

SYMULACJA DWÓCH MROWISK, „MRÓWKI”

Szymon Adamczak

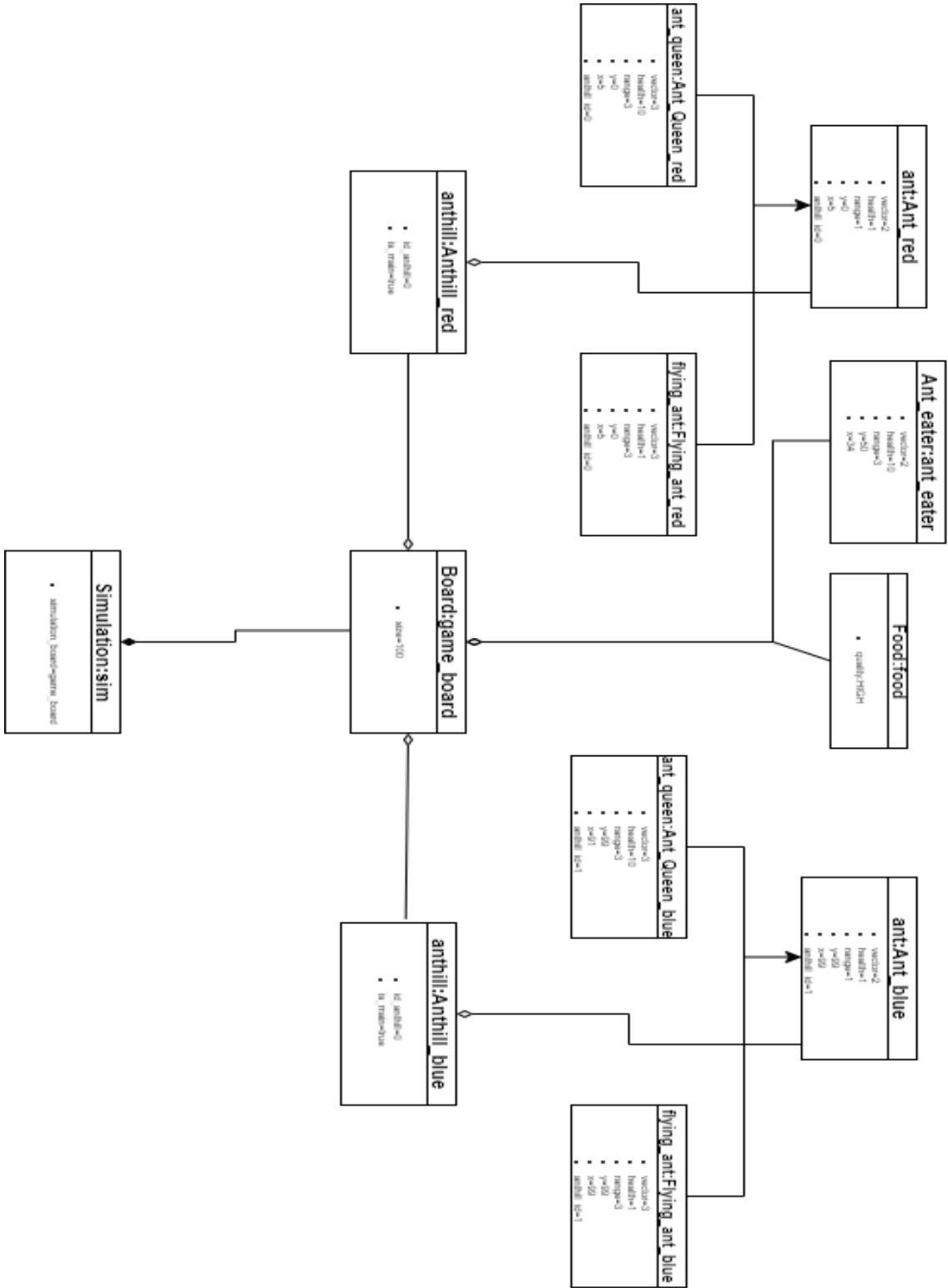
Maria Kranz

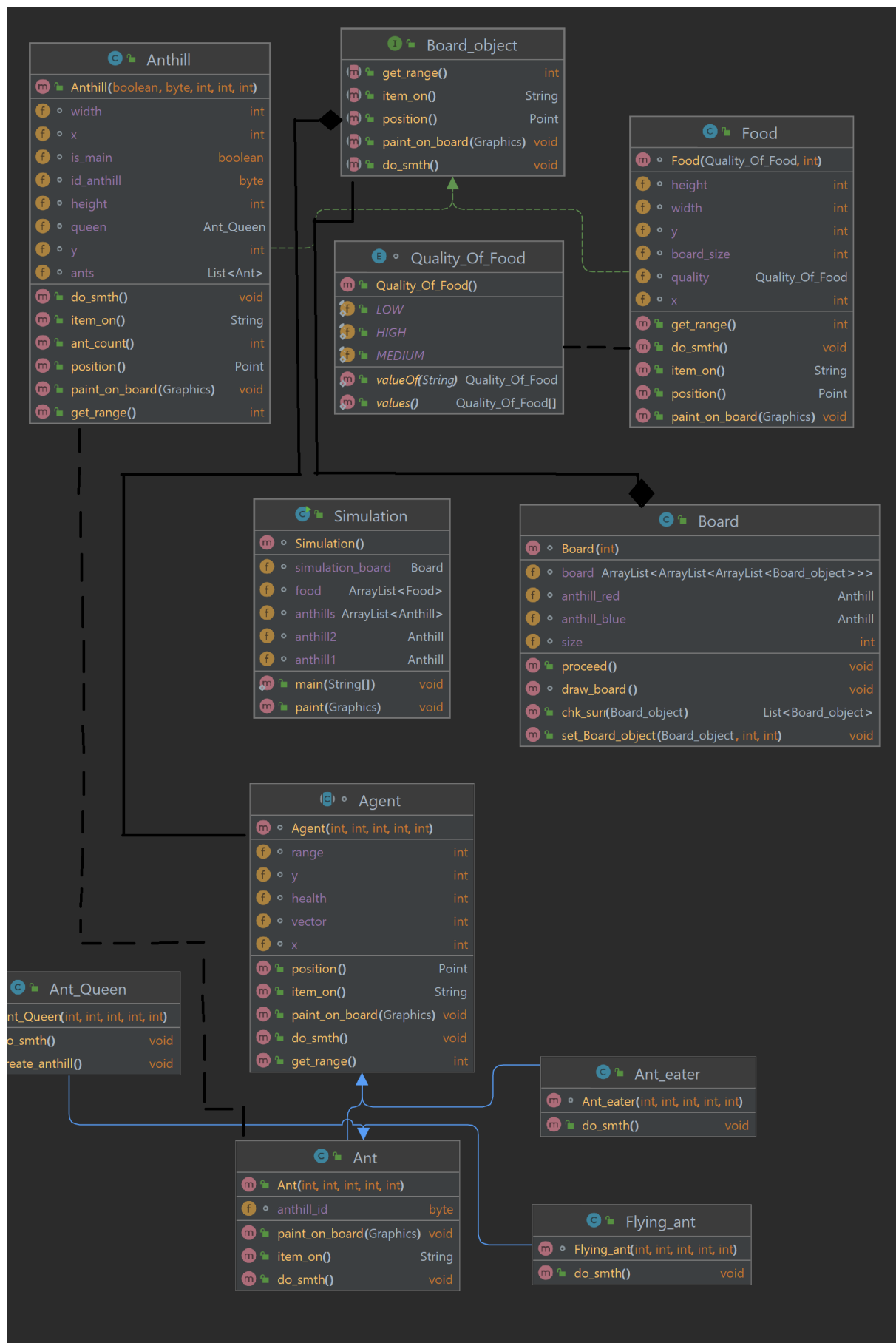
Projekt na kurs „Programowanie
Obiektowe”

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 wchodzi 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 pozycję i symulacja toczy się dalej. Jedzenie co kilka iteracji „starzeje 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 (jakiegokolwiek 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 prędkie 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 przyspiesza zakończenie całej symulacji.

