ANIMAL

Sugeneruota Doxygen 1.13.2

skyrius 1

Hierarchijos Indeksas

1.1 Klasių hierarchija

Šis paveldėjimo sąrašas yra beveik surikiuotas abėcėlės tvarka:

skyrius 2

Klasės Indeksas

2.1 Klasės

Klasės, struktūros, sąjungos ir sąsajos su trumpais aprašymais:

skyrius 3

Failo Indeksas

3.1 Failai

Visų failų sąrašas su trumpais aprašymais:

skyrius 4

Klasės Dokumentacija

4.1 Animal Klasė

Represents an Animal with replaceable movement.

`#include <Animal.h>`

Vieši Metodai

- Animal (const std::string &name, std::unique_ptr< MovementStrategy > strat)

  Construct a new Animal object.
- Animal (const Animal &o)

  Copy preserves runtime strategy.
- Animal & operator= (const Animal &o)

  Copy assignment operator for the Animal class.
- void move (int dx, int dy)

  Move by (dx,dy) via current strategy.
- void setMovementStrategy (std::unique_ptr< MovementStrategy > strat)

  Set the Movement Strategy object.
- int getID () const

  get Identifier of the animal
- int getX () const

  get X coordinate of the animal
- int getY () const

  get Y coordinate of the animal
- std::string getName () const

  Get the Name object.

Draugai

- std::ostream & operator<< (std::ostream &os, const Animal &a)

  Binary serialization (writes x, y, name, then strategy tag + data).
- std::istream & operator>> (std::istream &is, Animal &a)

  Binary deserialization (reads x, y, name, then strategy tag + data).

4.1.1 Smulkus aprašymas

Represents an Animal with replaceable movement.

4.1.2 Konstruktoriaus ir Destruktoriaus Dokumentacija

4.1.2.1 Animal() [1/2]

```
Animal::Animal (
            const std::string & name,
            std::unique_ptr< MovementStrategy > strat)
```

Construct a new Animal object.

Parametrai

name	animal name
strat	initial movement strategy (default = NormalMovement)

4.1.2.2 Animal() [2/2]

```
Animal::Animal (
            const Animal & o)
```

Copy preserves runtime strategy.

Parametrai

o	Animal to copy from

4.1.3 Metodų Dokumentacija

4.1.3.1 getID()

```
int Animal::getID () const
```

get Identifier of the animal

Gražina

int

4.1.3.2 getName()

```
std::string Animal::getName () const
```

Get the Name object.

Gražina

std::string

4.1.3.3 getX()

```
int Animal::getX () const
```

get X coordinate of the animal

Gražina

int

4.1.3.4 getY()

```
int Animal::getY () const
```

get Y coordinate of the animal

Gražina

int

4.1.3.5 move()

```
void Animal::move (
            int dx,
            int dy)
```

Move by (dx,dy) via current strategy.

Parametrai

dx	delta x
dy	delta y

4.1.3.6 operator=()

```
Animal & Animal::operator= (
            const Animal & o)
```

Copy assignment operator for the Animal class.

Parametrai

o	

Gražina

Animal&

4.1.3.7 setMovementStrategy()

```
void Animal::setMovementStrategy (
            std::unique_ptr< MovementStrategy > strat)
```

Set the Movement Strategy object.

Parametrai

strat	strategy to set

4.1.4 Draugiškų Ir Susijusių Funkcijų Dokumentacija

4.1.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const Animal & a)  [friend]
```

Binary serialization (writes x, y, name, then strategy tag + data).

Parametrai

os	output stream
a	animal to write

Gražina

std::ostream&

4.1.4.2 operator>>

```
std::istream & operator>> (
            std::istream & is,
            Animal & a)  [friend]
```

Binary deserialization (reads x, y, name, then strategy tag + data).

Parametrai

is	input stream
a	animal to read into

Gražina

std::istream&

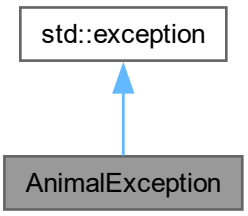
Dokumentacija šiai klasei sugeneruota iš šių failų:

- src/animal/Animal.h
- src/animal/Animal.cpp

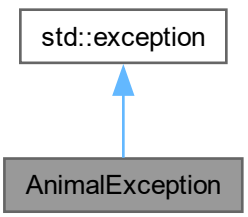
4.2 AnimalException Klasė

`#include <AnimalException.h>`

Paveldimumo diagrama AnimalException:



Bendradarbiavimo diagrama AnimalException:



Vieši Metodai

- AnimalException (const std::string &msg)

  Construct a new Animal Exception object.
- const char * what () const noexcept override

  just a wrapper for std::exception::what()

4.2.1 Konstruktoriaus ir Destruktoriaus Dokumentacija

4.2.1.1 AnimalException()

```
AnimalException::AnimalException (
            const std::string & msg)  [explicit]
```

Construct a new Animal Exception object.

Parametrai

msg	message to be displayed

4.2.2 Metodų Dokumentacija

4.2.2.1 what()

