategy™

## Field Summary

SEARCH: Q Search

# **Field Summary**

**Modifier and Type** 

#### **Fields**

(nealess neither to) final Board board	
(package private) final <b>Board board</b>	Tablica, na ktorej wykonaywana jest symulacja

## Fields inherited from class javax.swing.JComponent

**Field** 

listenerList<sup>®</sup>, TOOL\_TIP\_TEXT\_KEY<sup>®</sup>, ui<sup>®</sup>, UNDEFINED\_CONDITION<sup>®</sup>, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT<sup>®</sup>, WHEN\_FOCUSED<sup>®</sup>, WHEN\_IN\_FOCUSED\_WINDOW<sup>®</sup>

**Description** 

## Fields inherited from class java.awt.Component

accessibleContext<sup>©</sup>, BOTTOM\_ALIGNMENT<sup>©</sup>, CENTER\_ALIGNMENT<sup>©</sup>, LEFT\_ALIGNMENT<sup>©</sup>, RIGHT\_ALIGNMENT<sup>©</sup>, TOP\_ALIGNMENT<sup>©</sup>

## Fields inherited from interface java.awt.image.lmageObserver

ABORTE, ALLBITSE, ERRORE, FRAMEBITSE, HEIGHTE, PROPERTIESE, SOMEBITSE, WIDTHE

# **Constructor Summary**

#### **Constructors**

Constructor	Description

JPanelEnding(Board board) Tworzy panel koncowy

# **Method Summary**

**Modifier and Type** 

All Methods	Instance Methods	Concrete Methods
-------------	------------------	------------------

Method

void	paint(Graphics♂ g)	Metoda sluzaca do rysowania na panelu.
VUIU	nathrial applics = (1)	MELOUA SIUZACA UO TVSOWANIA NA DANEIU.

# Methods inherited from class javax.swing.JPanel

getAccessibleContext<sup>™</sup>, getUI<sup>™</sup>, getUIClassID<sup>™</sup>, paramString<sup>™</sup>, setUI<sup>™</sup>, updateUI<sup>™</sup>

## Methods inherited from class javax.swing.JComponent

**Description** 

SEARCH: Q Search

## **Method Summary**

**Modifier and Type** 

**Instance Methods Concrete Methods All Methods** 

**Method** 

paint(Graphics □ q) Metoda sluzaca do rysowania na panelu. void

Methods inherited from class javax.swing.JPanel

getAccessibleContext<sup>1</sup>, getUI<sup>1</sup>, getUIClassID<sup>1</sup>, paramString<sup>1</sup>, setUI<sup>1</sup>, updateUI<sup>1</sup>