Packages

Package	Description
agents	
env	

Package agents

package agents

Classes	
Class	Description
Agent	Klasa abstrakcyjna, po ktorej dziedzicza wszyscy agenci bioracy udzial 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[↗]
agents.Agent

All Implemented Interfaces:

Board_object

Direct Known Subclasses:

Ant, Ant_eater

```
public abstract class Agent
extends Object↗
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 [↗]	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	y	Pozycja agenta y
(package private) final int	y_move_range	zasieg mozliwego poruszania (rozmiar planszy) w osi y

Constructor Summary

Constructors

Constructor	Description
Agent(int x, int y, int range, int health, int x_move_range, int y_move_range)	

Constructor Summary

Constructors

Constructor	Description
Agent (int x, int y, int range, int health, int x_move_range, int y_move_range)	

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
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)
int	get_range ()	Metoda zwraca zasieg agenta
int	getHealth ()	Metoda zwracajaca wartosc pola health
Point [↗]	position ()	Metoda zwraca pozycje agenta w postaci Point

Methods inherited from class [java.lang.Object](#)[↗]

[clone](#)[↗], [equals](#)[↗], [finalize](#)[↗], [getClass](#)[↗], [hashCode](#)[↗], [notify](#)[↗], [notifyAll](#)[↗], [toString](#)[↗], [wait](#)[↗], [wait](#)[↗], [wait](#)[↗]

Field Details

x
<pre>public int x</pre>
Pozycja agenta x

rand

```
static Random↗ rand
```

zmienna sluzaca do losowania koordynatow na ktore przesunie sie agent

rand

static [Random](#)[↗] rand

zmienna sluzaca do losowania koordynatow na ktore przesunie sie agent

y

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 wejsci u 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

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

position

Metoda z intertejsu, ktora w przypadku klasy Agent i klas dziedziczacych po niej odpowiada za poruszanie sie po planszy (jedynie za zmiane parametrow x oraz y taktyczne 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 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

Package [agents](#)

Class Ant

java.lang.Object

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

Modifier and Type	Field	Description
final byte	anthill_id	id mrowiska do ktorego dana mrowka nalezy

Fields inherited from class [agents.Agent](#)

[health](#), [rand](#), [range](#), [x](#), [x_move_range](#), [y](#), [y_move_range](#)

Constructor Summary

Constructors

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	Description
-------------------	--------	-------------

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	do_smth (int x, int y)	rozszerzenie metody z klasy Agent z interfejsu Board_object.
Methods inherited from class agents.Agent		
get_range, getHealth, position		
Methods inherited from class java.lang.Object		
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait		

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

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 wejsciui 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 - ewentualna pozycja x jedzenia
y - ewentualna pozycja y jedzenia

Package agents

Class Flying_ant

java.lang.Object

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	
Constructor	Description
<code>Flying_ant</code> (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	Description
void	do_smth(int x, int y)	zadnych zmian w porownaniu z klasa nadrzedna

Methods inherited from class agents Agent

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	<code>do_smoth(int x, int y)</code>	zadnych zmian w porownaniu z klasa nadrzedna
Methods inherited from class <code>agents.Agent</code>		
<code>get_range</code> , <code>getHealth</code> , <code>position</code>		
Methods inherited from class <code>java.lang.Object</code>		
<code>clone</code> , <code>equals</code> , <code>finalize</code> , <code>getClass</code> , <code>hashCode</code> , <code>notify</code> , <code>notifyAll</code> , <code>toString</code> , <code>wait</code> , <code>wait</code> , <code>wait</code>		

Constructor Details

Flying_ant
<pre>public Flying_ant(int x, int y, int range, int health, int x_move_range, int y_move_range, byte anthill_id)</pre>

Method Details

do_smoth
<pre>public void do_smoth(int x, int y)</pre>
zadnych zmian w porownaniu z klasa nadrzedna
Specified by: <code>do_smoth</code> in interface <code>Board_object</code>
Overrides: <code>do_smoth</code> in class <code>Ant</code>
Parameters: x - pozycja x y - pozycja y

Package agents

Class Ant_Queen

java.lang.Object

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	
Constructor	Description
<code>Ant_Queen(int x, int y, int range, int health, int x_move_range, int y_move_range, byte anthill_id)</code>	

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	do_smth(int x, int y)	zadnych zmian w porownaniu z klasa nadrzedna

Methods inherited from class agents Agent

Methods inherited from class agents.Agent

get_range, getHealth, position

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

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