```
const char * AnimalException::what () const  [override], [noexcept]
```

just a wrapper for std::exception::what()

Gražina

const char*

Dokumentacija šiai klasei sugeneruota iš šių failų:

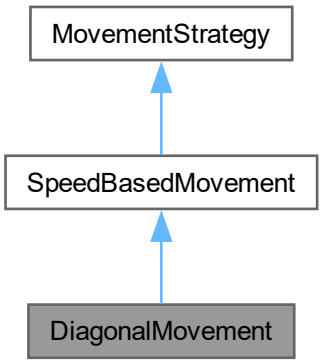
- src/animal/exception/AnimalException.h
- src/animal/exception/AnimalException.cpp

4.3 DiagonalMovement Klasė

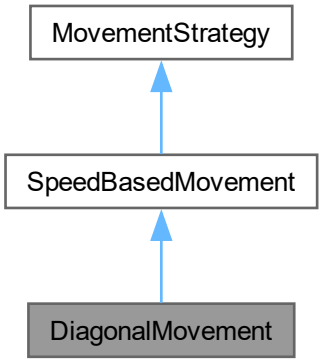
Allows diagonal moves at constant cost.

`#include <DiagonalMovement.h>`

Paveldimumo diagrama DiagonalMovement:



Bendradarbiavimo diagrama DiagonalMovement:



Vieši Metodai

- DiagonalMovement (double mult=1.0)
  Construct a new Diagonal Movement object.
- void move (int &x, int &y, int dx, int dy) override
  Move the animal by (dx,dy), updating x,y in place.
- char typeTag () const override
  Return a tag to identify concrete type in the stream.

Vieši Metodai inherited from SpeedBasedMovement

- SpeedBasedMovement (double mult=1.0)

  Construct a new Speed Based Movement object.
- void serialize (std::ostream &os) const override

  Serialize strategy-specific data to a binary stream.
- void deserialize (std::istream &is) override

  Deserialize strategy-specific data from a binary stream.
- double getSpeedMultiplier () const

  Get the speed multiplier.
- void setSpeedMultiplier (double m)

  Set the speed multiplier.

Vieši Metodai inherited from MovementStrategy

- virtual ~MovementStrategy ()=default

  Destroy the Movement Strategy object.

Additional Inherited Members

Apsaugoti Atributai inherited from SpeedBasedMovement

- double speedMultiplier_

4.3.1 Smulkus aprašymas

Allows diagonal moves at constant cost.

4.3.2 Konstruktoriaus ir Destruktoriaus Dokumentacija

4.3.2.1 DiagonalMovement()

```
DiagonalMovement::DiagonalMovement (
            double mult = 1.0)  [inline], [explicit]
```

Construct a new Diagonal Movement object.

Parametrai

mult	initial speed multiplier, default = 1.0

4.3.3 Metodų Dokumentacija

4.3.3.1 move()

```
void DiagonalMovement::move (
            int & x,
            int & y,
            int dx,
            int dy)  [inline], [override], [virtual]
```

Move the animal by (dx,dy), updating x,y in place.

Parametrai

x	int& x coordinate of the animal
y	int& y coordinate of the animal
dx	int x displacement
dy	int y displacement

Realizuoja MovementStrategy.

4.3.3.2 typeTag()

```
char DiagonalMovement::typeTag () const  [inline], [override], [virtual]
```

Return a tag to identify concrete type in the stream.

Gražina

char

Realizuoja MovementStrategy.

Dokumentacija šiai klasei sugeneruota iš šio failo:

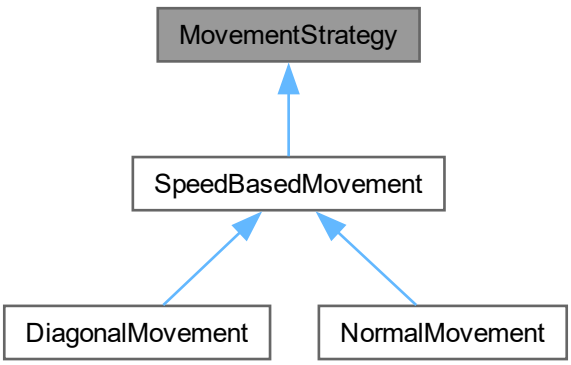
- src/movement/DiagonalMovement.h

4.4 MovementStrategy Klasė

Abstract interface for animal movement behavior.

