com.george.als.entities.lifeforms

Class ALifeForm

```
java.lang.Object
  └com.george.als.entities.lifeforms.ALifeForm
```

```
public abstract class ALifeForm
extends java.lang.Object
```

The base class for all life forms The class creates and updates all of the life forms on the current map It holds the name and type of life form The x and y positions of the life form And all the other information that is used to update their positions on the map

Author:

Georges Beast

Nested Class Summary

static class	**ALifeForm.Direction** The direction the life form can move

Constructor Summary

ALifeForm(java.lang.String name, java.lang.String type, com.george.als.world.Map map) Contructor for creating a Herbivore or Carnivore

Method Summary

void	**checkBugHasNest**() Check if the life form has a nest
boolean	**checkFood**(com.george.als.entities.food.Food food) Checks if the food the bug is moving to still exists on the map Prevents the bug searching for food that has been eaten
boolean	**detectObstacle**(ALifeForm.Direction direction) Detects and obstacle one cell ahead of the bug in the given direction
ALifeForm.Direction	**getCurrentDirection**()
int	**getEnergy**() Gets the bugs energy
com.george.als.entities.food.Food	**getFood**()
com.george.als.entities.food.Food	**getFoodAt**(int x, int y) Gets the food at a particular position
com.george.als.entities.food.Food	**getFoodToMoveTo**() Gets the food the bug needs to move to

com.george.als.entities.nest.Nest	**getHome**() Gets the bugs home nest
boolean	**getInNest**() Gets if the bug is in its nest
com.george.als.world.Map	**getMap**() Gets the map the bug belongs to
int	**getMaxSmellingDistance**() Gets the bugs smelling distance
boolean	**getMovingToFood**() Gets if the bug is moving to food
boolean	**getMovingToNest**() Gets if the bug is moving to its nest
java.lang.String	**getName**() Gets the name of the bug
ALifeForm.Direction	**getRandomDirectionToMove**() Generates a random direction to move by using random numbers *
java.lang.String	**getType**() Gets the type of bug
int	**getX**() Gets the bugs X position
int	**getY**() Gets the bugs y position
void	**move**(ALifeForm.Direction direction) Moves the bug in a direction This method is called by moveToFood, moveToNest, moveToParent
void	**moveToFood**(com.george.als.entities.food.Food food) Moves the bug to a food item on the map Uses the A* search algorithm
void	**moveToNest**(com.george.als.entities.nest.Nest nest) Moves the bug to a nest on the map Uses the A* start search algorithm
boolean	**overFood**() Checks if the bug is standing over food
void	**setCurrentDirection**(ALifeForm.Direction direction) Sets the current Direction
void	**setEnergy**(int energy) sets the bugs energy
void	**setFoodToMoveTo**(com.george.als.entities.food.Food foodToMoveTo) Sets the bugs food to move to
void	**setHome**(com.george.als.entities.nest.Nest home) Sets the bugs home nest
void	**setInNest**(boolean inNest) Sets if the bug is in its nest
void	**setMovingToFood**(boolean movingToFood) Sets if the bug is moving to food
void	**setMovingToNest**(boolean movingToNest) sets if the bug is moving to a nest
void	**setName**(java.lang.String name)

	Sets a bugs name
void	**setX**(int x) sets the bugs x position
void	**setY**(int y) sets the bugs y position
com.george.als.entities.food.Food	**smellFood**() Checks for food in a give radius This radius is determined by maxSmellingDistance
java.lang.String	**toString**()
void	**update**() The bug update method that is overriden by the different types of bug

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

ALifeForm

```
public ALifeForm(java.lang.String name,
                 java.lang.String type,
                 com.george.als.world.Map map)
```

Contructor for creating a Herbivore or Carnivore

Parameters:

`name` - the name of the life form
`type` - the type of the life form
`map` - the map it belongs to

Method Detail

getName

```
public java.lang.String getName()
```

Gets the name of the bug

Returns:

the bugs name

---

getType

```
public java.lang.String getType()
```

Gets the type of bug

Returns:

the bugs type

---

getX

```
public int getX()
```

Gets the bugs X position

Returns:
the bugs x coordinate

---

getY

```
public int getY()
```

Gets the bugs y position

Returns:
the bugs y coordinate

---

getEnergy

```
public int getEnergy()
```

Gets the bugs energy

Returns:
the bugs energy

---

getMaxSmellingDistance

```
public int getMaxSmellingDistance()
```

Gets the bugs smelling distance

Returns:
the bugs smelling distance

---

getMovingToFood

```
public boolean getMovingToFood()
```

Gets if the bug is moving to food

Returns:
if the bug is moving to food

---

getHome

```
public com.george.als.entities.nest.Nest getHome()
```

Gets the bugs home nest

Returns:
the bugs nest

---

getMap

```
public com.george.als.world.Map getMap()
```

Gets the map the bug belongs to

Returns:
the bugs map

---

getInNest

```
public boolean getInNest()
```

Gets if the bug is in its nest

Returns:
if the bug is in its nest

---

getMovingToNest

```
public boolean getMovingToNest()
```

Gets if the bug is moving to its nest

Returns:
if the bug is moving to its nests

---

getFoodToMoveTo

```
public com.george.als.entities.food.Food getFoodToMoveTo()
```