#### Methods inherited from class javax.swing.JComponent

addAncestorListener<sup>®</sup>, addNotify<sup>®</sup>, addVetoableChangeListener<sup>®</sup>, computeVisibleRect<sup>®</sup>, createToolTip<sup>®</sup>, disable<sup>®</sup>, enable<sup>®</sup>, firePropertyChange<sup>®</sup>, firePropertyChange<sup>®</sup>, fireVetoableChange<sup>®</sup>, getActionForKeyStroke<sup>®</sup>, getActionMap<sup>®</sup>, getAlignmentX<sup>®</sup>, getAncestorListeners<sup>®</sup>, getAutoscrolls<sup>®</sup>, getBaseline<sup>®</sup>, getBaselineResizeBehavior<sup>®</sup>, getBorder<sup>®</sup>, getBounds<sup>®</sup>, getClientProperty<sup>®</sup>, getComponentGraphics<sup>®</sup>, getComponentPopupMenu<sup>®</sup>, getConditionForKeyStroke<sup>®</sup>, getDebugGraphics<sup>®</sup>, getFontMetrics<sup>®</sup>, getGraphics<sup>®</sup>, getGraphics<sup>®</sup>, getInheritsPopupMenu<sup>®</sup> , getInputMap<sup>®</sup>, getInputMap<sup>®</sup>, getInputVerifier<sup>®</sup>, getInsets<sup>®</sup>, getInsets<sup>®</sup>, getLocation<sup>®</sup>, getMaximumSize<sup>®</sup>, getMinimumSize<sup>®</sup>, getNextFocusableComponent<sup>®</sup>, getPopupLocation<sup>®</sup>, getPreferredSize<sup>®</sup>, getRegisteredKeyStrokes<sup>®</sup>, getRootPane<sup>®</sup>, getSize<sup>®</sup>, getToolTipLocation<sup>®</sup>, getToolTipText<sup>®</sup>, getToolTipText<sup>®</sup>, getTopLevelAncestor<sup>®</sup>, getTransferHandler<sup>®</sup>, getVerifyInputWhenFocusTarget<sup>®</sup>, getVetoableChangeListeners<sup>®</sup>, getVisibleRect<sup>®</sup>, getWidth<sup>®</sup>, getX<sup>®</sup>, getX<sup>®</sup>, getY<sup>®</sup>, grabFocus<sup>®</sup>, isDoubleBuffered<sup>®</sup>, isLightweightComponent<sup>®</sup>, isManagingFocus<sup>®</sup>, isOpaque<sup>®</sup>, isOptimizedDrawingEnabled<sup>®</sup>, isPaintingForPrint<sup>®</sup>, isPaintingOrigin<sup>®</sup>, isPaintingTile<sup>®</sup>, isRequestFocusEnabled<sup>®</sup>, isValidateRoot<sup>®</sup>, paintChildren<sup>®</sup>, paintComponent<sup>®</sup>, paintImmediately<sup>®</sup>, paintImmediately<sup>®</sup>, print<sup>®</sup>, printAll<sup>®</sup>, printBorder<sup>©</sup>, printChildren<sup>©</sup>, printComponent<sup>©</sup>, processComponentKeyEvent<sup>©</sup>, processKeyEvent<sup>©</sup>, processMouseEvent<sup>©</sup>, processMouseMotionEvent<sup>©</sup>, putClientProperty<sup>©</sup>, registerKeyboardAction<sup>©</sup>, registerKeyboardAction<sup>®</sup>, removeAncestorListener<sup>®</sup>, removeNotify<sup>®</sup>, removeVetoableChangeListener<sup>®</sup>, repaint<sup>®</sup>, requestDefaultFocus<sup>®</sup>, requestFocus<sup>®</sup>, requestFocus<sup>®</sup>, requestFocus<sup>®</sup>, requestFocus<sup>®</sup>, removeNotify<sup>®</sup>, re requestFocusInWindow<sup>®</sup>, resetKeyboardActions<sup>®</sup>, reshape<sup>®</sup>, revalidate<sup>®</sup>, scrollRectToVisible<sup>®</sup>, setActionMap<sup>®</sup>, setAlignmentX<sup>®</sup>, setAlignmentY<sup>®</sup>, setAutoscrolls<sup>®</sup>, setBackground<sup>®</sup>, setBorder<sup>®</sup>, setComponentPopupMenu<sup>®</sup>, setDebugGraphicsOptions<sup>®</sup>, setDefaultLocale<sup>®</sup>, setEnabled<sup>®</sup>, setFocusTraversalKeys<sup>®</sup>, setFont<sup>®</sup>, setForeground<sup>®</sup>, setInheritsPopupMenu<sup>®</sup>, setInputMap<sup>®</sup>, setInputVerifier<sup>®</sup>, setMaximumSize<sup>®</sup>, setMinimumSize<sup>®</sup>, setNextFocusableComponent<sup>®</sup>, setOpaque<sup>®</sup>, setPreferredSize<sup>®</sup>, setRequestFocusEnabled<sup>®</sup>, setToolTipText<sup>®</sup>, setTransferHandler<sup>®</sup>, setUI<sup>®</sup>, setVerifyInputWhenFocusTarget<sup>™</sup>, setVisible<sup>™</sup>, unregisterKeyboardAction<sup>™</sup>, update<sup>™</sup>