```
#include <MovementStrategy.h>
```

Paveldimumo diagrama MovementStrategy:



Vieši Metodai

- virtual ~MovementStrategy ()=default

  Destroy the Movement Strategy object.
- virtual void move (int &x, int &y, int dx, int dy)=0

  Move the animal by (dx,dy), updating x,y in place.
- virtual void serialize (std::ostream &os) const =0

  Serialize strategy-specific data to a binary stream.
- virtual void deserialize (std::istream &is)=0

  Deserialize strategy-specific data from a binary stream.
- virtual char typeTag () const =0

  Return a tag to identify concrete type in the stream.

4.4.1 Smulkus aprašymas

Abstract interface for animal movement behavior.

4.4.2 Konstruktoriaus ir Destruktoriaus Dokumentacija

4.4.2.1 ~MovementStrategy()

```
virtual MovementStrategy::~MovementStrategy ()  [virtual], [default]
```

Destroy the Movement Strategy object.

4.4.3 Metodų Dokumentacija

4.4.3.1 deserialize()

```
virtual void MovementStrategy::deserialize (
            std::istream & is)  [pure virtual]
```

Deserialize strategy-specific data from a binary stream.

Parametrai

is	input stream to read from

Realizuota SpeedBasedMovement.

4.4.3.2 move()

```
virtual void MovementStrategy::move (
            int & x,
            int & y,
            int dx,
            int dy)  [pure virtual]
```

Move the animal by (dx,dy), updating x,y in place.

Parametrai

x	int& x coordinate of the animal
y	int& y coordinate of the animal
dx	int x displacement
dy	int y displacement

Realizuota DiagonalMovement, ir NormalMovement.

4.4.3.3 serialize()

```
virtual void MovementStrategy::serialize (
            std::ostream & os) const  [pure virtual]
```

Serialize strategy-specific data to a binary stream.

Parametrai

os	output stream to write to

Realizuota SpeedBasedMovement.

4.4.3.4 typeTag()

```
virtual char MovementStrategy::typeTag () const  [pure virtual]
```

Return a tag to identify concrete type in the stream.

Gražina

char

Realizuota DiagonalMovement, ir NormalMovement.

Dokumentacija šiai klasei sugeneruota iš šio failo:

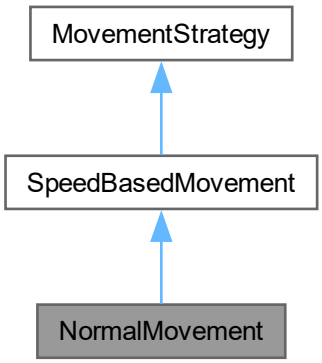
- src/movement/MovementStrategy.h

4.5 NormalMovement Klasė

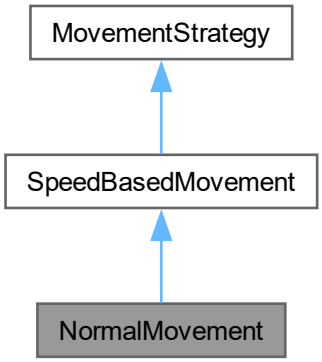
Simple axis-aligned movement.

`#include <NormalMovement.h>`

Paveldimumo diagrama NormalMovement:



Bendradarbiavimo diagrama NormalMovement:



Vieši Metodai

- NormalMovement (double mult=1.0)

  Construct a new Normal Movement object.
- void move (int &x, int &y, int dx, int dy) override

  Move the animal by (dx,dy), updating x,y in place.
- char typeTag () const override

  Return a tag to identify concrete type in the stream.

Vieši Metodai inherited from SpeedBasedMovement

- SpeedBasedMovement (double mult=1.0)

  Construct a new Speed Based Movement object.
- void serialize (std::ostream &os) const override

  Serialize strategy-specific data to a binary stream.
- void deserialize (std::istream &is) override

  Deserialize strategy-specific data from a binary stream.
- double getSpeedMultiplier () const

  Get the speed multiplier.
- void setSpeedMultiplier (double m)

  Set the speed multiplier.

Vieši Metodai inherited from MovementStrategy

- virtual ~MovementStrategy ()=default

  Destroy the Movement Strategy object.

Additional Inherited Members

Apsaugoti Atributai inherited from SpeedBasedMovement

- double speedMultiplier_

4.5.1 Smulkus aprašymas

Simple axis-aligned movement.

4.5.2 Konstruktoriaus ir Destruktoriaus Dokumentacija

4.5.2.1 NormalMovement()

```
NormalMovement::NormalMovement (
            double mult = 1.0)  [inline], [explicit]
```

Construct a new Normal Movement object.

Parametrai

mult	initial speed multiplier, default = 1.0

4.5.3 Metodų Dokumentacija

4.5.3.1 move()

```
void NormalMovement::move (
            int & x,
            int & y,
            int dx,
            int dy)  [inline], [override], [virtual]
```

Move the animal by (dx,dy), updating x,y in place.

Parametrai

x	int& x coordinate of the animal
y	int& y coordinate of the animal
dx	int x displacement
dy	int y displacement

Realizuoja MovementStrategy.

4.5.3.2 typeTag()

