K-średnich

v1.0

Wygenerowano przez Doxygen 1.8.14

Spis treści

1 Indeks hierarchiczny **1**

1.1 Hierarchia klas 1

2 Indeks klas **3**

2.1 Lista klas 3

3 Dokumentacja klas **5**

3.1 Dokumentacja klasy bmp::Bitmap 5

3.1.1 Opis szczegółowy 5

3.1.2 Dokumentacja konstruktora i destruktora 5

3.1.2.1 Bitmap() [1/2] 6

3.1.2.2 Bitmap() [2/2] 6

3.1.2.3 ~Bitmap() 6

3.1.3 Dokumentacja funkcji składowych 6

3.1.3.1 circle() 6

3.1.3.2 owieczka() 7

3.1.3.3 Pcircle() 7

3.1.3.4 setPixel() 7

3.1.3.5 write() 8

3.1.4 Dokumentacja atrybutów składowych 8

3.1.4.1 m_height 8

3.1.4.2 m_pPixel 8

3.1.4.3 m_width 9

3.2 Dokumentacja klasy bmp::BitmapFile 9

3.2.1 Opis szczegółowy . . . 9
3.2.2 Dokumentacja atrybutów składowych . . . 9
3.2.2.1 dataOffset . . . 9
3.2.2.2 fileSize . . . 9
3.2.2.3 header . . . 10
3.2.2.4 reserved . . . 10
3.3 Dokumentacja klasy bmp::BitmapInfo . . . 10
3.3.1 Opis szczegółowy . . . 10
3.4 Dokumentacja klasy Object . . . 11
3.4.1 Opis szczegółowy . . . 11
3.4.2 Dokumentacja konstruktora i destruktora . . . 11
3.4.2.1 Object() [1/2] . . . 12
3.4.2.2 Object() [2/2] . . . 12
3.4.2.3 ∼Object() . . . 12
3.4.3 Dokumentacja funkcji składowych . . . 12
3.4.3.1 getn() . . . 12
3.4.3.2 getname() . . . 13
3.4.3.3 getx() . . . 13
3.4.3.4 setn() . . . 13
3.4.3.5 toString() . . . 13
3.4.4 Dokumentacja przyjaciół i funkcji związanych . . . 14
3.4.4.1 Pasterz . . . 14
3.4.5 Dokumentacja atrybutów składowych . . . 14
3.4.5.1 n . . . 14
3.4.5.2 name . . . 14
3.4.5.3 x . . . 14
3.5 Dokumentacja klasy ObjectFactory . . . 14
3.5.1 Opis szczegółowy . . . 15
3.5.2 Dokumentacja funkcji składowych . . . 15
3.5.2.1 create() . . . 15

3.6 Dokumentacja klasy Owca . . . 15
3.6.1 Opis szczegółowy . . . 16
3.6.2 Dokumentacja konstruktora i destruktora . . . 16
3.6.2.1 Owca() [1/2] . . . 16
3.6.2.2 Owca() [2/2] . . . 16
3.6.3 Dokumentacja funkcji składowych . . . 16
3.6.3.1 FindNearestPasterz() . . . 17
3.6.3.2 toString() . . . 17
3.6.4 Dokumentacja atrybutów składowych . . . 17
3.6.4.1 wlasciciel . . . 17
3.7 Dokumentacja klasy Pasterz . . . 18
3.7.1 Opis szczegółowy . . . 18
3.7.2 Dokumentacja konstruktora i destruktora . . . 18
3.7.2.1 Pasterz() [1/2] . . . 19
3.7.2.2 Pasterz() [2/2] . . . 19
3.7.3 Dokumentacja funkcji składowych . . . 20
3.7.3.1 deleteStadko() . . . 20
3.7.3.2 FindMiddle() . . . 20
3.7.3.3 getliczbastadka() . . . 20
3.7.3.4 setliczba_stadka() . . . 20
3.7.3.5 showElementStadka() . . . 21
3.7.3.6 showStadko() . . . 21
3.7.3.7 toString() . . . 21
3.7.4 Dokumentacja przyjaciół i funkcji związanych . . . 21
3.7.4.1 Owca . . . 22
3.7.5 Dokumentacja atrybutów składowych . . . 22
3.7.5.1 liczba_stadka . . . 22
3.7.5.2 stadko . . . 22
3.8 Dokumentacja klasy Simulation . . . 22
3.8.1 Opis szczegółowy . . . 23

3.8.2 Dokumentacja konstruktora i destruktora . . . 23
3.8.2.1 Simulation() [1/2] . . . 23
3.8.2.2 Simulation() [2/2] . . . 23
3.8.2.3 ∼Simulation() . . . 23
3.8.3 Dokumentacja funkcji składowych . . . 23
3.8.3.1 CreateContainers() . . . 24
3.8.3.2 CreateFile() . . . 24
3.8.3.3 Go() . . . 24
3.8.3.4 pozwolenieDestruktor() . . . 24
3.8.4 Dokumentacja atrybutów składowych . . . 25
3.8.4.1 para . . . 25
3.8.4.2 x . . . 25
3.9 Dokumentacja klasy StadoOwiec . . . 25
3.9.1 Opis szczegółowy . . . 25
3.9.2 Dokumentacja konstruktora i destruktora . . . 25
3.9.2.1 StadoOwiec() . . . 25
3.9.3 Dokumentacja atrybutów składowych . . . 26
3.9.3.1 lista . . . 26
3.10 Dokumentacja klasy UserArguments . . . 26
3.10.1 Opis szczegółowy . . . 27
3.10.2 Dokumentacja konstruktora i destruktora . . . 27
3.10.2.1 UserArguments() . . . 27
3.10.3 Dokumentacja funkcji składowych . . . 27
3.10.3.1 CreateData() . . . 27
3.10.3.2 getn() . . . 27
3.10.3.3 getw() . . . 28
3.10.3.4 Help() . . . 28
3.10.3.5 LoadArguments() . . . 28
3.10.3.6 RecognizeDimension() . . . 28
3.10.3.7 VectorCentoridowZData() . . . 28