#### Methods inherited from class java.awt.Container

add<sup>©</sup>, add<sup>©</sup>, add<sup>©</sup>, add<sup>©</sup>, addContainerListener<sup>©</sup>, addImpl<sup>©</sup>, addPropertyChangeListener<sup>©</sup>, addPropertyChangeListener<sup>©</sup>, applyComponentOrientation<sup>©</sup>, areFocusTraversalKeysSet<sup>©</sup>, countComponents<sup>©</sup>, deliverEvent<sup>®</sup>, doLayout<sup>®</sup>, findComponentAt<sup>®</sup>, getComponentAt<sup>®</sup>, getComponentAt<sup>®</sup>, getComponentAt<sup>®</sup>, getComponentCount<sup>®</sup>, getComponentSount<sup>®</sup>, getComponentZorder<sup>®</sup>, getComponentAt<sup>®</sup>, getCompone getFocusTraversalKeys<sup>17</sup>, getFocusTraversalPolicy<sup>17</sup>, getLayout<sup>17</sup>, getMousePosition<sup>18</sup>, insets<sup>18</sup>, invalidate<sup>18</sup>, isFocusCycleRoot<sup>18</sup>, isFocusCycl isFocusTraversalPolicySet<sup>®</sup>, layout<sup>®</sup>, list<sup>®</sup>, list<sup>®</sup>, locate<sup>®</sup>, minimumSize<sup>®</sup>, paintComponents<sup>®</sup>, preferredSize<sup>®</sup>, printComponents<sup>®</sup>, processContainerEvent<sup>®</sup>, processEvent<sup>®</sup>, remove<sup>®</sup>, remove<sup>®</sup>, removeAll<sup>®</sup>, removeContainerListener<sup>13</sup>, setComponentZOrder<sup>13</sup>, setFocusCycleRoot<sup>13</sup>, setFocusTraversalPolicy<sup>13</sup>, setFocusTraversalPolicy<sup>13</sup>, setLayout<sup>13</sup>, transferFocusDownCycle<sup>13</sup>, validate<sup>13</sup>, validateTree<sup>13</sup>

## Methods inherited from class java.awt.Component

action<sup>®</sup>, add<sup>®</sup>, addComponentListener<sup>®</sup>, addFocusListener<sup>®</sup>, addHierarchyBoundsListener<sup>®</sup>, addHierarchyListener<sup>®</sup>, addInputMethodListener<sup>®</sup>, addKeyListener<sup>®</sup>, addMouseListener<sup>®</sup>, addMouseMotionListener<sup>®</sup>, addMouseWheelListener<sup>®</sup>, bounds<sup>®</sup>, checkImage<sup>®</sup>, coalesceEvents<sup>®</sup>, createImage<sup>®</sup>, createImage<sup>®</sup>, createVolatileImage<sup>®</sup>, createVolatileImage<sup>®</sup>, disableEvents<sup>®</sup>, dispatchEvent<sup>®</sup>, enable<sup>®</sup>, enableEvents<sup>®</sup>, enableInputMethods<sup>®</sup>, firePropertyChange<sup>®</sup>, fireP 법, getComponentListeners법, getComponentOrientation법, getCursor법, getForeground법, getFocusCycleRootAncestor법, getFocusListeners법, getFocusTraversalKeysEnabled법, getFont법, getForeground법, getGraphicsConfiguration<sup>®</sup>, getHierarchyBoundsListeners<sup>®</sup>, getHierarchyListeners<sup>®</sup>, getInputContext<sup>®</sup>, getInputMethodListeners<sup>®</sup>, getInputMethodReguests<sup>®</sup>, getKeyListeners<sup>®</sup>, getLocale<sup>®</sup>, getLocation<sup>®</sup>, getLocation0nScreen<sup>®</sup>, getMouseListeners<sup>®</sup>, getMouseMotionListeners<sup>®</sup>, getMouseWheelListeners<sup>®</sup>, getPropertyChangeListeners<sup>®</sup>, getPropertyChangeListeners<sup>®</sup>

SEARCH: Q Search

# Methods inherited from class java.lang.Object

clone<sup>1</sup>, equals<sup>1</sup>, finalize<sup>1</sup>, getClass<sup>1</sup>, hashCode<sup>1</sup>, notify<sup>1</sup>, notifyAll<sup>1</sup>, wait<sup>1</sup>, wait<sup>1</sup>

## Field Details

#### board

final Board board

Tablica, na ktorej wykonaywana jest symulacja

## **Constructor Details**

## **JPanelEnding**

JPanelEnding(Board board)

Tworzy panel koncowy

#### **Parameters:**

board - tablica symulacji

## **Method Details**

## paint

public void paint(Graphics<sup>™</sup> g)

Metoda sluzaca do rysowania na panelu.