```
char NormalMovement::typeTag () const  [inline], [override], [virtual]
```

Return a tag to identify concrete type in the stream.

Gražina

char

Realizuoja MovementStrategy.

Dokumentacija šiai klasei sugeneruota iš šio failo:

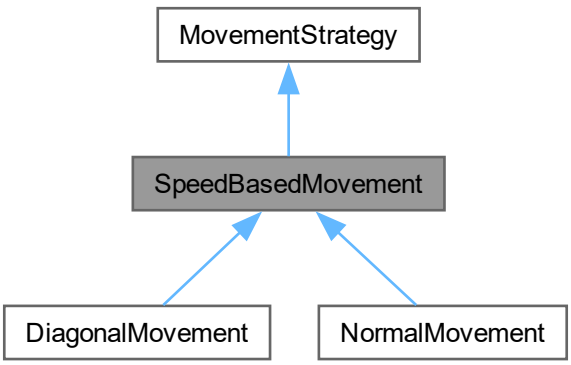
- src/movement/NormalMovement.h

4.6 SpeedBasedMovement Klasė

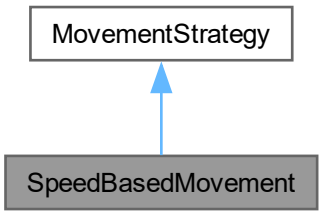
Base class for strategies that share a speedMultiplier.

```
#include <SpeedBasedMovement.h>
```

Paveldimumo diagrama SpeedBasedMovement:



Bendradarbiavimo diagrama SpeedBasedMovement:



Vieši Metodai

- SpeedBasedMovement (double mult=1.0)

  Construct a new Speed Based Movement object.
- void serialize (std::ostream &os) const override

  Serialize strategy-specific data to a binary stream.
- void deserialize (std::istream &is) override

  Deserialize strategy-specific data from a binary stream.
- double getSpeedMultiplier () const

  Get the speed multiplier.
- void setSpeedMultiplier (double m)

  Set the speed multiplier.

Vieši Metodai inherited from MovementStrategy

- virtual ~MovementStrategy ()=default

  Destroy the Movement Strategy object.
- virtual void move (int &x, int &y, int dx, int dy)=0

  Move the animal by (dx,dy), updating x,y in place.
- virtual char typeTag () const =0

  Return a tag to identify concrete type in the stream.

Apsaugoti Atributai

- double speedMultiplier_

4.6.1 Smulkus aprašymas

Base class for strategies that share a speedMultiplier.

4.6.2 Konstruktoriaus ir Destruktoriaus Dokumentacija

4.6.2.1 SpeedBasedMovement()

```
SpeedBasedMovement::SpeedBasedMovement (
            double mult = 1.0)  [inline], [explicit]
```

Construct a new Speed Based Movement object.

Parametrai

mult	speed multiplier, default = 1.0

4.6.3 Metodų Dokumentacija

4.6.3.1 deserialize()

```
void SpeedBasedMovement::deserialize (
            std::istream & is)  [inline], [override], [virtual]
```

Deserialize strategy-specific data from a binary stream.

Parametrai

is	input stream to read from

Realizuoja MovementStrategy.

4.6.3.2 getSpeedMultiplier()

```
double SpeedBasedMovement::getSpeedMultiplier () const  [inline]
```

Get the speed multiplier.

Gražina

double

4.6.3.3 serialize()

```
void SpeedBasedMovement::serialize (
            std::ostream & os) const  [inline], [override], [virtual]
```

Serialize strategy-specific data to a binary stream.

Parametrai

os	output stream to write to

Realizuoja MovementStrategy.

4.6.3.4 setSpeedMultiplier()

```
void SpeedBasedMovement::setSpeedMultiplier (
            double m)  [inline]
```

Set the speed multiplier.

Parametrai

m	

4.6.4 Atributų Dokumentacija

4.6.4.1 speedMultiplier_

```
double SpeedBasedMovement::speedMultiplier_  [protected]
```

Dokumentacija šiai klasei sugeneruota iš šio failo:

- src/movement/SpeedBasedMovement.h

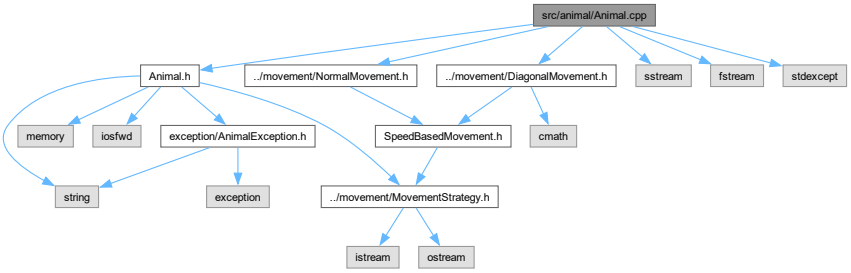
skyrius 5

Failo Dokumentacija

5.1 src/animal/Animal.cpp Failo Nuoroda

