

Nebuchadrezzar Podręcznik  
1.0.0.0

Wygenerowano przez Doxygen 1.5.1

Tue May 8 13:26:17 2007

## Spis treści

<a href="#">1 Nebuchadrezzar Indeks hierarchiczny</a>	1
<a href="#">2 Nebuchadrezzar Indeks klas</a>	2
<a href="#">3 Nebuchadrezzar Dokumentacja klas</a>	3

## 1 Nebuchadrezzar Indeks hierarchiczny

### 1.1 Nebuchadrezzar Hierarchia klas

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

<code>pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle</code>	3
<code>pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base</code>	7
<code>pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.MySqlBase</code>	26
<code>pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PostgresBase</code>	39
<code>pl.umk.mat.pz2006.nebuchadrezzar.marduk.Exporter</code>	16
<code>pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter</code>	39
<code>pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter</code>	31
<code>pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporter</code>	41
<code>pl.umk.mat.pz2006.nebuchadrezzar.marduk.TXTExporter</code>	43
<code>pl.umk.mat.pz2006.nebuchadrezzar.marduk.XMLExporter</code>	45
<code>pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXExporter</code>	20
<code>pl.umk.mat.pz2006.nebuchadrezzar.marduk.KMLExporter</code>	24
<code>pl.umk.mat.pz2006.nebuchadrezzar.marduk.GisInfo</code>	17
<code>pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread</code>	12
<code>pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXBase</code>	19
<code>pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXParser</code>	22
<code>pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Main</code>	26
<code>pl.umk.mat.pz2006.nebuchadrezzar.marduk.NmeaParser</code>	27
<code>pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineMerger</code>	33
<code>pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSimplifier</code>	34

<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSimplifier.SpecialTable</a>	37
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSmoother</a>	37
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.TXTParser</a>	44
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.XMLMarkers</a>	46

## 2 Nebuchadrezzar Indeks klas

### 2.1 Nebuchadrezzar Lista klas

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

<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle</a>	3
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base</a>	7
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread</a>	12
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.Exporter</a>	16
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.GisInfo</a>	17
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXBase</a>	19
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXExporter</a>	20
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXParser</a>	22
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.KMLExporter</a>	24
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Main</a>	26
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.MySQLBase</a>	26
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.NmeaParser</a>	27
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter</a>	31
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineMerger</a>	33
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSimplifier</a>	34
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSimplifier.SpecialTable</a>	37
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSmoother</a>	37
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PostgresBase</a>	39
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter</a>	39
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporter</a>	41
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.TXTExporter</a>	43

<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.TXTParser</a>	44
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.XMLExporter</a>	45
<a href="#">pl.umk.mat.pz2006.nebuchadrezzar.marduk.XMLMarkers</a>	46

## 3 Nebuchadrezzar Dokumentacja klas

### 3.1 Dokumentacja klasy `pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle`

#### Metody publiczne

- [Angle](#) (int angle)
- [Angle](#) (int angle, boolean nmea)
- void [fromDegreeMillion](#) (int angle)
- String [format](#) (String fmt)
- int [getSign](#) ()
- int [getDegreeMillion](#) ()
- int [getMinuteMillion](#) ()
- int [getSecondMillion](#) ()
- int [getDegree](#) ()
- int [getMinute](#) ()
- int [getSecond](#) ()
- int [getMiliSecond](#) ()

#### Metody prywatne

- void [doLoad](#) (int angle, boolean nmea)
- StringBuffer [doformat](#) (int value, int size, int prec, boolean space)

#### 3.1.1 Opis szczegółowy

Obsługa katow - współrzędnych geograficznych.

Klasa przechowuje długość lub szerokość geograficzną (od użytkownika zależy interpretacja która z nich jest w danym obiekcie przechowywana). Dostarcza wygodne metody dostępu do składowych współrzędnych czyli stopni, minut i sekund. Jedną z najważniejszych możliwości jest proste, podobne do `printf()` formatowanie do łańcucha.

#### Autor:

Krzysztof Kosyl

#### 3.1.2 Dokumentacja konstruktora i destruktor

##### 3.1.2.1 `pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle.Angle` (int *angle*)

Konstruktor klasy.

Tworzy obiekt klasy mając dostępne dane w postaci podobnej do tej w standardzie NMEA. Jeśli mamy kat DDD stopni i MM.mmmmm minut to DDDMMmmmm jest reprezentacją dziesiętną argumentu. W zależności od znaku argumentu interpretuje się półkole. Tak więc wartości ujemne przypadają na półkole południowe, zachodnie a dodatnie na północne, wschodnie.

**Parametry:**

*angle* kat w formacie podobnym do NMEA

**3.1.2.2 `pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle.Angle (int angle, boolean nmea)`**

Konstruktor klasy.

Tworzy obiekt klasy mając dostępne dane w postaci liczby całkowitej. Jeśli parametr *nmea* jest równy `true` to kat jest interpretowany podobnie jak w standardzie NMEA. W przeciwnym wypadku *angle* jest traktowane jako zawierające kat wyrażony w stopniach pomnożonych przez milion.