```
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

Package agents

Class Ant_eater

java.lang.Object

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

All MethodsInstance MethodsConcrete Methods

Modifier and Type	Method	Description
void	do_smth(int x, int y)	zadnych zmian w porównaniu z klasa nadrzedna

Methods inherited from class agents.Agent
get_range, getHealth, position

Methods inherited from class java.lang.Object

Methods inherited from class agents.Agent

get_range, getHealth, position

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

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 env

package env

All Classes and Interfaces	Interfaces	Classes	Enum Classes
Class	Description		
Anthill	Klasa odpowiadajaca za mrowisko		
Board	Klasa odpowiadajaca za symulacje i cala plansze		
Board_object			
Food	Klasa odpowiadajaca za jedzenie		
JPanelEnding	Klasa sluzaca do wyswietlenia panelu koncowego		
JPanelSimulation	Klasa sluzaca do przedstawienia graficznego symulacji		
JPanelStatistics	Klasa sluzaca do stworzenia wykresu.		
Quality_Of_Food			
Simulation	Klasa odpowiadajaca za nadzor calej symulacji		

Package env

Class Board

java.lang.Object
env.Board

```
public class Board
extends Object
```

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<ArrayList<ArrayList<Board_object>>>	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

Constructor	Description
Board(int size)	

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
void	ant_collision(ArrayList<Board_object> list, Simulation sim)	Metoda spinajaca wszelkie mozliwe kolizje istotne dla symulacji
List<Board_object>	chk_surr(Board_object item)	Metoda sluzaca do sprawdzenia sasiadow znajdujacych sie w zasiegu jedzenia w celu znalezienia jedzenia
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)

Methods inherited from class `java.lang.Object`
`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

board

Lista list list typu Board_object, traktowana przez cala symulacje jako plansza, na ktorej dzieje sie symulacja

rozmiar planszy

anthill_red

Czerwone mrowisko

anthill_blue

Niebieskie mrowisko

Constructor Details

Board
<div>Board(int size)</div>

Method Details

proceed
<div>public void proceed(Simulation sim)</div> <div>Metoda ktora przeprowadza zasadniczo cala symulacje, odpowiada za wywołanie metod odpowiadajacych za konkretne rzeczy w przebiegu symulacji t.j poruszanie sie agentow, starzenie jedzenia</div> <div>Parameters:<div>sim -</div></div>
chk_surr
<div>public List<Board_object> chk_surr(Board_object item)</div> <div>Metoda sluzaca do sprawdzenia sasiadow znajdujacych sie w zasiegu jedzenia w celu znalezienia jedzenia</div> <div>Parameters:<div>item - agent, ktorego sasiadow sprawdzamy</div></div> <div>Returns:<div>zwraca liste obiektow znajdujacych sie w obrebie agenta</div></div>

set_Board_object
<div>public void set_Board_object(Board_object thing,<div>int x,<div>int y)</div></div><div>metoda sluzaca do wpisania obiektu do planszy</div><div>Parameters:<div>thing - obiekt ktory wstawiamy do planszy</div><div>x - jego pozycja x</div></div></div>

metoda sluzaca do wpisania obiektu do planszy

- 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↗ 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 wartosci 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