3.10.3.8 VectorDanych() . . . 29
3.10.4 Dokumentacja przyjaciół i funkcji związanych . . . 29
3.10.4.1 Simulation . . . 29
3.10.5 Dokumentacja atrybutów składowych . . . 29
3.10.5.1 d . . . 29
3.10.5.2 file . . . 29
3.10.5.3 k . . . 30
3.10.5.4 m . . . 30
3.10.5.5 n . . . 30
3.10.5.6 w . . . 30
3.11 Dokumentacja klasy ZoorganizowanaGrupaPasterzy . . . 30
3.11.1 Opis szczegółowy . . . 30
3.11.2 Dokumentacja konstruktora i destruktora . . . 30
3.11.2.1 ZoorganizowanaGrupaPasterzy() . . . 30
3.11.3 Dokumentacja atrybutów składowych . . . 31
3.11.3.1 lista . . . 31

Indeks **33**

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

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

bmp::Bitmap 5
bmp::BitmapFile 9
bmp::BitmapInfo 10
Object 11
  Owca 15
  Pasterz 18
ObjectFactory 14
Simulation 22
StadoOwiec 25
UserArguments 26
ZoorganizowanaGrupaPasterzy 30

Rozdział 2

Indeks klas

2.1 Lista klas

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

bmp::Bitmap 5
bmp::BitmapFile 9
bmp::BitmapInfo 10
Object 11
ObjectFactory 14
Owca 15
Pasterz 18
Simulation 22
StadoOwiec 25
UserArguments 26
ZoorganizowanaGrupaPasterzy 30

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja klasy bmp::Bitmap

`#include <Bitmap.h>`

Metody publiczne

- Bitmap ()
- Bitmap (int width, int height)
- void setPixel (int x, int y, uint8_t red, uint8_t green, uint8_t blue)
- void circle (int x, int y, uint8_t red, uint8_t green, uint8_t blue)
- void Pcircle (int x, int y, uint8_t red, uint8_t green, uint8_t blue)
- void owieczka (int x, int y, uint8_t red, uint8_t green, uint8_t blue)
- bool write (const std::string &filename)
- ~Bitmap ()

Atrybuty prywatne

- int m_width
- int m_height
- std::unique_ptr< uint8_t[ ]> m_pPixel

3.1.1 Opis szczegółowy

Klasa reprezentująca Bitmapę. Określa się w niej jej wymiary, stawia się dzięki niej piksele w konkretnych punktach oraz pozwala na zapisanie do pliku o rozszerzeniu ".bmp" ustawionych pikseli.

3.1.2 Dokumentacja konstruktora i destruktora

3.1.2.1 Bitmap() [1/2]

```
Bitmap::Bitmap ( )
```

Konstruktor inicjalizujący domyślne składowe bitmapy.

3.1.2.2 Bitmap() [2/2]

```
Bitmap::Bitmap (
            int width,
            int height )
```

Konstruktor inicjalizujący składowe bitmapy.

Parametry

width	Szerokość bitmapy.
height	Wysokość bitmapy.

3.1.2.3 ∼Bitmap()

```
Bitmap::∼Bitmap ( )
```

Domyślny destruktor bitmapy.

3.1.3 Dokumentacja funkcji składowych

3.1.3.1 circle()

```
void Bitmap::circle (
            int x,
            int y,
            uint8_t red,
            uint8_t green,
            uint8_t blue )
```

Metoda rysująca okrąg wokół danej pozycji, reprezentujący owieczkę .

Parametry

x	Współrzędna osi odciętych określająca środkowy piksel.
y	Współrzędna osi rzędnych określająca środkowy piksel.
red	Wartość koloru czerwonego zapisana na 8 bitach.
green	Wartość koloru zielonego zapisana na 8 bitach.
blue	Wartość koloru niebieskiego zapisana na 8 bitach.

3.1.3.2 owieczka()

```
void Bitmap::owieczka (
            int x,
            int y,
            uint8_t red,
            uint8_t green,
            uint8_t blue )
```

Metoda rysująca owieczkę której środek jest na danej pozycji.

Parametry

x	Współrzędna osi odciętych określająca środkowy piksel.
y	Współrzędna osi rzędnych określająca środkowy piksel.
red	Wartość koloru czerwonego zapisana na 8 bitach.
green	Wartość koloru zielonego zapisana na 8 bitach.
blue	Wartość koloru niebieskiego zapisana na 8 bitach.