**Parametry:**

*angle* liczba całkowita reprezentująca kat

*nmea* informuje czy dane są w formacie NMEA

**3.1.3 Dokumentacja funkcji składowych****3.1.3.1 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle.fromDegreeMillion (int angle)`**

Laduje nowe dane do obiektu.

Laduje nowe dane do obiektu traktując je jako zawierające kat wyrażony w stopniach pomnożonych przez milion.

**Parametry:**

*angle* liczba całkowita reprezentująca kat

**3.1.3.2 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle.doLoad (int angle, boolean nmea)` [private]**

Metoda dokonująca konwersji danych pomiędzy formatami.

W zależności od parametru *nmea* przekształca dane z formatu NMEA lub z milionów stopni do wewnętrznej reprezentacji klasy.

**Parametry:**

*angle* liczba całkowita reprezentująca kat

*nmea* jeśli `true` to dane są interpretowane jako w formacie NMEA, jeśli `false` to jako miliony stopni

**3.1.3.3 `StringBuffer pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle.doformat (int value, int size, int prec, boolean space)` [private]**

Formatuje liczbę do zadanej postaci

**Parametry:**

*value* liczba do sformatowania

*size* minimalna ilość znaków jaka ma zajmować część przed kropką

*prec* ilość miejsc po kropce, jeśli równa 0 to kropka nie jest dodawana

*space* jeśli `true` to zamiast zer przed kropką są wstawiane spacje

### 3.1.3.4 `String pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle.format (String fmt)`

Metoda przekształcająca kat do postaci tekstowej, podobna do `printf()`.

Zamienia wystąpienia specjalnych specyfikatorów przekształcen na odpowiadające im wartości liczbowe. Specyfikator rozpoczyna się znakiem `"` po nim mogą wystąpić:

- `%` - po prostu wstawia znak `"` w miejscu wystąpienia specyfikatora
- `{[polkolaA],[polkolaB]}` - wstawia jeden z napisów `polkolaA`, `polkolaB` w zależności od tego do jakiej polkuli odwołuje się kat
- `[_][rozmiar][.precyzja]X` - w miejscu `X` może się znajdować jedna z liter `'d'`, `'m'`, `'s'`. Oznaczają one odpowiednio:
  - `'d'` stopnie (degree)
  - `'m'` minuty
  - `'s'` sekundy

Rozmiar to ilość cyfr części całkowitej, jeśli jest zbyt mała aby przestawić wartość to jest pomijana. Jeśli zaistnieje taka potrzeba to nadmiarowe miejsca są uzupełniane zerami.

Jeśli wystąpi znak podkreślenia to nadmiarowe miejsca rozmiaru są uzupełniane spacjami a nie zerami.

Precyzja to ilość miejsc po kropce w reprezentacji liczby. Jeśli jest równa 0 to pomijana jest kropka dziesiętna.

Zarówno rozmiar jak i precyzja muszą być z przedziału `[0, 6]`.

Przykłady (wszystkie dla `Angle(1220.74067)` stopni):

- `"%3d" -> "012"`
- `"%3.4d" -> "012.3456"`
- `"%d %m %s" -> "12 20 44"`
- `"%3d %2m %2s" -> "012 20 44"`
- `"%_3d %2m %2s" -> " 12 20 44"`
- `"%{S,N} %3d %2m %2.4s" -> "N 012 20 44.4402"`
- `"%{S,N} %3d %2m %2.4s == %{poludnie,polnoc} %3.5d" -> "N 012 20 44.4402 == polnoc 012.34567"`

#### Parametry:

*fmt* łańcuch formatujący

#### Zwraca:

łańcuch powstały w wyniku formatowania

### 3.1.3.5 `int pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle.getSign ()`

Określa której polkuli dotyczy kat.

#### Zwraca:

- -1 dla polkoli południowej lub zachodniej
- +1 dla polkoli północnej lub wschodniej

**3.1.3.6 int pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle.getDegreeMillion ()**

Zwraca stopnie pomnozone przez milion

**Zwraca:**

miliony stopni

**3.1.3.7 int pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle.getMinuteMillion ()**

Zwraca minuty pomnozone przez milion

Wynik zawiera sie w przedziale [0, 59999999]

**Zwraca:**

miliony minut

**3.1.3.8 int pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle.getSecondMillion ()**

Zwraca sekundy pomnozone przez milion

Wynik zawiera sie w przedziale [0, 59999999]

**Zwraca:**

miliony sekund

**3.1.3.9 int pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle.getDegree ()**

Zwraca stopnie

**Zwraca:**

stopnie

**3.1.3.10 int pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle.getMinute ()**

Zwraca minuty

Wynik zawiera sie w przedziale [0, 59]

**Zwraca:**

minuty

**3.1.3.11 int pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle.getSecond ()**

Zwraca sekundy

Wynik zawiera sie w przedziale [0, 59]

**Zwraca:**

sekund

### 3.1.3.12 int pl.umk.mat.pz2006.nebuchadrezzar.marduk.Angle.getMiliSecond ()

Zwraca tysięczne części sekundy

Wynik zawiera się w przedziale [0, 999]