#### **Overrides:**

paint<sup>™</sup> in class JComponent<sup>™</sup>

g - the Graphics context in which to paint

## **Class JPanelSimulation**

java.lang.Object<sup>™</sup> java.awt.Component<sup>™</sup> java.awt.Ċontainer™ javax.swing.JComponent<sup>™</sup> javax.swing.JPanel □ env.JPanelSimulation

#### **All Implemented Interfaces:**

ImageObserver<sup>™</sup>, MenuContainer<sup>™</sup>, Serializable<sup>™</sup>, Accessible<sup>™</sup>

public class JPanelSimulation extends JPanel □

Klasa sluzaca do przedstawienia graficznego symulacji

See Also:

JPanel<sup>년</sup>, Serialized Form

## **Nested Class Summary**

Nested classes/interfaces inherited from class javax.swing.JPanel

JPanel.AccessibleJPanel<sup>™</sup>

Nested classes/interfaces inherited from class javax.swing.JComponent

JComponent.AccessibleJComponent<sup>™</sup>

Nested classes/interfaces inherited from class java.awt.Container

Container.AccessibleAWTContainer<sup>™</sup>

Nested classes/interfaces inherited from class java.awt.Component

Component.AccessibleAWTComponent.G, Component.BaselineResizeBehavior<sup>™</sup>, Component.BltBufferStrategy <sup>™</sup>, Component.FlipBufferStrategy <sup>™</sup>

## Field Summary

# **Field Summary**

#### **Fields**

Modifier and Type	Field	Description

(package private) final Board board Tablica, na ktorej wykonywana jest symulacja

## Fields inherited from class javax.swing.JComponent

listenerList<sup>®</sup>, TOOL\_TIP\_TEXT\_KEY<sup>®</sup>, ui<sup>®</sup>, UNDEFINED\_CONDITION<sup>®</sup>, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT<sup>®</sup>, WHEN\_FOCUSED<sup>®</sup>, WHEN\_IN\_FOCUSED\_WINDOW<sup>®</sup>

## Fields inherited from class java.awt.Component

accessibleContext<sup>©</sup>, BOTTOM\_ALIGNMENT<sup>©</sup>, CENTER\_ALIGNMENT<sup>©</sup>, LEFT\_ALIGNMENT<sup>©</sup>, RIGHT\_ALIGNMENT<sup>©</sup>, TOP\_ALIGNMENT<sup>©</sup>

## Fields inherited from interface java.awt.image.lmageObserver

ABORTE, ALLBITSE, ERRORE, FRAMEBITSE, HEIGHTE, PROPERTIESE, SOMEBITSE, WIDTHE

## **Constructor Summary**

#### **Constructors**

Constructor **Description** 

JPanelSimulation(Board board) Tworzy nowy panel do rysowania obiektow

# **Method Summary**

**Instance Methods Concrete Methods All Methods** 

**Modifier and Type Method Description** 

Metoda rysujaca obiekty na panelu paint(Graphics <sup>™</sup> g) void

## Methods inherited from class javax.swing.JPanel

getAccessibleContext<sup>♂</sup>, getUI<sup>♂</sup>, getUIClassID<sup>♂</sup>, paramString<sup>♂</sup>, setUI<sup>♂</sup>, updateUI<sup>♂</sup>

## Methods inherited from class javax.swing.JComponent

SEARCH: Q Search

# Methods inherited from class java.lang.Object

clone<sup>1</sup>, equals<sup>1</sup>, finalize<sup>1</sup>, getClass<sup>1</sup>, hashCode<sup>1</sup>, notify<sup>1</sup>, notifyAll<sup>1</sup>, wait<sup>1</sup>, wait<sup>1</sup>

## Field Details

#### board

final Board board

Tablica, na ktorej wykonywana jest symulacja

## **Constructor Details**

## **JPanelSimulation**

JPanelSimulation(Board board)

Tworzy nowy panel do rysowania obiektow

#### **Parameters:**

board - Obiekt klasy Board

## **Method Details**

## paint

public void paint(Graphics ♂ g)