```
#include "Animal.h"
#include "../movement/NormalMovement.h"
#include "../movement/DiagonalMovement.h"
#include <sstream>
#include <fstream>
#include <stdexcept>
```

Įtraukimo priklausomybių diagrama Animal.cpp:



Funkcijos

- std::ostream & operator<< (std::ostream &os, const Animal &a)
- std::istream & operator>> (std::istream &is, Animal &a)

5.1.1 Smulkus aprašymas

Autorius

Rokas Braidokas (rokasbraidokas@gmail.com)

Versija

0.1

Data

2025-05-06

Copyright

Copyright (c) 2025

5.1.2 Funkcijos Dokumentacija

5.1.2.1 operator<<()

```
std::ostream & operator<< (
            std::ostream & os,
            const Animal & a)
```

Parametrai

os	output stream
a	animal to write

Gražina

std::ostream&

5.1.2.2 operator>>()

```
std::istream & operator>> (
            std::istream & is,
            Animal & a)
```

Parametrai

is	input stream
a	animal to read into

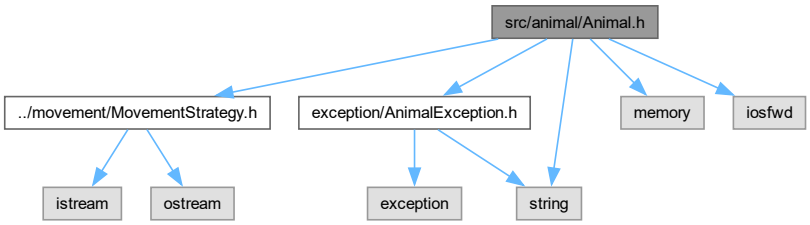
Gražina

std::istream&

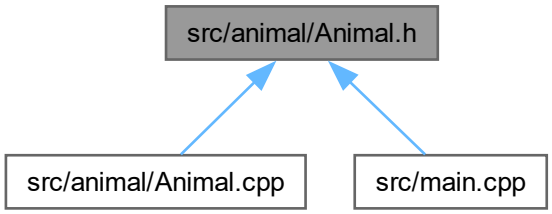
5.2 src/animal/Animal.h Failo Nuoroda

```
#include "../movement/MovementStrategy.h"
#include "exception/AnimalException.h"
#include <memory>
#include <string>
#include <iosfwd>
```

Įtraukimo priklausomybių diagrama Animal.h:



Šis grafas rodo, kuris failas tiesiogiai ar netiesiogiai įtraukia šį failą:



Klasės

- class Animal

  Represents an Animal with replaceable movement.

5.2.1 Smulkus aprašymas

Autorius

Rokas Braidokas (rokasbraidokas@gmail.com)

Versija

0.1

Data

2025-05-06

Copyright

5.3 Animal.h

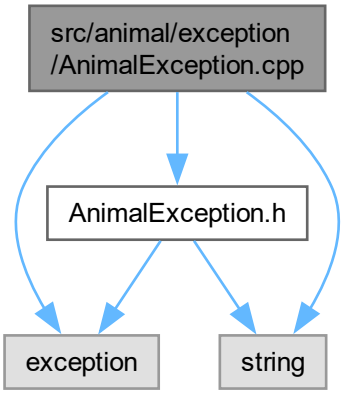
Eiti į šio failo dokumentaciją.

```
00001
00011 #ifndef ANIMAL_H
00012 #define ANIMAL_H
00013
00014 #include "../movement/MovementStrategy.h"
00015 #include "exception/AnimalException.h"
00016 #include <memory>
00017 #include <string>
00018 #include <iosfwd>
00019
00024 class Animal {
00025     const int ID_;
00026     int x_, y_;
00027     std::string name_;
00028     std::unique_ptr<MovementStrategy> mover_;
00029
00030     static int nextID_;
00031 public:
00038     Animal(const std::string& name,
00039            std::unique_ptr<MovementStrategy> strat);
00040
00046     Animal(const Animal& o);
00053     Animal& operator=(const Animal& o);
00054
00061     void move(int dx, int dy);
00062
00068     void setMovementStrategy(std::unique_ptr<MovementStrategy> strat);
00069
00075     int getID()        const;
00081     int getX()         const;
00087     int getY()         const;
00093     std::string getName() const;
00094
00102     friend std::ostream& operator<<(std::ostream& os, const Animal& a);
00110     friend std::istream& operator>>(std::istream& is, Animal& a);
00111 };
00112
00113 #endif // ANIMAL_H
```

5.4 src/animal/exception/AnimalException.cpp Failo Nuoroda