#### Zwraca:

miliony sekund

Dokumentacja dla tej klasy została wygenerowana z pliku:

- Angle.java

## 3.2 Dokumentacja klasy pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base

Diagram dziedziczenia dla pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base

### Metody publiczne

- String[] [newPath](#) (ArrayList p, ArrayList spec) throws Exception
- ArrayList [returnPath](#) (int id) throws Exception
- ArrayList [returnPath](#) (int id, int type) throws Exception
- ArrayList [returnPath](#) (double minx, double miny, double maxx, double maxy, String exc) throws Exception
- void [end](#) ()
- ArrayList [getSpecialK](#) (double minx, double miny, double maxx, double maxy) throws Exception
- byte[] [export](#) (double minx, double miny, double maxx, double maxy, String type, String id) throws Exception
- void [delete](#) (int id) throws Exception
- double[] [getRect](#) (int id) throws Exception

### Metody chronione

- double[] [findRect](#) (ArrayList [points](#))
- String [insertPath](#) () throws Exception
- String [parseSpecial](#) () throws Exception
- String [parse](#) (ArrayList as, int pid) throws Exception
- boolean [checkSpecial](#) (double x, double y, int type, String other) throws Exception

### Atrybuty pakietu

- int [what](#) = 19
- int [num](#) = 20
- double [proc](#) = 0.00000001
- double [neight](#) = 0.0035
- ArrayList [points](#) = new ArrayList()
- ArrayList [special](#) = new ArrayList()
- ArrayList [road](#) = new ArrayList()



### 3.2.1 Opis szczegółowy

Klasa odpowiadająca za kontakt z bazą danych, nawiązanie połączenia, wstawianie danych do bazy i ich pobieranie.

**Autor:**

Anna Gogolinska

### 3.2.2 Dokumentacja funkcji składowych

#### 3.2.2.1 `double [] pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.findRect (ArrayList points)` [protected]

Metoda znajdująca najmniejszą i największą wartość długości i szerokości geograficznej wśród punktów, które mają być wstawione do bazy.

**Parametry:**

*points* Lista punktów.

**Zwraca:**

Tablica minimalnych i maksymalnych wartości długości i szerokości geograficznej.

#### 3.2.2.2 `String [] pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.newPath (ArrayList p, ArrayList spec) throws Exception`

Metoda, która kopiuje punkty do wstawienia do bazy do lokalnej listy oraz rozdziela listę punktów w specjalnych na dwie listy. Pierwsza z nich zawiera punkty określające rodzaj drogi, druga obiekty na ścieżce. Następnie wywoływana jest metoda wstawiająca punkty do bazy oraz parsująca punkty specjalne.

**Parametry:**

*p* Lista punktów do wstawienia do bazy, zawierająca tablice trzelementowe w których są szerokość, długość geograficzna i wysokość.

*spec* Lista punktów specjalnych czytanych z pliku składająca się z tablic czteroelementowych zawierających szerokość, długość geograficzną, typ punktu oraz ewentualny opis.

**Zwraca:**

Tablica stringów, w której pierwszy element zawiera id nowych ścieżek, a drugi informacje o punktach specjalnych.

#### 3.2.2.3 `String pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.insertPath () throws Exception` [protected]

Metoda wstawiająca punkty z listy `points` do bazy danych.

**Zwraca:**

String zawierający id nowych ścieżek wstawionych do bazy.

#### 3.2.2.4 `ArrayList pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.returnPath (int id) throws Exception`

Metoda wyszukująca w bazie i zwracająca ścieżkę o podanym identyfikatorze.

##### Parametry:

*id* Identyfikator ścieżki, która ma być zwrócona.

##### Zwraca:

Ścieżka w postaci listy czteroelementowych tablic zmiennych typu `double` zawierających: szerokość geograficzną, długość geograficzną, wysokość i typ punktu.

#### 3.2.2.5 `ArrayList pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.returnPath (int id, int type) throws Exception`

Metoda zwracająca punkty ze ścieżki o podanych identyfikatorze, które są określonego typu.

##### Parametry:

*id* Identyfikator ścieżki.

*type* Typ punktów.

##### Zwraca:

Lista zwracana punkty.

#### 3.2.2.6 `ArrayList pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.returnPath (double minx, double miny, double maxx, double maxy, String exc) throws Exception`

Metoda zwracająca liczącą się ścieżki znajdujące się w danym prostokacie. Zwracane są całe ścieżki, których chociaż 1 punkt znajduje się w określonym obszarze, z wyjątkiem ścieżek o numerach podanych w parametrze `exc`.

##### Parametry:

*minx* Minimalna szerokość geograficzna punktów prostokata.

*miny* Minimalna długość geograficzna punktów prostokata.

*maxx* Maksymalna szerokość geograficzna punktów prostokata.

*maxy* Maksymalna długość geograficzna punktów prostokata.

*exc* String zawierający numery ścieżek, które nie mają być zwracane. Numery oddzielone są spacjami.