move_Board_object

```
public void move_Board_object(Board_object thing,
                               Point↗ 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 wartosci 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 zgodnosc 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
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 List<Ant>	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 ArrayList<Integer>	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	y	pozycja mrowiska oraz maksymalna wielkosc planszy
(package private) int	y_max	pozycja mrowiska oraz maksymalna wielkosc planszy

Constructor Summary

Constructors

Constructor	Description
-------------	-------------

Constructor Summary

Constructors

Constructor	Description
<code>Anthill</code> (boolean is_main, byte id_anthill, int ant_number, int x, int y, int x_max, int y_max)	

Method Summary

All Methods

Instance Methods

Concrete Methods

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

Methods inherited from class `java.lang.Object`[↗]

`clone`[↗], `equals`[↗], `finalize`[↗], `getClass`[↗], `hashCode`[↗], `notify`[↗], `notifyAll`[↗], `toString`[↗], `wait`[↗], `wait`[↗], `wait`[↗]

Field Details

ants

```
private List↗<Ant> ants
```

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

Field Details

ants

```
private List<Ant> ants
```

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

is_main

```
boolean is_main
```

wstepnie miala 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

y

```
int y
```

y

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)
```

Method Details

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_smoth

```
public void do_smoth(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_smoth 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

generate_f_ant

```
public void generate_f_ant()
```

Metoda tworzcza 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 - przekazwany 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

```
public int 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 position()
```

Metoda, ktora zwraca pozycje mrowiska

Specified by:
[position](#) in interface [Board_object](#)

Returns:
zwraca punkt, w ktorym znajduje sie mrowka

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type	Method	Description
void	do_smoth (int x, int y)	metoda pozwalajaca na wykonanie np ruchu dla mrowki czy zestarzenia sie jedzenia
int	get_range ()	Metoda sluzaca do uzyskania zasiegu na jaki dany agent moze sie poruszac i/lub widziec
Point	position ()	Metoda zwracajaca pozycje obiektu na planszy

Method Details

do_smoth

void do_smoth(int x,
int y)

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

Parameters:

x -
y -

get_range

int get_range()

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

Returns:

position

Point 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
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	y	pozycja jedzenia

Constructor Summary

Constructors

Constructor	Description
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	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	do_smth (int x, int y)	Metoda sluzaca do starzenia jedzenia
int	get_range ()	Metoda sluzaca do uzyskania zasiegu na jaki dany agent moze sie poruszac i/lub widziec
Point [↗]	position ()	Metoda zwracajaca pozycje obiektu na planszy
Methods inherited from class java.lang.Object [↗]		
clone [↗] , equals [↗] , finalize [↗] , getClass [↗] , hashCode [↗] , notify [↗] , notifyAll [↗] , toString [↗] , wait [↗] , wait [↗] , wait [↗]		

Field Details

quality
<div>Quality_Of_Food quality</div> <div>Jakosc jedzenia</div>
iteration
<div>private int iteration</div> <div>sluzy do postarzania jedzenia</div>
x
<div>int x</div> <div>pozycja jedzenia</div>
y
<div>int y</div> <div>pozycja jedzenia</div>

pozycja jedzenia

y

int y

pozycja jedzenia

board_size

int board_size

pozycja jedzenia

Constructor Details

Food

Food(Quality_Of_Food quality,
int board_size)

Method Details

do_smth

public void do_smth(int x,
int y)

Metoda sluzaca do starzenia jedzenia

Specified by:
do_smth in interface Board_object

Parameters:
x -
y -

get range

Method Details

do_smth

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

Metoda sluzaca do starzenia jedzenia

Specified by:
[do_smth](#) in interface [Board_object](#)

Parameters:
x -
y -

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↗ position()
```

Description copied from interface: [Board_object](#)
Metoda zwracajaca pozycje obiektu na planszy

Specified by:
[position](#) in interface [Board_object](#)

Returns:

Package env

Class JPanelEnding

java.lang.Object

java.awt.Component

java.awt.Container

javax.swing.JComponent

javax.swing.JPanel

env.JPanelEnding

All Implemented Interfaces:

ImageObserver, MenuContainer, Serializable, Accessible

```
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

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

JComponent.AccessibleJComponent

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

Container.AccessibleAWTContainer

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

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

Field Summary

Fields

Field Summary

Fields

Modifier and Type	Field	Description
(package private) final	Board board	Tablica, na ktorej wykonaywana jest symulacja

Fields inherited from class javax.swing.JComponent

[listenerList](#), [TOOL_TIP_TEXT_KEY](#), [ui](#), [UNDEFINED_CONDITION](#), [WHEN_ANCESTOR_OF_FOCUSED_COMPONENT](#), [WHEN_FOCUSED](#), [WHEN_IN_FOCUSED_WINDOW](#)

Fields inherited from class java.awt.Component

[accessibleContext](#), [BOTTOM_ALIGNMENT](#), [CENTER_ALIGNMENT](#), [LEFT_ALIGNMENT](#), [RIGHT_ALIGNMENT](#), [TOP_ALIGNMENT](#)

Fields inherited from interface java.awt.image.ImageObserver

[ABORT](#), [ALLBITS](#), [ERROR](#), [FRAMEBITS](#), [HEIGHT](#), [PROPERTIES](#), [SOMEBITS](#), [WIDTH](#)

Constructor Summary

Constructors

Constructor	Description
JPanelEnding (Board board)	Tworzy panel koncowy

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
void	paint (Graphics g)	Metoda sluzaca do rysowania na panelu.

Methods inherited from class javax.swing.JPanel

[getAccessibleContext](#), [getUI](#), [getUIClassID](#), [paramString](#), [setUI](#), [updateUI](#)

Methods inherited from class javax.swing.JComponent

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

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 g)`

Metoda sluzaca do rysowania na panelu.

Overrides:

[paint](#) in class [JComponent](#)

Parameters:

g - the Graphics context in which to paint

Package env

Class JPanelSimulation

java.lang.Object

java.awt.Component

java.awt.Container

javax.swing.JComponent

javax.swing.JPanel

env.JPanelSimulation

All Implemented Interfaces:

ImageObserver, MenuContainer, Serializable, Accessible

public class **JPanelSimulation**
extends JPanel

Klasa sluzaca do przedstawienia graficznego symulacji

See Also:

JPanel, Serialized Form

Nested Class Summary

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

JPanel.AccessibleJPanel

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

JComponent.AccessibleJComponent

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

Container.AccessibleAWTContainer

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

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

Field Summary

Fields

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](#), [TOOL_TIP_TEXT_KEY](#), [ui](#), [UNDEFINED_CONDITION](#), [WHEN_ANCESTOR_OF_FOCUSED_COMPONENT](#), [WHEN_FOCUSED](#), [WHEN_IN_FOCUSED_WINDOW](#)

Fields inherited from class java.awt.Component

[accessibleContext](#), [BOTTOM_ALIGNMENT](#), [CENTER_ALIGNMENT](#), [LEFT_ALIGNMENT](#), [RIGHT_ALIGNMENT](#), [TOP_ALIGNMENT](#)

Fields inherited from interface java.awt.image.ImageObserver

[ABORT](#), [ALLBITS](#), [ERROR](#), [FRAMEBITS](#), [HEIGHT](#), [PROPERTIES](#), [SOMEBITS](#), [WIDTH](#)

Constructor Summary

Constructors

Constructor	Description
JPanelSimulation (Board board)	Tworzy nowy panel do rysowania obiektow

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
void	paint (Graphics g)	Metoda rysujaca obiekty na panelu

Methods inherited from class javax.swing.JPanel

[getAccessibleContext](#), [getUI](#), [getUIClassID](#), [paramString](#), [setUI](#), [updateUI](#)

Methods inherited from class javax.swing.JComponent

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

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](#) in class [JComponent](#)

Parameters:

g - the Graphics context in which to paint

Package env

Class JPanelStatistics

java.lang.Object

java.awt.Component

java.awt.Container

javax.swing.JComponent

javax.swing.JPanel

env.JPanelStatistics

All Implemented Interfaces:

ActionListener, ImageObserver, MenuContainer, Serializable, EventListener, Accessible