3.1.3.3 Pcircle()

```
void Bitmap::Pcircle (
            int x,
            int y,
            uint8_t red,
            uint8_t green,
            uint8_t blue )
```

Metoda rysująca okrąg wokół danej pozycji, reprezentujący pasterza. Okrąg przypomina lekko order.

Parametry

x	Współrzędna osi odciętych określająca środkowy piksel.
y	Współrzędna osi rzędnych określająca środkowy piksel.
red	Wartość koloru czerwonego zapisana na 8 bitach.
green	Wartość koloru zielonego zapisana na 8 bitach.
blue	Wartość koloru niebieskiego zapisana na 8 bitach.

3.1.3.4 setPixel()

```
void Bitmap::setPixel (
            int x,
```

```
            int y,
            uint8_t red,
            uint8_t green,
            uint8_t blue )
```

Metoda ustawiająca piksel na danej pozycji.

Parametry

x	Współrzędna osi odciętych na której ma zostac postawiony piksel.
y	Współrzędna osi rzędnych na której ma zostac postawiony piksel.
red	Wartość koloru czerwonego zapisana na 8 bitach.
green	Wartość koloru zielonego zapisana na 8 bitach.
blue	Wartość koloru niebieskiego zapisana na 8 bitach.

3.1.3.5 write()

```
bool Bitmap::write (
            const std::string & filename )
```

Metoda tworząca plik z tablicy wszystkich pikseli. Tworzy również obiekty BitmapInfo i BitmapFile oraz definiuje ich wartości, potrzebne do utworzenia pliku.

Parametry

filename	Nazwa pliku do której ma być zapisana bitmapa.

Zwraca

Prawdę jeśli udało się utworzyć plik lub fałsz jeśli nie.

3.1.4 Dokumentacja atrybutów składowych

3.1.4.1 m_height

```
int bmp::Bitmap::m_height  [private]
```

Wysokość bitmapy. Specyfikuję ile pixeli ma być w pionie.

3.1.4.2 m_pPixel

```
std::unique_ptr<uint8_t[]> bmp::Bitmap::m_pPixel  [private]
```

Unikalny wskaźnik na tablicę wszystkich pikseli. Pozwala na ustawienie na konkretnej pozycji(pikselu) danego koloru. Kolor składa się z trzech uint8_t(red, green, blue (RGB)). Zmieniając wartości RGB w zakresie od 0 do 255 można korygować kolor.

3.1.4.3 m_width

```
int bmp::Bitmap::m_width  [private]
```

Szerokość bitmapy. Specyfikuję ile pixeli ma być w poziomie.

Dokumentacja dla tej klasy została wygenerowana z plików:

- Projekt/Bitmap.h
- Projekt/Bitmap.cpp

3.2 Dokumentacja klasy bmp::BitmapFile

```
#include <BitmapFile.h>
```

Atrybuty publiczne

- char header [2] { 'B','M' }
- int32_t fileSize
- int32_t reserved {0}
- int32_t dataOffset

3.2.1 Opis szczegółowy

Klasa reprezentująca dane pliku bitmapy takie jak np. rozmiar, rozmiar bez tablicy pikseli czy sygnatura.

3.2.2 Dokumentacja atrybutów składowych

3.2.2.1 dataOffset

```
int32_t bmp::BitmapFile::dataOffset
```

Rozmiar pliku bez tablicy pikseli.

3.2.2.2 fileSize

```
int32_t bmp::BitmapFile::fileSize
```

Całkowity rozmiar pliku.

3.2.2.3 header

```
char bmp::BitmapFile::header[2] { 'B','M' }
```

Sygnatura pliku określająca, że będzie to bitmapa.

3.2.2.4 reserved

```
int32_t bmp::BitmapFile::reserved {0}
```

Składowa rezerwująca potrzebna do utworzenia pliku.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- Projekt/BitmapFile.h

3.3 Dokumentacja klasy bmp::BitmapInfo

```
#include <BitmapInfo.h>
```

Atrybuty publiczne

- int32_t **headerSize** { 40 }
- int32_t **width**
- int32_t **height**
- int16_t **planes** { 1 }
- int16_t **bitsPerPixel** {24}
- int32_t **compression** {0}
- int32_t **dataSize** {0}
- int32_t **horizontalResolution** {2400}
- int32_t **verticalResolution** {2400}
- int32_t **colors** {0}
- int32_t **importantColors** {0}

3.3.1 Opis szczegółowy

Klasa reprezentująca dane potrzebne do stworzenia bitmapy (bity na piksel, szerokość, wysokość). W większości inicjalizowane domyślnie żeby móc określić rozmiar pliku. Jedynie szerokość i wysokość jest przypisywana potem, aby móc obliczyć rozmiar z tablicą pikseli i bez tablicy.

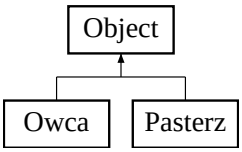
Dokumentacja dla tej klasy została wygenerowana z pliku:

- Projekt/BitmapInfo.h

3.4 Dokumentacja klasy Object

`#include <Object.h>`

Diagram dziedziczenia dla Object



Metody publiczne

- Object ()
- Object (const std::string &name, double ∗other)
- ∼Object ()
- virtual std::string toString ()=0
- double ∗ getx () const
- std::string getname ()

Statyczne metody publiczne

- static void setn (int a)
- static int getn ()

Atrybuty chronione

- double ∗ x
- std::string name

Statyczne atrybuty chronione

- static int n = 2

Przyjaciele

- class Pasterz

3.4.1 Opis szczegółowy

Klasa abstrakcyjna reprezentująca obiekt posiadający współrzędne o danym wymiarze i nazwę. Klasami pochodnymi jest klasa Pasterz oraz klasa Owca.