##### Zwraca:

Lista Stringów, z których każdy String zawiera jedną ścieżkę.

#### 3.2.2.7 `void pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.end ()`

Zakończenie połączenia z bazą.

**3.2.2.8 String pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.parseSpecial () throws Exception** [protected]

Metoda zmieniajaca liste punktow specjalnych na Stringa.

**Zwraca:**

String w ktorym kazda linia opisuje 1 punkt. Ma ona postac: #szerokosc geograficzna:dlugosc:typ:ewentulany opis

**3.2.2.9 ArrayList pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.getSpecialK (double minx, double miny, double maxx, double maxy) throws Exception**

Metoda wydobywajaca z bazy punkty specjalne, znajdujace sie w wyznaczonym prostokacie.

**Parametry:**

- minx* Minimalna szerokosc geograficzna punktow prostokata.
- miny* Minimalna dlugosc geograficzna punktow prostokata.
- maxx* Maksymalna szerokosc geograficzna punktow prostokata.
- maxy* Maksymalna dlugosc geograficzna punktow prostokata.

**Zwraca:**

Lista punktow z danego obszaru, zawierajaca czteroelementowe tablice obiektow. Kazda tablica opisuje 1 punkt i zawiera szerokosc geograficzna, dlugosc, typ i ewentualnie opis.

**3.2.2.10 String pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.parse (ArrayList as, int pid) throws Exception** [protected]

Metoda zmieniajaca liste punktow sciezki o podanym id na Stringa.

**Parametry:**

- as* Lista punktow.
- pid* Id sciezki.

**Zwraca:**

String zawierajacy punkty sciezki. Kazda linia Stringa zawiera liste punktow o innym typie. String ma postac id\_sciezki:typ\_punktow:kolejne punkty, znak konca linii i kolejne linie w tej samej postaci.

**3.2.2.11 byte [ ] pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.export (double minx, double miny, double maxx, double maxy, String type, String id) throws Exception**

Metoda wydobywajaca z bazy sciezki na okreslonym obszarze lub sciezki o okreslonym id i dokonujaca eksportu tych danych do pliku o okreslonym typie.

**Parametry:**

- minx* Minimalna szerokosc geograficzna punktow prostokata.
- miny* Minimalna dlugosc geograficzna punktow prostokata.

*maxx* Maksymalna szerokosc geograficzna punktow prostokata.

*maxy* Maksymalna dlugosc geograficzna punktow prostokata.

*type* typ eksportowanego pliku

*id* Numer sciezki, ktora ma byc zwrcona.

**Zwraca:**

Wynikowy plik w postaci tablicy bajtow.

**3.2.2.12 boolean `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.checkSpecial (double x, double y, int type, String other)` throws Exception** [protected]

Metoda sprawdzajaca czy w bazie nie znajduje sie juz punkt specjalny o podanych wspolrzecznych oraz typie.

**Parametry:**

*x* Szerokosc geograficzna punktu, ktory ma byc wstawiony do bazy.

*y* Dlugosc geograficzna punktu, ktory ma byc wstawiony do bazy.

*type* Typ wstawianego punktu.

*other* Opis punktu jesli jego typ jest num lub null w przeciwnym wypadku.

**3.2.2.13 void `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.delete (int id)` throws Exception**

Metoda usuwajaca z bazy sciezke o podanym numerze.

**Parametry:**

*id* Numer sciezki do usuniecia.

**3.2.2.14 double [] `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.getRect (int id)` throws Exception**

Metoda zwraca prostokat w ktorym znajduje sie sciezka.

**Parametry:**

*id* Numer sciezki dla ktorej ma byc zwroczony prostokat.

**Zwraca:**

Tablica `double[4]` ktora zawiera w kolejnosci `latA`, `lngA`, `latB`, `lngB`.

**3.2.3 Dokumentacja atrybutów składowych**

**3.2.3.1 int `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.what = 19`** [package]

Zmienna okreslajaca do ktorego numeru punkty specjalne okreslaja rodzaj drogi.

**3.2.3.2** `int pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.num = 20` [package]

Zmienna określająca jaki typ punktu specjalnego określa punkt z opisem użytkownika.

**3.2.3.3** `double pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.proc = 0.00000001`  
[package]

Zmienna określająca jaki procent całego wyswietlanego obszaru musi stanowić obszar zwracanej ścieżki, aby była ona wyswietlana.

**3.2.3.4** `double pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.neight = 0.0035` [package]

Zmienna określająca w jakim promieniu punkty specjalne uznawane są za równe sobie.

**3.2.3.5** `ArrayList pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.points = new ArrayList()`  
[package]

Lista punktów, które mają być wstawione do bazy. Zawiera ona trzelementowe tablice zmiennych typu `double`, zawierające szerokość geograficzną, długość geograficzną, wysokość.

**3.2.3.6** `ArrayList pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.special = new ArrayList()`  
[package]

Lista punktów specjalnych, zawiera ona w kolejności szerokość geograficzną (`double`), długość (`double`), typ punktu oraz jeśli typ jest równy (`int`) `num` to również opis punktu (`String`).