```
public class JPanelStatistics
extends JPanel
implements ActionListener
```

Klasa sluzaca do stworzenia wykresu.

See Also:

Serialized Form

Nested Class Summary

<i>Nested classes/interfaces inherited from class javax.swing.JPanel</i>
JPanel.AccessibleJPanel
<i>Nested classes/interfaces inherited from class javax.swing.JComponent</i>
JComponent.AccessibleJComponent
<i>Nested classes/interfaces inherited from class java.awt.Container</i>
Container.AccessibleAWTContainer
<i>Nested classes/interfaces inherited from class java.awt.Component</i>
Component.AccessibleAWTComponent, Component.BaselineResizeBehavior, Component.BltBufferStrategy, Component.FlipBufferStrategy

Field Summary

Fields

Field Summary

Fields

Modifier and Type	Field	Description
(package private) final	Simulation sim	Obiekt klasy Simulation

Fields inherited from class javax.swing.JComponent

[listenerList](#), [TOOL_TIP_TEXT_KEY](#), [ui](#), [UNDEFINED_CONDITION](#), [WHEN_ANCESTOR_OF_FOCUSED_COMPONENT](#), [WHEN_FOCUSED](#), [WHEN_IN_FOCUSED_WINDOW](#)

Fields inherited from class java.awt.Component

[accessibleContext](#), [BOTTOM_ALIGNMENT](#), [CENTER_ALIGNMENT](#), [LEFT_ALIGNMENT](#), [RIGHT_ALIGNMENT](#), [TOP_ALIGNMENT](#)

Fields inherited from interface java.awt.image.ImageObserver

[ABORT](#), [ALLBITS](#), [ERROR](#), [FRAMEBITS](#), [HEIGHT](#), [PROPERTIES](#), [SOMEBITS](#), [WIDTH](#)

Constructor Summary

Constructors

Constructor	Description
JPanelStatistics (Simulation sim)	Tworzy nowy panel do rysowania wykresu

Method Summary

- All Methods
- Instance Methods
- Concrete Methods

Modifier and Type	Method	Description
void	actionPerformed (ActionEvent e)	Metoda sluzaca do ponownego wywolania metody paint(Graphics g)
void	paint (Graphics G)	Metoda sluzaca do rysowania wykresu na panelu.

Methods inherited from class javax.swing.JPanel

[getAccessibleContext](#), [getUI](#), [getUIClassID](#), [paramString](#), [setUI](#), [updateUI](#)

Field Details

sim
<pre>final Simulation sim</pre>
Obiekt klasy Simulation

Constructor Details

JPanelStatistics
<pre>JPanelStatistics(Simulation sim)</pre>
Tworzy nowy panel do rysowania wykresu
Parameters: sim - Obiekt klasy Simulation

Method Details

paint
<pre>public void paint(Graphics G)</pre>
Metoda sluzaca do rysowania wykresu na panelu.
Overrides: paint in class JComponent
Parameters: G - the Graphics context in which to paint

actionPerformed
<pre>public void actionPerformed(ActionEvent e)</pre>
Metoda sluzaca do ponownego wywolania metody paint(Graphics g)

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](#) in class [JComponent](#)

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](#) in interface [ActionListener](#)

Parameters:

e - the event to be processed

Package env

Enum Class Quality_Of_Food

java.lang.Object
 java.lang.Enum<Quality_Of_Food>
 env.Quality_Of_Food

All Implemented Interfaces:

Serializable, Comparable<Quality_Of_Food>, Constable

enum **Quality_Of_Food**
extends Enum<Quality_Of_Food>

Nested Class Summary

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

Enum.EnumDesc<E> extends Enum<E>>

Enum Constant Summary

Enum Constants

Enum Constant	Description
HIGH	
LOW	
MEDIUM	

Constructor Summary

Constructors

Modifier	Constructor	Description
private	Quality_Of_Food()	

Method Summary

Constructors

Modifier	Constructor	Description
private	<code>Quality_Of_Food()</code>	

Method Summary

- All Methods
- Static Methods
- Concrete Methods

Modifier and Type	Method	Description
static <code>Quality_Of_Food</code>	<code>valueOf(String name)</code>	Returns the enum constant of this class with the specified name.
static <code>Quality_Of_Food[]</code>	<code>values()</code>	Returns an array containing the constants of this enum class, in the order they are declared.

Methods inherited from class java.lang.Enum

`clone`, `compareTo`, `describeConstable`, `equals`, `finalize`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `toString`, `valueOf`

Methods inherited from class java.lang.Object

`getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

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
```