```
#include "AnimalException.h"
#include <string>
#include <exception>
```

Įtraukimo priklausomybių diagrama AnimalException.cpp:



5.4.1 Smulkus aprašymas

Autorius

Rokas Braidokas (rokasbraidokas@gmail.com)

Versija

0.1

Data

2025-05-09

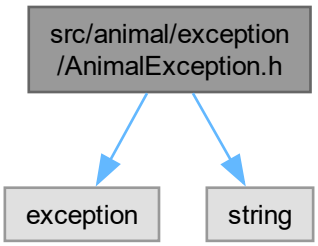
Copyright

Copyright (c) 2025

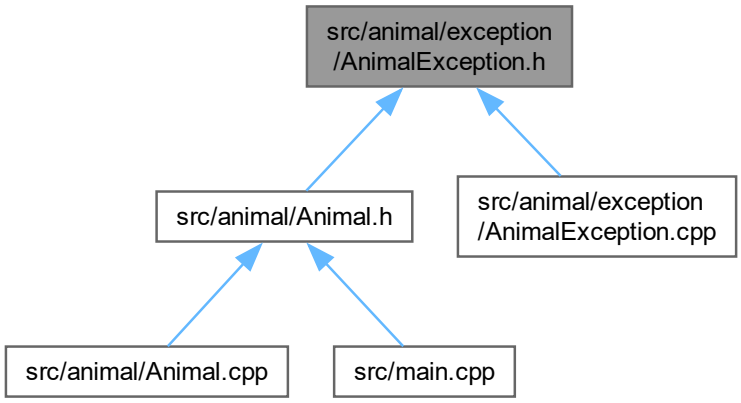
5.5 src/animal/exception/AnimalException.h Failo Nuoroda

```
#include <exception>
#include <string>
```

Įtraukimo priklausomybių diagrama AnimalException.h:



Šis grafas rodo, kuris failas tiesiogiai ar netiesiogiai įtraukia šį failą:



Klasės

- class AnimalException

5.5.1 Smulkus aprašymas

Autorius

Rokas Braidokas (rokasbraidokas@gmail.com)

Versija

0.1

Data

2025-05-09

Copyright

Copyright (c) 2025

5.6 AnimalException.h

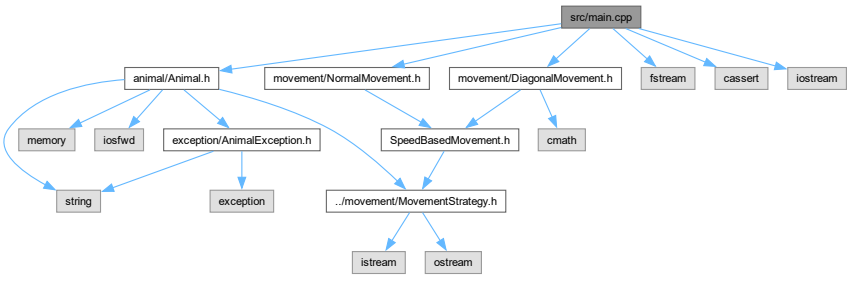
Eiti į šio failo dokumentaciją.

```
00001
00011 #ifndef ANIMALEXCEPTION_H
00012 #define ANIMALEXCEPTION_H
00013
00014 #include <exception>
00015 #include <string>
00016
00017 class AnimalException : public std::exception {
00018 private:
00019     std::string message;
00020 public:
00026     explicit AnimalException(const std::string& msg);
00032     const char* what() const noexcept override;
00033 };
00034
00035 #endif
```

5.7 src/main.cpp Failo Nuoroda

```
#include "animal/Animal.h"
#include "movement/NormalMovement.h"
#include "movement/DiagonalMovement.h"
#include <fstream>
#include <cassert>
#include <iostream>
```

Įtraukimo priklausomybių diagrama main.cpp:



Funkcijos

- int main ()

5.7.1 Smulkus aprašymas

Autorius

Rokas Braidokas (rokasbraidokas@gmail.com)

Versija

0.1

Data

2025-05-06

Copyright

Copyright (c) 2025

5.7.2 Funkcijos Dokumentacija

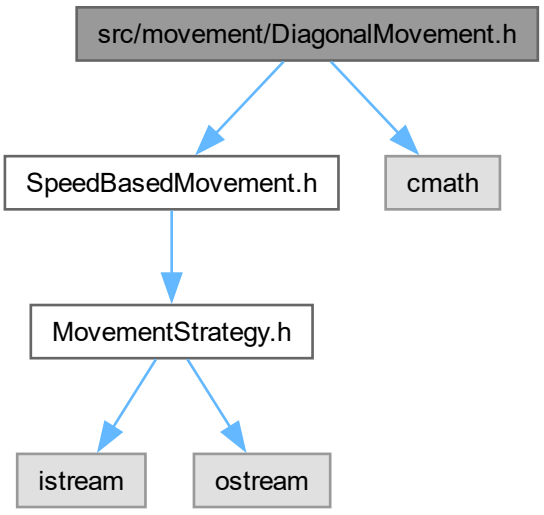
5.7.2.1 main()

```
int main ()
```

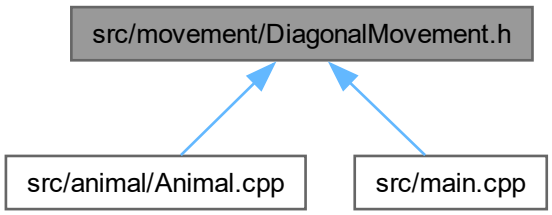
5.8 src/movement/DiagonalMovement.h Failo Nuoroda