**3.2.3.7** `ArrayList pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Base.road = new ArrayList()`  
[package]

Lista punktów specjalnych określających rodzaj drogi. Zawiera ona trzelementowe tablice zmiennych typu `double`, zawierające szerokość geograficzną, długość oraz typ drogi.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `Base.java`

### 3.3 Dokumentacja klasy `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread`

Diagram dziedziczenia dla `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThreadDiagram` współpracy dla `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread`:

#### Metody publiczne

- `BaseThread` (`Socket s1`, `HashMap hm`)
- `void run ()`
- `void position (Angle latitude, Angle longitude)`
- `void altitude (int altitude)`
- `void time (int time)`
- `void date (int date)`
- `void satellite (int number, int elevation, int azimuth, int strength)`
- `void accuracy (int satellites, int dilution)`

### Atrybuty pakietu

- `int what = 19`
- `int num = 20`
- `double alt = -1000`
- `ArrayList points = new ArrayList()`
- `ArrayList special = new ArrayList()`
- `String error = " "`

### Metody prywatne

- `String[] readPoints (BufferedReader br) throws IOException`

#### 3.3.1 Opis szczegółowy

##### Autor:

Ania

#### 3.3.2 Dokumentacja konstruktora i destruktora

##### 3.3.2.1 `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread.BaseThread (Socket sI, HashMap hm)`

Stworzenie nowego obiektu typu `BaseThread`. Zczytanie parametrów takich jak rodzaj bazy danych, nazwa bazy, użytkownika i hasło, adres IP bazy oraz parametry `num` oraz `what`. Następnie wątek tworzy obiekt bazy danych w zależności od rodzaju.

##### Parametry:

*sI* Gniazdo utworzone dla klienta, którego obsługuje wątek.

*hm* Mapa hashowana zawierająca parametry konfiguracyjne.

#### 3.3.3 Dokumentacja funkcji składowych

##### 3.3.3.1 `void pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread.run ()`

Metoda startująca nowy wątek. Wątek czyta z gniazda i w zależności od tego co przeczyta wywołuje odpowiednią funkcję bazy danych.

##### 3.3.3.2 `String [] pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread.readPoints (BufferedReader br) throws IOException [private]`

Metoda czytająca punkty z gniazda, dokonująca parsowania tych punktów i wywołująca odpowiednią funkcję z bazy danych, która wstawia je do bazy.

##### Parametry:

*br* Strumień podpięty do gniazda z którego czyta i pisze wątek oraz z którego czyta klient.

##### Zwraca:

Tablica Stringów, w których pierwszy element to wymienione ID ścieżek, które zostały wstawione do bazy. Drugi to lista punktów specjalnych.

**3.3.3.3 void `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread.position` ([Angle latitude](#), [Angle longitude](#))**

Metoda parsera NMEA, wywoływana jest ona jeśli parser odczyta szerokość i długość geograficzną. Zczytane wartości wstawiane są do listy punktów.

**Parametry:**

*latitude* Szerokość geograficzna,

*longitude* Długość geograficzna.

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.GisInfo](#).

**3.3.3.4 void `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread.altitude` ([int altitude](#))**

Metoda parsera NMEA, wywoływana jest ona jeśli parser odczyta wysokość punktu. Zmieniana jest aktualna wysokość punktu.

**Parametry:**

*altitude* Wysokość nad poziomem morza.

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.GisInfo](#).

**3.3.3.5 void `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread.time` ([int time](#))**

Wywołanie dla czasu.

Czas jest wyrażony dla strefy UTC (Coordinated Universal Time).

**Parametry:**

*time* ilość milisekund od północy

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.GisInfo](#).

**3.3.3.6 void `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread.date` ([int date](#))**

Wywołanie dla daty.

Data jest podana względem UTC (Coordinated Universal Time).

**Parametry:**

*date* aktualna data (jeśli d - dzień miesiąca; m - numer miesiąca; r - rok to parametr przyjmuje wartość  $(d * 10000 + m * 100 + y \% 100)$ )

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.GisInfo](#).

**3.3.3.7 void `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread.satellite` ([int number](#), [int elevation](#), [int azimuth](#), [int strength](#))**

Wywołanie dla informacji o pozycjach na niebie widocznych satelitów.

Jest wywoływana tyle razy ile satelitów znajduje się w zasięgu.

**Parametry:**

*number* numer identyfikacyjny satelity

*elevation* wysokość nad horyzontem w stopniach, wskazująca pozycję satelity, wartość z przedziału [0,90]

*azimuth* azymut wskazujący pozycję satelity, wartość z przedziału [0,359]

*strength* siła sygnału z przedziału [0,100], wartości są mocno zależne od producenta odbiornika, im większa tym lepsza, jeśli jest równa zero satelita nie brał udziału w wyznaczaniu pozycji

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.GisInfo](#).

**3.3.3.8 void `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread.accuracy` (int *satellites*, int *dilution*)**

Wywołanie dla informacji o spodziewanej dokładności odczytu.