3.4.2 Dokumentacja konstruktora i destruktora

3.4.2.1 Object() [1/2]

```
Object::Object ( )
```

Konstruktor inicjalizujący domyślne składowe obiektu klasy Object.

3.4.2.2 Object() [2/2]

```
Object::Object (
            const std::string & name,
            double * other )
```

Konstruktor tworzący głęboką kopię tablicy współrzędnych oraz inicjalizujący nazwę.

Parametry

name	Nazwa obiektu dziedziczącego po Object.
other	Tablica współrzędnych z której ma zostać stworzona głeboka kopia.

3.4.2.3 ~Object()

```
Object::~Object ( )
```

Domyślny destruktor klasy Object.

3.4.3 Dokumentacja funkcji składowych

3.4.3.1 getn()

```
int Object::getn ( )  [static]
```

Statyczny getter służący zajrzenia w jakim wymiarze program będzie realizowany.

Zwraca

Wymiar realizowania programu.

3.4.3.2 getname()

```
std::string Object::getname ( )
```

Getter służący do podejrzenia nazwy danego obiektu.

Zwraca

Nazwa obiektu.

3.4.3.3 getx()

```
double * Object::getx ( ) const
```

Getter służący do podejrzenia współrzędnych danego obiektu.

Zwraca

Wskaźnik na tablicę współrzędnych typu double.

3.4.3.4 setn()

```
void Object::setn (
            int a )  [static]
```

Statyczny setter służący ustawiania składowej określającej wymiar w jakim program będzie realizowany.

Parametry

a	Wartość na jaką seter ma ustawić wymiar.

3.4.3.5 toString()

```
virtual std::string Object::toString ( )  [pure virtual]
```

Metoda czysto wirtualna służąca do reprezentacji składowych danej klasy dziedziczącej (Owca lub Pasterz) w postaci tekstu.

Zwraca

Obliguje metody dziedziczące do zwrócenia łańcucha znaków.

Implementowany w Pasterz i Owca.

3.4.4 Dokumentacja przyjaciół i funkcji związanych

3.4.4.1 Pasterz

```
friend class Pasterz  [friend]
```

Deklaracja przyjaźni klasy Object z klasą Pasterz. Potrzebna aby obiekt klasy Pasterz mógł zaglądać do składowych klasy Owca.

3.4.5 Dokumentacja atrybutów składowych

3.4.5.1 n

```
int Object::n = 2  [static], [protected]
```

Wymiar w jakim bedzie realizowany program.

3.4.5.2 name

```
std::string Object::name  [protected]
```

Nazwa obiektu.

3.4.5.3 x

```
double* Object::x  [protected]
```

Tablica liczb zmiennoprzecinkowych podwójnej precyzji reperezentująca współrzędne obiektów.

Dokumentacja dla tej klasy została wygenerowana z plików:

- Projekt/Object.h
- Projekt/Object.cpp

3.5 Dokumentacja klasy ObjectFactory

```
#include <ObjectFactory.h>
```

Statyczne metody publiczne

- static Object * create (const std::string &s)

3.5.1 Opis szczegółowy

Klasa reprezentująca generyczną fabrykę obiektów klasy bazowej (Object)

3.5.2 Dokumentacja funkcji składowych

3.5.2.1 create()

```
Object * ObjectFactory::create (
            const std::string & s )  [static]
```

Statyczna metoda tworząca obiekt wg linii jakiej do niej podamy. Wiedząc jaki jest wymiar, rozpoznaje czy ma stworzyć obiekt klasy Pasterz czy Owca oraz ustawia ich składowe.

Parametry

s	Tekst z którego ma zostać stworzony obiekt.

Zwraca

Czysty wskaźnik na obiekt typu Object.

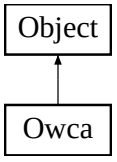
Dokumentacja dla tej klasy została wygenerowana z plików:

- Projekt/ObjectFactory.h
- Projekt/ObjectFactory.cpp

3.6 Dokumentacja klasy Owca

```
#include <Owca.h>
```

Diagram dziedziczenia dla Owca



Metody publiczne

- Owca ()
- Owca (const std::string &name, double ∗other)
- std::string toString () override
- bool FindNearestPasterz (const ZoorganizowanaGrupaPasterzy &pasterze)

Atrybuty prywatne

- Pasterz ∗ wlasciciel

Dodatkowe Dziedziczone Składowe

3.6.1 Opis szczegółowy

Klasa reprezentująca element zestawu danych, zwany owcą czy też owieczką. Jest klasą pochodną po klasie Object, więc posiada jej składowe oraz metody.

3.6.2 Dokumentacja konstruktora i destruktora

3.6.2.1 Owca() [1/2]

```
Owca::Owca ( )
```

Konstruktor inicjalizujący domyślne składowe obiektu klasy Owca.