Gets the food the bug needs to move to

Returns:
the food the bug is moving to

---

setName

```
public void setName(java.lang.String name)
```

Sets a bugs name

Parameters:
`name` - the new name

---

setX

```
public void setX(int x)
```

sets the bugs x position

Parameters:
`x` - the new x position

---

setY

```
public void setY(int y)
```

sets the bugs y position

Parameters:

`y` - the new y position

---

setEnergy

```
public void setEnergy(int energy)
```

sets the bugs energy

Parameters:

`energy` - the new energy

---

setMovingToFood

```
public void setMovingToFood(boolean movingToFood)
```

Sets if the bug is moving to food

Parameters:

`movingToFood` - the boolean value

---

setHome

```
public void setHome(com.george.als.entities.nest.Nest home)
```

Sets the bugs home nest

Parameters:

`home` - the new home nest

---

setInNest

```
public void setInNest(boolean inNest)
```

Sets if the bug is in its nest

Parameters:

`inNest` - the boolean value

---

setMovingToNest

```
public void setMovingToNest(boolean movingToNest)
```

sets if the bug is moving to a nest

Parameters:

`movingToNest` - the boolean value

---

setFoodToMoveTo

```
public void setFoodToMoveTo(com.george.als.entities.food.Food foodToMoveTo)
```

Sets the bugs food to move to

Parameters:

`foodToMoveTo` - the new food the bug should move to

---

getCurrentDirection

```
public ALifeForm.Direction getCurrentDirection()
```

Returns:

the current direction the life form is moving

---

setCurrentDirection

```
public void setCurrentDirection(ALifeForm.Direction direction)
```

Sets the current Direction

Parameters:

`direction` - the new direction the life form is to move

---

update

```
public void update()
```

The bug update method that is overriden by the different types of bug

---

checkBugHasNest

```
public void checkBugHasNest()
```

Check if the life form has a nest

---

moveToFood

```
public void moveToFood(com.george.als.entities.food.Food food)
```

Moves the bug to a food item on the map Uses the A* search algorithm

Parameters:

`food` - the food to move to

See Also:

`AStarPathFinder`

---

moveToNest

```
public void moveToNest(com.george.als.entities.nest.Nest nest)
```

Moves the bug to a nest on the map Uses the A* start search algorithm

Parameters:

`nest` - the nest to move to

See Also:

`AStarPathFinder`

---

move

```
public void move(ALifeForm.Direction direction)
```

Moves the bug in a direction This method is called by moveToFood, moveToNest, moveToParent

Parameters:

`direction` - the direction to move the bug

---

getFood

```
public com.george.als.entities.food.Food getFood()
```

Returns:

the food at the bugs current position

---

getFoodAt

```
public com.george.als.entities.food.Food getFoodAt(int x,
                                                   int y)
```

Gets the food at a particular position

Parameters:

`x` - the x coordinate

`y` - the y coordinate

Returns:

the food at a the x and y coordinate

---

overFood

```
public boolean overFood()
```

Checks if the bug is standing over food

Returns:

true if standing over food, false if not

---

smellFood

```
public com.george.als.entities.food.Food smellFood()
```

Checks for food in a give radius This radius is determined by maxSmellingDistance

Returns:

the food item that is found

---

checkFood

```
public boolean checkFood(com.george.als.entities.food.Food food)
```

Checks if the food the bug is moving to still exists on the map Prevents the bug searching for food that has been eaten

Parameters:
`food` - the food to check
Returns:
true if the food still exist, false if not

---

detectObstacle

```
public boolean detectObstacle(ALifeForm.Direction direction)
```

Detects and obstacle one cell ahead of the bug in the given direction

Parameters:
`direction` - the direction to check
Returns:
true if there is an obstacle, false if not

---

getRandomDirectionToMove

```
public ALifeForm.Direction getRandomDirectionToMove()
```

Generates a random direction to move by using random numbers *

Returns:
the directio to move
See Also:
`Random`

---

toString

```
public java.lang.String toString()
```

Overrides:
`toString` in class `java.lang.Object`

Returns:
the bug information to a string