**Parametry:**

*satellites* ilość satelitów z których dane brały udział w wyznaczaniu aktualnej pozycji, im więcej tym lepiej, do wyznaczenia pozycji wymagane są minimum 3

*dilution* wyliczony spodziewany błąd w wyliczeniu współrzędnych geograficznych, wyrażony w deymetrach, nie należy zbyt sugerować się jego wartością

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.GisInfo](#).

**3.3.4 Dokumentacja atrybutów składowych****3.3.4.1 int `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread.what` = 19** [package]

Zmienna określająca do którego numeru punkty specjalne określają rodzaj drogi.

**3.3.4.2 int `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread.num` = 20** [package]

Zmienna określająca jaki typ punktu specjalnego określa punkt z opisem użytkownika.

**3.3.4.3 double `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread.alt` = -1000** [package]

Domyslna wysokość na której leży punkt, wartość oznaczająca, że wysokość nie jest określona.

**3.3.4.4 ArrayList `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread.points` = new ArrayList()** [package]

Lista punktów, które mają być wstawione do bazy. Zawiera ona trzelementowe tablice zmiennych typu double, zawierające szerokość geograficzną, długość geograficzną, wysokość.

**3.3.4.5 ArrayList `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread.special` = new ArrayList()** [package]

Lista punktów specjalnych, zawiera ona w kolejności szerokość geograficzną (double), długość geograficzną (double), typ punktu oraz jeśli jest równy (int) num to również opis punktu (String).



**3.3.4.6 String** `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread.error` = " "  
[package]

Ciag znakow, ktory bedzie wpisywany do gniazda jesli wystawi jakis blad.

Dokumentacja dla tej klasy zostala wygenerowana z pliku:

- `BaseThread.java`

### 3.4 Dokumentacja klasy `pl.umk.mat.pz2006.nebuchadrezzar.marduk.Exporter`

Diagram dziedziczenia dla `pl.umk.mat.pz2006.nebuchadrezzar.marduk.Exporter`

#### Metody publiczne

- `abstract byte[] export` (`ArrayList pts`, `ArrayList sp`, `double[] brd`)

#### Metody chronione

- `double[] findBorders` ()
- `int[] pavingColor` (`int n`)

#### Atrybuty chronione

- `ArrayList pathes`  
*sciezki wystepujace w podanym zakresie kuli ziemskiej*
- `ArrayList special`  
*punkty specjalne wystepujace na podanym obszarze*
- `HashMap map`  
*mapa haszowana przypisujaca numerom nazwy punktow specjalnych (na podstawie tabeli w bazie danych)*
- `double[] borders`  
*wspolrzedne ograniczajace obszar do zrzutowania na mape*

#### 3.4.1 Opis szczegolowy

Klasa eksportujaca podana liste sciezek i punktow specjalnych z danego obszaru do formatu zewnetrznego (PDF, postscript)

#### Autor:

Agnieszka Holka

### 3.4.2 Dokumentacja funkcji składowych

#### 3.4.2.1 `abstract byte [] pl.umk.mat.pz2006.nebuchadrezzar.marduk.Exporter.export (ArrayList pts, ArrayList sp, double[] brd)` [pure virtual]

Główna metoda klasy, odpowiadająca za cały przebieg procesu eksportowania danych do pliku. Tworzy plik dokumentu wyników, zapisuje do niego podstawowe informacje i wywołuje wyspecyfikowane metody, odpowiedzialne za nanoszenie szciezek, punktów specjalnych, siatki oraz legendy.

#### Parametry:

*pts* lista szciezek w postaci list punktów

*sp* lista punktów specjalnych

*brd* granice tworzonej mapy

#### Zwraca:

tablica bajtów zawierająca treść wynikowego pliku

Implementowany w [pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXExporter](#), [pl.umk.mat.pz2006.nebuchadrezzar.marduk.KMLExporter](#), [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter](#), [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSEExporter](#) i [pl.umk.mat.pz2006.nebuchadrezzar.marduk.TXTExporter](#).

#### 3.4.2.2 `double [] pl.umk.mat.pz2006.nebuchadrezzar.marduk.Exporter.findBorders` ( ) [protected]

Metoda określa prostokąt ograniczający podany zbiór szciezek, aby na jego podstawie określić skalę mapy. Wywoływana jest jedynie, jeśli zakres mapy nie został podany jako argument do metody `export`

#### 3.4.2.3 `int [] pl.umk.mat.pz2006.nebuchadrezzar.marduk.Exporter.pavingColor` (int n) [protected]

Przydziela kolory dla oznaczenia różnych nawierzchni trasy

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `Exporter.java`

## 3.5 Dokumentacja interfejsu `pl.umk.mat.pz2006.nebuchadrezzar.marduk.GisInfo`

Diagram dziedziczenia dla `pl.umk.mat.pz2006.nebuchadrezzar.marduk.GisInfo`

### Metody publiczne

- void `position` (`Angle` latitude, `Angle` longitude)
- void `altitude` (int altitude)
- void `time` (int time)
- void `date` (int date)
- void `satellite` (int number, int elevation, int azimuth, int strength)
- void `accuracy` (int satellites, int dilution)

### 3.5.1 Opis szczegółowy

Interface wywołan zwrotnych.