enum

```
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.

Returns:

the enum constant with the specified name

Throws:

[IllegalArgumentException](#) - if this enum class has no constant with the specified name

[NullPointerException](#) - if the argument is null

Package env

Class Simulation

java.lang.Object
env.Simulation

All Implemented Interfaces:

ActionListener, EventListener

```
public class Simulation
extends Object
implements ActionListener
```

Klasa odpowiadajaca za nadzor calej symulacji

Field Summary

Fields

Modifier and Type	Field	Description
(package private) Timer	animationtimer	Czas miedzy wykonaniem kolejnych petli
(package private) final JPanelSimulation	ansim	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 File	file	pobieranie z pliku
(package private) final int	number_of_ant_eaters	bazowa ilosc mrowkojadow
private Scanner	scanner	pobieranie z pliku
private Board	simulation_board	
(package private) int	size	Rozmiar tablicy (planszy)

Constructor Summary

Constructors

Constructor Summary

Constructors

Constructor	Description
<code>Simulation()</code>	

Method Summary

All MethodsStatic MethodsInstance MethodsConcrete Methods

Modifier and Type	Method	Description
void	<code>actionPerformed(ActionEvent e)</code>	Metoda sluzaca do wykonywania kolejnych petli symulacji
static void	<code>main(String[] args)</code>	

Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Field Details

simulation_board

```
private Board simulation_board
```

animsim

```
final JPanelSimulation animsim
```

Panel, na ktorym rysowane sa obiekty

anthill1

```
Anthill anthill1
```

Niebieskie mrowisko

Field Details

simulation_board
<pre>private Board simulation_board</pre>
animsim
<pre>final JPanelSimulation animsim</pre>
Panel, na którym rysowane sa obiekty
anthill1
<pre>Anthill anthill1</pre>
Niebieskie mrowisko
anthill2
<pre>Anthill anthill2</pre>
Czerwone mrowisko
size
<pre>int size</pre>
Rozmiar tablicy (planszy)
number_of_ant_eaters
<pre>final int number_of_ant_eaters</pre>
bazowa ilosc mrowkojadow
base_ant_count
<pre>int base_ant_count</pre>
bazowa ilosc mrowkojadow, ktora jest wstawiana do tablicy

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 animationtimer
```

Czas miedzy wykonaniem kolejnych petli

file

```
private File file
```

pobieranie z pliku

scanner

```
private Scanner scanner
```

pobieranie z pliku

Constructor Details

Simulation

```
Simulation()  
    throws FileNotFoundException
```

Throws:
`FileNotFoundException`

Scanner

private [Scanner](#) scanner

pobieranie z pliku

Constructor Details

Simulation

Simulation()
throws [FileNotFoundException](#)

Throws:
[FileNotFoundException](#)

Method Details

main

public static void main([String](#)[] args)
throws [FileNotFoundException](#)

Throws:
[FileNotFoundException](#)

actionPerformed

public void actionPerformed([ActionEvent](#) e)

Metoda sluzaca do wykonywania kolejnych petli symulacji

Specified by:
[actionPerformed](#) in interface [ActionListener](#)

Parameters:
e - the event to be processed