```
#include "SpeedBasedMovement.h"
#include <cmath>
```

Įtraukimo priklausomybių diagrama DiagonalMovement.h:



Šis grafas rodo, kuris failas tiesiogiai ar netiesiogiai įtraukia šį failą:



Klasės

- class DiagonalMovement

  Allows diagonal moves at constant cost.

5.8.1 Smulkus aprašymas

Autorius

Rokas Braidokas (rokasbraidokas@gmail.com)

Versija

0.1

Data

2025-05-06

Copyright

Copyright (c) 2025

5.9 DiagonalMovement.h

Eiti į šio failo dokumentaciją.

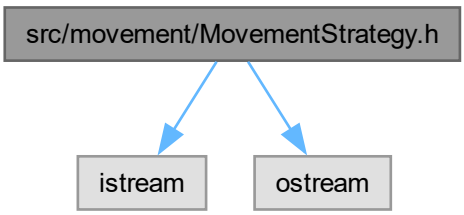
```
00001
00011 #ifndef DIAGONALMOVEMENT_H
00012 #define DIAGONALMOVEMENT_H
00013
00014 #include "SpeedBasedMovement.h"
00015 #include <cmath>
00016
00021 class DiagonalMovement : public SpeedBasedMovement {
00022 public:
00028     explicit DiagonalMovement(double mult = 1.0)
00029       : SpeedBasedMovement(mult)
```

```
00030     {}
00039     void move(int& x, int& y, int dx, int dy) override {
00040         double dist = std::hypot(dx, dy);
00041         if (dist > 0) {
00042             x += static_cast<int>((dx / dist) * speedMultiplier_);
00043             y += static_cast<int>((dy / dist) * speedMultiplier_);
00044         }
00045     }
00046
00052     char typeTag() const override { return 'D'; }
00053 };
00054
00055 #endif // DIAGONALMOVEMENT_H
```

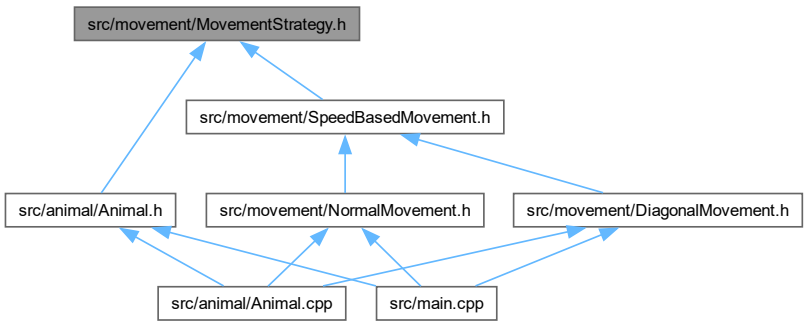
5.10 src/movement/MovementStrategy.h Failo Nuoroda

```
#include <istream>
#include <ostream>
```

Įtraukimo priklausomybių diagrama MovementStrategy.h:



Šis grafas rodo, kuris failas tiesiogiai ar netiesiogiai įtraukia šį failą:



Klasės

- class MovementStrategy

  Abstract interface for animal movement behavior.

5.10.1 Smulkus aprašymas

Autorius

Rokas Braidokas (rokasbraidokas@gmail.com)

Versija

0.1

Data

2025-05-06

Copyright

5.11 MovementStrategy.h

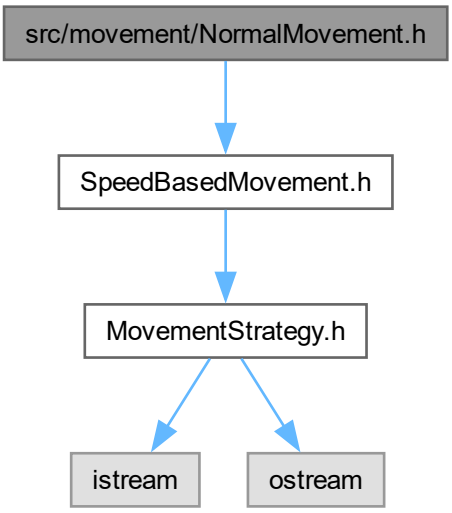
Eiti į šio failo dokumentaciją.