Metoda rysujaca obiekty na panelu

#### **Overrides:**

paint<sup>™</sup> in class JComponent<sup>™</sup>

g - the Graphics context in which to paint

## **Class JPanelStatistics**

java.lang.Object<sup>™</sup> java.awt.Component™ java.awt.Ċontainer™ javax.swing.JComponent<sup>™</sup> javax.swing.JPanel <sup>™</sup> env.JPanelStatistics

#### **All Implemented Interfaces:**

ActionListener<sup>17</sup>, ImageObserver<sup>17</sup>, MenuContainer<sup>17</sup>, Serializable<sup>17</sup>, EventListener<sup>17</sup>, Accessible<sup>17</sup>

public class JPanelStatistics extends JPanel □ implements ActionListener<sup>™</sup>

Klasa sluzaca do stworzenia wykresu.

See Also:

Serialized Form

## **Nested Class Summary**

Nested classes/interfaces inherited from class javax.swing.JPanel

JPanel.AccessibleJPanel<sup>™</sup>

Nested classes/interfaces inherited from class javax.swing.JComponent

JComponent.AccessibleJComponent<sup>™</sup>

Nested classes/interfaces inherited from class java.awt.Container

Container.AccessibleAWTContainer<sup>™</sup>

Nested classes/interfaces inherited from class java.awt.Component

Component.AccessibleAWTComponent<sup>®</sup>, Component.BaselineResizeBehavior<sup>®</sup>, Component.BltBufferStrategy<sup>®</sup>, Component.FlipBufferStrategy<sup>®</sup>

## Field Summary

SEARCH: Q Search

# Field Summary

## Fields

Modifier and Type	Field	Description
(package private) final <b>Simulation sim</b>		Obiekt klasy Simulation

## Fields inherited from class javax.swing.JComponent

listenerList<sup>®</sup>, TOOL\_TIP\_TEXT\_KEY<sup>®</sup>, ui<sup>®</sup>, UNDEFINED\_CONDITION<sup>®</sup>, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT<sup>®</sup>, WHEN\_FOCUSED<sup>®</sup>, WHEN\_IN\_FOCUSED\_WINDOW<sup>®</sup>

## 

accessibleContext<sup>1</sup>, BOTTOM\_ALIGNMENT<sup>1</sup>, CENTER\_ALIGNMENT<sup>1</sup>, LEFT\_ALIGNMENT<sup>1</sup>, RIGHT\_ALIGNMENT<sup>1</sup>, TOP\_ALIGNMENT<sup>1</sup>

## Fields inherited from interface java.awt.image.lmageObserver

ABORT<sup>12</sup>, ALLBITS<sup>12</sup>, ERROR<sup>12</sup>, FRAMEBITS<sup>12</sup>, HEIGHT<sup>12</sup>, PROPERTIES<sup>12</sup>, SOMEBITS<sup>13</sup>, WIDTH<sup>13</sup>

## **Constructor Summary**

## Constructors

Constructor	Description

JPanelStatistics(Simulation sim) Tworzy nowy panel do rysowania wykresu

## **Method Summary**

All Methods Inst	ance Methods Con	crete Methods	
<b>Modifier and Type</b>	Method	De	escription
void	actionPerformed(Ac	ctionEvent <sup>™</sup> e) M	letoda sluzaca do ponownego wywolania metody paint(Graphics g)
void	paint(Graphics ♂ G)	) M	letoda sluzaca do rysowania wykresu na panelu.

## Methods inherited from class javax.swing.JPanel

getAccessibleContext<sup>™</sup>, getUI<sup>™</sup>, getUIClassID<sup>™</sup>, paramString<sup>™</sup>, setUI<sup>™</sup>, updateUI<sup>™</sup>

SEARCH: Q Search

## Field Details

#### sim

final Simulation sim

Obiekt klasy Simulation

## **Constructor Details**

## **JPanelStatistics**

JPanelStatistics(Simulation sim)

Tworzy nowy panel do rysowania wykresu

#### **Parameters:**

sim - Obiekt klasy Simulation

## **Method Details**

## paint

public void paint(Graphics G)

Metoda sluzaca do rysowania wykresu na panelu.