3.6.2.2 Owca() [2/2]

```
Owca::Owca (
            const std::string & name,
            double * other )
```

Konstruktor tworzący głęboką kopię tablicy współrzędnych oraz inicjalizujący nazwę dla klasy Owca.

Parametry

name	Nazwa obiektu klasy Owca.
other	Tablica współrzędnych z której ma zostać stworzona głeboka kopia.

3.6.3 Dokumentacja funkcji składowych

3.6.3.1 FindNearestPasterz()

```
bool Owca::FindNearestPasterz (
            const ZoorganizowanaGrupaPasterzy & pasterze )
```

Metoda szukająca najbliższego pasterza. Jeśli owca znajdzie bliższego niż poprzednio to ustawia go jako właściciela i informuje o tym zwrotem z funkcji.

Parametry

pasterze	Klasa reprezentująca listę unikalnych wskaźników na obiekty klasy Pasterz.

Zwraca

Zwraca prawdę jeśli owieczka zmieniła swojego poprzedniego właściciela lub fałsz jeśli nie.

3.6.3.2 toString()

```
std::string Owca::toString ( )  [override], [virtual]
```

Metoda wirtualna (polimorfizm) służąca do reprezentacji składowych klasy Owca w postaci tekstu. Oprócz składowych podanych w klasie bazowej wyświetla również właściciela.

Zwraca

Tekst opisujący owcę.

Implementuje Object.

3.6.4 Dokumentacja atrybutów składowych

3.6.4.1 wlasciciel

```
Pasterz* Owca::wlasciciel  [private]
```

Wskaźnik informujący, który pasterz jest właścicielem.

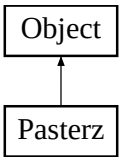
Dokumentacja dla tej klasy została wygenerowana z plików:

- Projekt/Owca.h
- Projekt/Owca.cpp

3.7 Dokumentacja klasy Pasterz

`#include <Pasterz.h>`

Diagram dziedziczenia dla Pasterz



Metody publiczne

- Pasterz ()
- Pasterz (const std::string &name, double ∗other)
- std::string toString () override
- void FindMiddle ()
- std::string showElementStadka (int i)
- std::string showStadko ()
- void setliczba_stadka (int liczba_stadka)
- int getliczbastadka () const
- void deleteStadko ()

Atrybuty prywatne

- std::vector< Owca ∗ > stadko
- int liczba_stadka

Przyjaciele

- class Owca

Dodatkowe Dziedziczone Składowe

3.7.1 Opis szczegółowy

Klasa reprezentująca element wg którego dzieli się dane, zwany pasterzem czy też centroidem. Jest klasą pochodną po klasie Object, więc posiada jej składowe oraz metody.

3.7.2 Dokumentacja konstruktora i destruktora

3.7.2.1 Pasterz() [1/2]

```
Pasterz::Pasterz ( )
```

Konstruktor inicjalizujący domyślne składowe obiektu klasy Pasterz.

3.7.2.2 Pasterz() [2/2]

```
Pasterz::Pasterz (
            const std::string & name,
            double * other )
```

Konstruktor tworzący głęboką kopię tablicy współrzędnych oraz inicjalizujący nazwę dla klasy Pasterz.

Parametry

name	Nazwa obiektu klasy Pasterz.
other	Tablica współrzędnych z której ma zostać stworzona głeboka kopia.

3.7.3 Dokumentacja funkcji składowych

3.7.3.1 deleteStadko()

```
void Pasterz::deleteStadko ( )
```

Metoda czyszcząca wektor stadka. Potrzebna przy symulacji, bo pasterz przy każdej iteracji sprawdza jak wygląda jego stadko na nowo, gdyż się przesunął.

3.7.3.2 FindMiddle()

```
void Pasterz::FindMiddle ( )
```

Metoda ustawiająca pasterza na środkowej pozycji wyliczonej ze średniej arytmetycznej współrzędnych każdej owieczki z jego stadka.

3.7.3.3 getliczbastadka()

```
int Pasterz::getliczbastadka ( ) const
```

Getter umożliwiający podgląd liczby stadka.

Zwraca

Liczba stadka.

3.7.3.4 setliczba_stadka()

```
void Pasterz::setliczba_stadka (
            int liczba_stadka )
```

Setter ustawiający liczbę stadka.

Parametry

liczba_stadka	Wartość jaką chce się ustawić jako liczbę stadka.

3.7.3.5 showElementStadka()

```
std::string Pasterz::showElementStadka (
            int i )
```

Metoda przedstawiająca owieczkę ze stadka w postaci stringa ułatwiającego wrzucenie danych do bitmapy.

Parametry

i	Numer owieczki w stadku.

Zwraca

Tekst ze współrzędnymi oddzielony spacjami.

3.7.3.6 showStadko()

```
std::string Pasterz::showStadko ( )
```

Metoda przedstawiająca całe stadko w postaci stringa ułatwiającego wrzucenie danych do pliku tekstowego.

Zwraca

Tekst ze współrzędnymi oddzielony spacjami i owieczkami oddzielonymi enterami.