```
00001
00011 #ifndef MOVEMENTSTRATEGY_H
00012 #define MOVEMENTSTRATEGY_H
00013
00014 #include <istream>
00015 #include <ostream>
00016
00018 class MovementStrategy {
00019 public:
00024     virtual ~MovementStrategy() = default;
00025
00034     virtual void move(int& x, int& y, int dx, int dy) = 0;
00035
00041     virtual void serialize(std::ostream& os) const = 0;
00042
00048     virtual void deserialize(std::istream& is) = 0;
00049
00055     virtual char typeTag() const = 0;
00056 };
00057
00058 #endif // MOVEMENTSTRATEGY_H
```

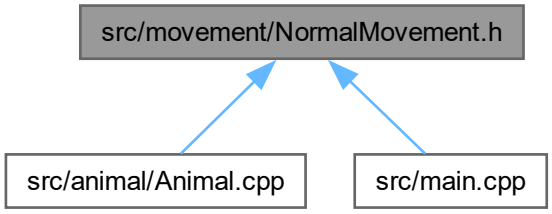
5.12 src/movement/NormalMovement.h Failo Nuoroda

```
#include "SpeedBasedMovement.h"
```

Įtraukimo priklausomybių diagrama NormalMovement.h:



Šis grafas rodo, kuris failas tiesiogiai ar netiesiogiai įtraukia šį failą:



Klasės

- class NormalMovement

  Simple axis-aligned movement.

5.12.1 Smulkus aprašymas

Autorius

Rokas Braidokas (rokasbraidokas@gmail.com)

Versija

0.1

Data

2025-05-09

Copyright

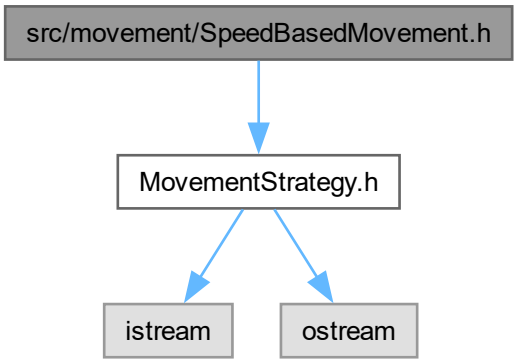
5.13 NormalMovement.h

Eiti į šio failo dokumentaciją.

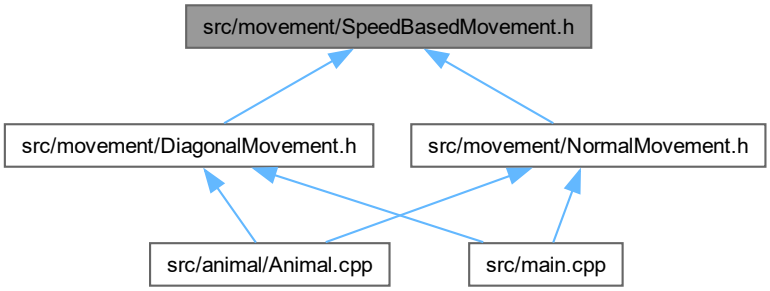
```
00001
00011 #ifndef NORMALMOVEMENT_H
00012 #define NORMALMOVEMENT_H
00013
00014 #include "SpeedBasedMovement.h"
00015
00020 class NormalMovement : public SpeedBasedMovement {
00021 public:
00027     explicit NormalMovement(double mult = 1.0)
00028       : SpeedBasedMovement(mult)
00029     {}
00038     void move(int& x, int& y, int dx, int dy) override {
00039         x += static_cast<int>(dx * speedMultiplier_);
00040         y += static_cast<int>(dy * speedMultiplier_);
00041     }
00042
00048     char typeTag() const override { return 'N'; }
00049 };
00050
00051 #endif // NORMALMOVEMENT_H
```

5.14 src/movement/SpeedBasedMovement.h Failo Nuoroda

```
#include "MovementStrategy.h"
```
Įtraukimo priklausomybių diagrama SpeedBasedMovement.h:



Šis grafas rodo, kuris failas tiesiogiai ar netiesiogiai įtraukia šį failą:



Klasės

- class SpeedBasedMovement

  Base class for strategies that share a speedMultiplier.

5.14.1 Smulkus aprašymas

Autorius

Rokas Braidokas (rokasbraidokas@gmail.com)

Versija

0.1

Data

2025-05-06

Copyright

5.15 SpeedBasedMovement.h

Eiti į šio failo dokumentaciją.

```
00001
00011 #ifndef SPEEDBASEDMOVEMENT_H
00012 #define SPEEDBASEDMOVEMENT_H
00013
00014 #include "MovementStrategy.h"
00015
00020 class SpeedBasedMovement : public MovementStrategy {
00021 protected:
00022     double speedMultiplier_;
00023
00024 public:
00030     explicit SpeedBasedMovement(double mult = 1.0)
00031       : speedMultiplier_(mult)
00032     {}
00033
00039     void serialize(std::ostream& os) const override {
00040         os.write(reinterpret_cast<const char*>(&speedMultiplier_), sizeof(speedMultiplier_));
00041     }
00042
00048     void deserialize(std::istream& is) override {
00049         is.read(reinterpret_cast<char*>(&speedMultiplier_), sizeof(speedMultiplier_));
00050     }
00051
00057     double getSpeedMultiplier() const { return speedMultiplier_; }
00058
00064     void setSpeedMultiplier(double m) { speedMultiplier_ = m; }
00065 };
00066
00067 #endif // SPEEDBASEDMOVEMENT_H
```

Rodyklė