#### **Overrides:**

paint<sup>™</sup> in class JComponent<sup>™</sup>

#### **Parameters:**

G - the Graphics context in which to paint

## actionPerformed

public void actionPerformed(ActionEvent e)

Motoda cluzaca do nonovinogo varivolania motody naint(Cranbice g)

SEARCH: Q Search

## **Constructor Details**

## **JPanelStatistics**

JPanelStatistics(Simulation sim)

Tworzy nowy panel do rysowania wykresu

**Parameters:** 

sim - Obiekt klasy Simulation

## **Method Details**

## paint

public void paint(Graphics G)

Metoda sluzaca do rysowania wykresu na panelu.

**Overrides:** 

paint<sup>™</sup> in class JComponent<sup>™</sup>

**Parameters:** 

G - the Graphics context in which to paint

## actionPerformed

public void actionPerformed(ActionEvent e)

Metoda sluzaca do ponownego wywolania metody paint(Graphics g)

Specified by:

actionPerformed<sup>™</sup> in interface ActionListener<sup>™</sup>

**Parameters:** 

e - the event to be processed

## **Enum Class Quality\_Of\_Food**

java.lang.Object<sup>™</sup> java.lang.Enum <sup>™</sup><Quality\_Of\_Food> env.Quality\_Of\_Food

#### **All Implemented Interfaces:**

Serializable<sup>17</sup>, Comparable<sup>17</sup><Quality\_Of\_Food>, Constable<sup>17</sup>

enum Quality\_Of\_Food

extends Enum⊡<Quality\_Of\_Food>

## **Nested Class Summary**

Nested classes/interfaces inherited from class java.lang.Enum

Enum.EnumDesc <a>E</a> extends Enum <a>E</a> <a>E</a> <a>S</a> <a>

## **Enum Constant Summary**

**Enum Constants** 

**Description Enum Constant** 

HIGH

LOW

**MEDIUM** 

## **Constructor Summary**

#### Constructors

Description Modifier Constructor

Quality\_Of\_Food() private

## **Method Summary**

SUMMARY: NESTED | ENUM CONSTANTS | FIELD | METHOD DETAIL: ENUM CONSTANTS | FIELD | METHOD

SEARCH: Q Search

## Constructors

**Modifier** Constructor Description

Quality\_Of\_Food() private

All Methods Static Methods

## **Method Summary**

Madifian and Time	Mathad	Description
Modifier and Type	Method	Description

**Concrete Methods** 

Returns the enum constant of this class with the specified name. static Quality\_Of\_Food valueOf(String<sup>™</sup> name)

static Quality\_Of\_Food[] values() Returns an array containing the constants of this enum class, in the order they are declared.

## Methods inherited from class java.lang.Enum

clone<sup>1</sup>, compareTo<sup>1</sup>, describeConstable<sup>1</sup>, equals<sup>1</sup>, finalize<sup>1</sup>, getDeclaringClass<sup>1</sup>, hashCode<sup>1</sup>, name<sup>1</sup>, ordinal<sup>1</sup>, toString<sup>1</sup>, valueOf<sup>1</sup>

## Methods inherited from class java.lang.Object ✓

getClass<sup>1</sup>, notify<sup>1</sup>, notifyAll<sup>1</sup>, wait<sup>1</sup>, wait<sup>1</sup>

#### **Enum Constant Details**

#### LOW

public static final Quality\_Of\_Food LOW

#### **MEDIUM**

public static final Quality\_Of\_Food MEDIUM

#### HIGH

public static final Quality\_Of\_Food HIGH

SUMMARY: NESTED | ENUM CONSTANTS | FIELD | METHOD DETAIL: ENUM CONSTANTS | FIELD | METHOD

SEARCH: Q Search

пічп

public static final Quality\_Of\_Food HIGH

#### **Constructor Details**

## Quality\_Of\_Food

private Quality\_Of\_Food()

#### **Method Details**

#### values

public static Quality\_Of\_Food[] values()

Returns an array containing the constants of this enum class, in the order they are declared.

#### **Returns:**

an array containing the constants of this enum class, in the order they are declared

#### valueOf

public static Quality\_Of\_Food valueOf(String
 name)

Returns the enum constant of this class with the specified name. The string must match exactly an identifier used to declare an enum constant in this class. (Extraneous whitespace characters are not permitted.)