3.7.3.7 toString()

```
std::string Pasterz::toString ( )  [override], [virtual]
```

Metoda wirtualna (polimorfizm) służąca do reprezentacji składowych klasy Pasterz w postaci tekstu. Oprócz składowych podanych w klasie bazowej wyświetla również stadko i jego liczbę.

Zwraca

Tekst opisujący pasterza.

Implementuje Object.

3.7.4 Dokumentacja przyjaciół i funkcji związanych

3.7.4.1 Owca

```
friend class Owca  [friend]
```

Deklaracja przyjaźni klasy Pasterz z klasą Owca. Aby owca przy znalezieniu bliższego pasterza, używając wskaźnika na właściciela mogła usunąć siebie ze stadka i zmniejszyć jego liczbę.

3.7.5 Dokumentacja atrybutów składowych

3.7.5.1 liczba_stadka

```
int Pasterz::liczba_stadka  [private]
```

Ilość owiec w stadzie.

3.7.5.2 stadko

```
std::vector<Owca*> Pasterz::stadko  [private]
```

Wektor zawierający wskaźniki na owieczki których pasterz jest właścicielem.

Dokumentacja dla tej klasy została wygenerowana z plików:

- Projekt/Pasterz.h
- Projekt/Pasterz.cpp

3.8 Dokumentacja klasy Simulation

```
#include <Simulation.h>
```

Metody publiczne

- Simulation ()
- Simulation (std::pair< StadoOwiec *, ZoorganizowanaGrupaPasterzy *> para)
- void Go ()
- void CreateFile (const std::string &output, bool d)
- void CreateContainers (UserArguments &arg)
- void pozwolenieDestruktor (bool x)
- ~Simulation ()

Atrybuty publiczne

- std::pair< StadoOwiec *, ZoorganizowanaGrupaPasterzy * > para

Atrybuty chronione

- bool x

3.8.1 Opis szczegółowy

Klasa reprezentująca symulację programu. Zawiera parę dwóch głównych kontenerów danych (stada owiec i grupę pasterzy), które wypełnia, dzięki którym wykonuje najistotniejszy algorytm programu i według których tworzy pliki wynikowe.

3.8.2 Dokumentacja konstruktora i destruktora

3.8.2.1 Simulation() [1/2]

```
Simulation::Simulation ( )
```

Konstruktor inicjalizujący domyślne składowe symulacji.

3.8.2.2 Simulation() [2/2]

```
Simulation::Simulation (
            std::pair< StadoOwiec *, ZoorganizowanaGrupaPasterzy *> para )
```

Konstruktor inicjalizujący parę inną parą podaną w parametrze.

Parametry

para	Para gotowych kontenerów z owieczkami oraz pasterzami.

3.8.2.3 ~Simulation()

```
Simulation::~Simulation ( )
```

Destruktor symulacji usuwa dynamicznie zaalokowane kontenery StadoOwiec i ZoorganizowanaGrupaPasterzy w momencie gdy mu się pozwoli.

3.8.3 Dokumentacja funkcji składowych

3.8.3.1 CreateContainers()

```
void Simulation::CreateContainers (
            UserArguments & arg )
```

Metoda wypełniająca kontenery wg ilości podanych w argumentach programu lub wg pliku wejsciowego.

Parametry

arg	Klasa reprezentująca argumenty podane przez użytkownika.

3.8.3.2 CreateFile()

```
void Simulation::CreateFile (
            const std::string & output,
            bool d )
```

Metoda tworząca plik tekstowy oraz jeżeli program realizowany jest dla dwóch wymiarów to również tworząca bitmapę.

Parametry

output	Ścieżka do której ma zostać zapisany plik wyjściowy.
d	Jeśli jest prawdą zostaną na bitmapie narysowane kółeczka jako dane, jeśli jest fałszem to owieczki.

3.8.3.3 Go()

```
void Simulation::Go ( )
```

Metoda rozpoczynająca algorytm grupowania owieczek.

3.8.3.4 pozwolenieDestruktor()

```
void Simulation::pozwolenieDestruktor (
            bool x )
```

Setter służący do ustawienia flagi x w celu pozwolenia na użycie destruktora.

Parametry

x	Jeśli prawda to pozwala, jeśli fałsz to nie.

3.8.4 Dokumentacja atrybutów składowych

3.8.4.1 para

```
std::pair<StadoOwiec*, ZoorganizowanaGrupaPasterzy*> Simulation::para
```

Para wkaźników na kontenery owiec i pasterzy.

3.8.4.2 x

```
bool Simulation::x  [protected]
```

Flaga pozwalająca wykonanie destruktora.

Dokumentacja dla tej klasy została wygenerowana z plików:

- Projekt/Simulation.h
- Projekt/Simulation.cpp

3.9 Dokumentacja klasy StadoOwiec

```
#include <StadoOwiec.h>
```

Metody publiczne

- StadoOwiec (std::vector< Object *> *sheeps)

Atrybuty publiczne

- std::list< std::unique_ptr< Owca > > lista

3.9.1 Opis szczegółowy

Klasa reprezentująca kontener obiektów klasy Owca.

3.9.2 Dokumentacja konstruktora i destruktora

3.9.2.1 StadoOwiec()

```
StadoOwiec::StadoOwiec (
            std::vector< Object *> * sheeps )
```

Konstruktor listy owiec. Zamienia wektor w listę.

Parametry

sheeps	Wektor owiec.

3.9.3 Dokumentacja atrybutów składowych

3.9.3.1 lista

```
std::list<std::unique_ptr<Owca> > StadoOwiec::lista
```

Lista owiec.

Dokumentacja dla tej klasy została wygenerowana z plików:

- Projekt/StadoOwiec.h
- Projekt/StadoOwiec.cpp

3.10 Dokumentacja klasy UserArguments

```
#include <UserArguments.h>
```

Metody publiczne

- UserArguments ()
- bool getw () const
- bool getn () const
- void Help ()
- bool LoadArguments (int argc, char **argv)
- void RecognizeDimension ()
- void CreateData (int mode)
- std::vector< Object * > VectorDanych (const std::string &vfile)
- std::vector< Object * > VectorCentoridowZData (std::vector< Object *> &data)

Atrybuty prywatne

- std::string file
- int d
- int m
- int k
- bool w
- bool n

Przyjaciele

- class Simulation

3.10.1 Opis szczegółowy

Klasa reprezentująca argumenty podane przez użytkownika.

3.10.2 Dokumentacja konstruktora i destruktora

3.10.2.1 UserArguments()

```
UserArguments::UserArguments ( )
```

Konstruktor inicjalizujący domyślne składowe klasy UserArguments.

3.10.3 Dokumentacja funkcji składowych

3.10.3.1 CreateData()

```
void UserArguments::CreateData (
            int mode )
```

Metoda tworząca dane wg składowych obiektu na rzecz którego jest wywoływana.

Parametry

mode	Określa co wygenerować ma metoda. Przyjmuje trzy wartości 1; 2 lub 3. 1-tworzy tylko owce, 2-tworzy tylko pasterzy, 3-tworzy owce i pasterzy.

3.10.3.2 getn()

```
bool UserArguments::getn ( ) const
```

Getter umożliwiający podgląd flagi n, odpowiedzialnej za tworzenie kropek zamiast owieczek w bitmapie.

Zwraca

Flagę n.

3.10.3.3 getw()

```
bool UserArguments::getw ( ) const
```

Getter umożliwiający podgląd flagi w, odpowiedzialnej za uruchamianie bez pliku.

Zwraca

Flagę w.

3.10.3.4 Help()

```
void UserArguments::Help ( )
```

Metoda wyświetlająca instrukcję w konsoli.

3.10.3.5 LoadArguments()

```
bool UserArguments::LoadArguments (
            int argc,
            char ** argv )
```

Metoda ustawiająca składowe klasy na podane przez użytkownika.

Parametry

argc	Ilość argumentów podana przez użytkownika.
argv	Tablica wyrazów jakie podano do konsoli.

Zwraca

Prawda jeśli udało się ustawić oraz fałsz jeśli wartości były nie poprawne.

3.10.3.6 RecognizeDimension()

```
void UserArguments::RecognizeDimension ( )
```

Metoda określająca wymiar w jakim będzie realizowany program na podstawie pliku wejściowego. Ustawia składową d na wymiar jaki rozpoznał.

3.10.3.7 VectorCentoridowZData()

```
std::vector< Object * > UserArguments::VectorCentoridowZData (
            std::vector< Object *> & data )
```

Metoda wyciągająca z wektora danych pasterzy (centroidów) sugerując się nazwą obiektu.

Parametry

data	Wektor z którego mają zostać wyciągnięte obiekty.

Zwraca

Wektor wyciągniętych centoridów.

3.10.3.8 VectorDanych()

```
std::vector< Object * > UserArguments::VectorDanych (
            const std::string & vfile )
```

Metoda zamieniająca plik wejściowy w wektor danych. Plik z którego mają zostać pobrane dane.

Zwraca

Wektor obiektów.

3.10.4 Dokumentacja przyjaciół i funkcji związanych

3.10.4.1 Simulation

```
friend class Simulation  [friend]
```

Deklaracja przyjaźni klasy UserArguments z klasą Simulation, aby symulacja wiedziała jakie argumenty podał użytkownik.

3.10.5 Dokumentacja atrybutów składowych

3.10.5.1 d

```
int UserArguments::d  [private]
```

Wymiar podany przez użytkownika. Domyślnie 2.

3.10.5.2 file

```
std::string UserArguments::file  [private]
```

Plik wejściowy podany przez użytkownika. Domyślnie ustawiony na "owceplik.txt".

3.10.5.3 k

```
int UserArguments::k  [private]
```

Ilość grup (k-średnich). Domyślnie 20 grup.

3.10.5.4 m

```
int UserArguments::m  [private]
```

Ilość danych, ile owiec ma się wygenerować. Domyślnie 500.

3.10.5.5 n

```
bool UserArguments::n  [private]
```

Flaga określająca czy bitmapa ma wyswietlic kropki jeśli prawda, jesli fałsz to owieczki. Domyślnie ustawiona na fałsz.

3.10.5.6 w

```
bool UserArguments::w  [private]
```

Flaga określająca czy uruchomić program bez pliku wejściowego jeśli prawda. Domyślnie ustawiona na fałsz.