#### **Parameters:**

name - the name of the enum constant to be returned.

the enum constant with the specified name

IllegalArgumentException <sup>™</sup> - if this enum class has no constant with the specified name

NullPointerException <sup>™</sup> - if the argument is null

## **Class Simulation**

java.lang.Object<sup>™</sup> env.Simulation

## All Implemented Interfaces:

ActionListener<sup>©</sup>, EventListener<sup>©</sup>

public class **Simulation** extends Object<sup>™</sup> implements ActionListener<sup>™</sup>

Klasa odpowiadajaca za nadzor calej symulacji

# Field Summary

Fields

Modifier and True	Field	Description
Modifier and Type	Field	Description
(package private) <b>Timer</b> <sup>™</sup>	animationtimer	Czas miedzy wykonaniem kolejnych petli
(package private) final JPanelSimulation	animsim	Panel, na ktorym rysowane sa obiekty
(package private) Anthill	anthill1	Niebieskie mrowisko
(package private) Anthill	anthill2	Czerwone mrowisko
(package private) int	base_ant_count	bazowa ilosc mrowek pojawiajacych sie na poczatku
(package private) final int	base_food_cout	bazowa ilosc jedzenia na planszyq
private <b>File</b> <sup>™</sup>	file	pobieranie z pliku
(package private) final int	number_of_ant_eaters	bazowa ilosc mrowkojadow
private <b>Scanner</b> <sup>™</sup>	scanner	pobieranie z pliku
private <b>Board</b>	simulation_board	
(package private) int	size	Rozmiar tablicy (planszy)

## **Constructor Summary**

SEARCH: Q Search

# **Constructor Summary**

Constructors

Description Constructor

Simulation()

## **Method Summary**

actionPerformed(ActionEvent dots) e) Metoda sluzaca do wykonywania kolejnych petli symulacji void

main(String<sup>™</sup>[] args) static void

Methods inherited from class java.lang.Object

clone<sup>1</sup>, equals<sup>1</sup>, finalize<sup>1</sup>, getClass<sup>1</sup>, hashCode<sup>1</sup>, notify<sup>1</sup>, notifyAll<sup>1</sup>, toString<sup>1</sup>, wait<sup>1</sup>, wait<sup>1</sup>

## Field Details

## simulation\_board

private Board simulation\_board

## animsim

final JPanelSimulation animsim

Panel, na ktorym rysowane sa obiekty

## anthill1

Anthill anthill1

Niebieskie mrowisko

SEARCH: Q Search

# Field Details

# simulation\_board

private Board simulation\_board

## animsim

final JPanelSimulation animsim

Panel, na ktorym rysowane sa obiekty

# anthill1

Anthill anthill1

Niebieskie mrowisko

# anthill2

Anthill anthill2

Czerwone mrowisko

# int size

size

Rozmiar tablicy (planszy)

# number\_of\_ant\_eaters

final int number\_of\_ant\_eaters

bazowa ilosc mrowkojadow

# base\_ant\_count

int base\_ant\_count

SEARCH: Q Search

# base\_ant\_count

int base\_ant\_count

bazowa ilosc mrowek pojawiajacych sie na poczatku

# base\_food\_cout

final int base\_food\_cout

bazowa ilosc jedzenia na planszyq

## animationtimer

Timer<sup>™</sup> animationtimer

Czas miedzy wykonaniem kolejnych petli

## file

private File<sup>™</sup> file

pobieranie z pliku

## scanner

private Scanner<sup>™</sup> scanner

pobieranie z pliku

# **Constructor Details**

## **Simulation**

Simulation()

throws FileNotFoundException<sup>™</sup>

**Throws:** 

FileNotFoundException™

SEARCH: Q Search

private Scanner<sup>™</sup> scanner

pobieranie z pliku

## **Constructor Details**

## **Simulation**

Simulation()

throws FileNotFoundException<sup>™</sup>

Throws:

FileNotFoundException™

## **Method Details**

#### main

public static void main(Stringば[] args) throws FileNotFoundException<sup>™</sup>

**Throws:** 

FileNotFoundException♂

## actionPerformed

public void actionPerformed(ActionEvent e)

Metoda sluzaca do wykonywania kolejnych petli symulacji

Specified by:

actionPerformed<sup>™</sup> in interface ActionListener<sup>™</sup>

**Parameters:** 

e - the event to be processed