Dokumentacja dla tej klasy została wygenerowana z plików:

- Projekt/UserArguments.h
- Projekt/UserArguments.cpp

3.11 Dokumentacja klasy ZoorganizowanaGrupaPasterzy

```
#include <ZoorganizowanaGrupaPasterzy.h>
```

Metody publiczne

- ZoorganizowanaGrupaPasterzy (std::vector< Object *> *pastors)

Atrybuty publiczne

- std::list< std::unique_ptr< Pasterz > > lista

3.11.1 Opis szczegółowy

Klasa reprezentująca kontener obiektów klasy Pasterz.

3.11.2 Dokumentacja konstruktora i destruktora

3.11.2.1 ZoorganizowanaGrupaPasterzy()

```
ZoorganizowanaGrupaPasterzy::ZoorganizowanaGrupaPasterzy (
            std::vector< Object *> * pastors )
```

Konstruktor listy pasterzy. Zamienia wektor w listę.

Parametry

pastors	Wektor pasterzy.

3.11.3 Dokumentacja atrybutów składowych

3.11.3.1 lista

```
std::list<std::unique_ptr<Pasterz> > ZoorganizowanaGrupaPasterzy::lista
```

Lista pasterzy.

Dokumentacja dla tej klasy została wygenerowana z plików:

- Projekt/ZoorganizowanaGrupaPasterzy.h
- Projekt/ZoorganizowanaGrupaPasterzy.cpp

Skorowidz

~Bitmap
    bmp::Bitmap, 6
~Object
    Object, 12
~Simulation
    Simulation, 23

Bitmap
    bmp::Bitmap, 5, 6
bmp::Bitmap, 5
    ~Bitmap, 6
    Bitmap, 5, 6
    circle, 6
    m_height, 8
    m_pPixel, 8
    m_width, 8
    owieczka, 7
    Pcircle, 7
    setPixel, 7
    write, 8
bmp::BitmapFile, 9
    dataOffset, 9
    fileSize, 9
    header, 9
    reserved, 10
bmp::BitmapInfo, 10

circle
    bmp::Bitmap, 6
create
    ObjectFactory, 15
CreateContainers
    Simulation, 23
CreateData
    UserArguments, 27
CreateFile
    Simulation, 24

d
    UserArguments, 29
dataOffset
    bmp::BitmapFile, 9
deleteStadko
    Pasterz, 20

file
    UserArguments, 29
fileSize
    bmp::BitmapFile, 9
FindMiddle
    Pasterz, 20
FindNearestPasterz
    Owca, 16

getliczbastadka
    Pasterz, 20
getn
    Object, 12
    UserArguments, 27
getname
    Object, 12
getw
    UserArguments, 27
getx
    Object, 13
Go
    Simulation, 24

header
    bmp::BitmapFile, 9
Help
    UserArguments, 28

k
    UserArguments, 29

liczba_stadka
    Pasterz, 22
lista
    StadoOwiec, 26
    ZoorganizowanaGrupaPasterzy, 31
LoadArguments
    UserArguments, 28

m
    UserArguments, 30
m_height
    bmp::Bitmap, 8
m_pPixel
    bmp::Bitmap, 8
m_width
    bmp::Bitmap, 8

n
    Object, 14
    UserArguments, 30
name
    Object, 14

Object, 11
    ~Object, 12

getn, 12
getname, 12
getx, 13
n, 14
name, 14
Object, 11, 12
Pasterz, 14
setn, 13
toString, 13
x, 14
ObjectFactory, 14
create, 15
Owca, 15
FindNearestPasterz, 16
Owca, 16
Pasterz, 21
toString, 17
wlasciciel, 17
owieczka
bmp::Bitmap, 7

para
Simulation, 25
Pasterz, 18
deleteStadko, 20
FindMiddle, 20
getliczbastadka, 20
liczba_stadka, 22
Object, 14
Owca, 21
Pasterz, 18, 19
setliczba_stadka, 20
showElementStadka, 21
showStadko, 21
stadko, 22
toString, 21
Pcircle
bmp::Bitmap, 7
pozwolenieDestruktor
Simulation, 24

RecognizeDimension
UserArguments, 28
reserved
bmp::BitmapFile, 10

setPixel
bmp::Bitmap, 7
setliczba_stadka
Pasterz, 20
setn
Object, 13
showElementStadka
Pasterz, 21
showStadko
Pasterz, 21
Simulation, 22
∼Simulation, 23
CreateContainers, 23
CreateFile, 24
Go, 24
para, 25
pozwolenieDestruktor, 24
Simulation, 23
UserArguments, 29
x, 25
stadko
Pasterz, 22
StadoOwiec, 25
lista, 26
StadoOwiec, 25

toString
Object, 13
Owca, 17
Pasterz, 21

UserArguments, 26
CreateData, 27
d, 29
file, 29
getn, 27
getw, 27
Help, 28
k, 29
LoadArguments, 28
m, 30
n, 30
RecognizeDimension, 28
Simulation, 29
UserArguments, 27
VectorCentoridowZData, 28
VectorDanych, 29
w, 30

VectorCentoridowZData
UserArguments, 28
VectorDanych
UserArguments, 29

w
UserArguments, 30
wlasciciel
Owca, 17
write
bmp::Bitmap, 8

x
Object, 14
Simulation, 25

ZoorganizowanaGrupaPasterzy, 30
lista, 31
ZoorganizowanaGrupaPasterzy, 30