Program chcący korzystać z `NmeaParser` musi posiadać klasę implementującą ten interfejs. Jest ona przekazywana parserowi podczas jego tworzenia. W zależności od napływających danych parser wywołuje odpowiednie metody z interfejsu.

Realizuje abstrakcję źródła danych. Nie zwraca informacji o odczytanym poleceniu NMEA, lecz o danych, jakie ona zawierała. Ułatwia to obsługę różnorodnych odbiorników, gdyż nie wszystkie muszą obsługiwać dane polecenia.

#### Autor:

Krzysztof Kosyl

### 3.5.2 Dokumentacja funkcji składowych

#### 3.5.2.1 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.GisInfo.position (Angle latitude, Angle longitude)`

Wywołanie dla współrzędnych geograficznych.

#### Parametry:

*latitude* szerokość geograficzna

*longitude* długość geograficzna

Implementowany w `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread`.

#### 3.5.2.2 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.GisInfo.altitude (int altitude)`

Wywołanie dla wysokości nad poziomem morza.

Odbiorniki GPS nie są zbyt dokładne w wyznaczaniu wysokości nad poziomem morza, potrafią się mylić nawet o dziesiątki metrów. Dlatego nie należy zbyt sugerować się tymi danymi.

#### Parametry:

*altitude* wysokości nad poziomem morza wyrażona w metrach

Implementowany w `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread`.

#### 3.5.2.3 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.GisInfo.time (int time)`

Wywołanie dla czasu.

Czas jest wyrażony dla strefy UTC (Coordinated Universal Time).

#### Parametry:

*time* ilość milisekund od północy

Implementowany w `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread`.

**3.5.2.4 void `pl.umk.mat.pz2006.nebuchadrezzar.marduk.GisInfo.date` (int *date*)**

Wywołanie dla daty.

Data jest podana względem UTC (Coordinated Universal Time).

**Parametry:**

*date* aktualna data (jesli *d* - dzien miesiaca; *m* - numer miesiaca; *r* -rok to paramemetr przyjmuje wartosc ( $d * 10000 + m * 100 + y \% 100$ ))

Implementowany w [pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread](#).

**3.5.2.5 void `pl.umk.mat.pz2006.nebuchadrezzar.marduk.GisInfo.satellite` (int *number*, int *elevation*, int *azimuth*, int *strength*)**

Wywołanie dla informacji o pozycjach na niebie widocznych satelitow.

Jest wywoływana tyle razy ile satelitow znajduje sie w zasiegu.

**Parametry:**

*number* numer identyfikacyjny satelity

*elevation* wysokosc nad horyzontem w stopniach, wskazujaca pozucje satelity, wartosc z przedzialu [0,90]

*azimuth* azymut wskazujacy pozucje satelity, wartosc z przedzialu [0,359]

*strength* sila sygnalu z przedzialu [0,100], wartosci sa mocno zalezne od producenta odbiornika, im wieksza tym lepsza, jesli jest rowna zero satelita nie bral udzialu w wyznaczaniu pozycji

Implementowany w [pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread](#).

**3.5.2.6 void `pl.umk.mat.pz2006.nebuchadrezzar.marduk.GisInfo.accuracy` (int *satellites*, int *dilution*)**

Wywołanie dla informacji o spodiewanej dokladnosci odczytu.

**Parametry:**

*satellites* ilosc satelitow z ktorych dane braly udzial w wyznaczaniu aktualnej pozycji, im wiecej tym lepiej, do wyznaczenia pozycji wymagane sa minimum 3

*dilution* wyliczony spodziewany blad w wyliczeniu wspolrzednych geograficznych, wyrazony w decymetrach, nie nalezy zbytnio sugerowac sie jego wartoscia

Implementowany w [pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.BaseThread](#).

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- `GisInfo.java`

**3.6 Dokumentacja interfejsu `pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXBase`****Atrybuty publiczne**

- String `pathMarker` = "trk"

*znacznik trasy w formacie GPX*

- String `segMarker` = "trkseg"  
*znacznik fragmentu trasy*
- String `pointMarker` = "trkpt"  
*znacznik punktu nalezacego do trasy*
- String `path2Marker` = "rte"  
*znacznik trasy w formacie GPX*
- String `point2Marker` = "rtept"  
*znacznik punktu nalezacego do trasy*
- String `latMarker` = "lat"  
*znacznik szerokosci geograficznej*
- String `lngMarker` = "lon"  
*znacznik wysokosci geograficznej*
- String `altMarker` = "ele"  
*znacznik wysokosci n.p.m.*
- String `specialMarker` = "wpt"  
*znacznik okreslajacy punkt specjalny*
- String `nameMarker` = "name"  
*znacznik nazwy punktu*
- String `descMarker` = "desc"  
*znacznik komentarza do punktu*

### 3.6.1 Opis szczegółowy

Interfejs zawierający definicje potrzebnych znaczników GPX

**Autor:**

Agnieszka Holka

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- GPXBase.java

## 3.7 Dokumentacja klasy `pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXExporter`

Diagram dziedziczenia dla `pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXExporter`Diagram współpracy dla `pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXExporter`:

### Metody publiczne

- `byte[] export` (`ArrayList pts`, `ArrayList sp`, `double[] brd`)

### Metody chronione

- `void writeMetadata` ()
- `void writePathes` ()
- `void writeSpecial` ()

#### 3.7.1 Opis szczegółowy

Klasa sluzaca do eksportowania tras do formatu GPX (GPS Exchange System). Jest to forma XML opracowana w celu wymiany danych GPS (punkty na trasie, sciezki itp.) pomiedzy aplikacjami i serwisami webowymi. Format GPX jest obslugiwany przez takie programy jak GPSMan i EasyGPS, jak rowniez wiele stron internetowych

#### Autor:

Agnieszka Holka

#### 3.7.2 Dokumentacja funkcji składowych

##### 3.7.2.1 `byte [] pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXExporter.export` (`ArrayList pts`, `ArrayList sp`, `double[] brd`) [`virtual`]

Główna metoda klasy, odpowiadająca za cały przebieg procesu eksportowania danych do pliku. Tworzy plik dokument wynikowy, zapisuje do niego podstawowe informacje i wywołuje wyspecyfikowane metody, odpowiedzialne za nanoszenie sciezek i punktow specjalnych.

#### Parametry:

*pts* lista sciezek w postaci list punktow

*sp* lista punktow specjalnych

#### Zwraca:

tablica bajtow zawierajaca tresc wynikowego pliku

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.Exporter](#).

##### 3.7.2.2 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXExporter.writeMetadata` () [`protected`, `virtual`]

Metoda wprowadza do XML podstawowe dane dotyczace calego pliku (zakres szerokosci i dlugosci geograficznej)

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.XMLExporter](#).

##### 3.7.2.3 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXExporter.writePathes` () [`protected`, `virtual`]

Metoda przechodzi wszystkie ścieżki, a w ich obrębie kolejne punkty. Dla każdego elementu wywoływane są metody `startElement` oraz `endElement`, które powiadamiają o zmianach w XML i pozwalają na ich serializację.

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.XMLExporter](#).

**3.7.2.4 void `pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXExporter.writeSpecial` ()**  
[protected, virtual]

Eksport punktów specjalnych

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.XMLExporter](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `GPXExporter.java`

## 3.8 Dokumentacja klasy `pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXParser`

### Metody publiczne

- void `parse` (String `pathname`)
- void `startElement` (String `uri`, String `localName`, String `qName`, Attributes `atts`) throws `SAXException`
- void `endElement` (String `uri`, String `localName`, String `qName`) throws `SAXException`
- void `characters` (char `buf` [], int `offset`, int `len`) throws `SAXException`
- ArrayList `getPoints` ()

### Atrybuty prywatne

- SAXParser `saxParser`  
*parser XML*
- ArrayList `pathes` = null  
*lista ścieżek odczytanych z pliku*
- ArrayList `points` = null  
*lista punktów aktualnie parsowanej ścieżki*
- String `pathname` = null  
*ścieżka do parsowanego pliku*
- double `alt`  
*współrzędne geograficzne bieżącego punktu*
- boolean `isAlt` = false  
*czy punkt ma określoną wysokość n.p.m.*
- int `mult` = 1000000  
*mnożnik wykorzystywany podczas zapisu do pliku w formacie wykorzystywanym przez aplikację Nabuchodonozor (liczby w pliku tworzonym na telefonie komórkowym muszą być całkowite)*

### 3.8.1 Opis szczegółowy

Klasa parsująca pliki GPX (GPS Exchange Format) do formatu wymaganego przez aplikacje

**Autor:**

Agnieszka Holka

### 3.8.2 Dokumentacja funkcji składowych

#### 3.8.2.1 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXParser.parse (String pathname)`

Główna funkcja klasy, która pobiera plik w formacie GPX i wywołuje jego parsowanie przez obiekt klasy `SAXParser`

**Parametry:**

*pathname* nazwa pliku XML

#### 3.8.2.2 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXParser.startElement (String uri, String localName, String qName, Attributes atts) throws SAXException`

Nadpisana metoda klasy `DefaultHandler`; otrzymuje powiadomienia o rozpoczynających się elementach XML-a. Dokładniej, wywoływana jest za każdym razem, gdy `saxParser` natrafi na rozpoczynający się element w analizowanym pliku.

**Parametry:**

*uri* parametr pomijany

*localName* lokalna nazwa elementu

*qName* nazwa wykorzystywana, jeśli *localName* jest pustym ciągiem znaków

*atts* atrybuty rozpoczynającego się elementu (wykorzystywane są szerokość i długość geograficzna)

#### 3.8.2.3 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXParser.endElement (String uri, String localName, String qName) throws SAXException`

Metoda reaguje na powiadomienia o kończących się elementach. Faktycznie obsługuje jedynie znacznik zakończenia ścieżki; bieżąca trasa wstawia do listy tras

**Parametry:**

*uri* parametr pomijany

*localName* lokalna nazwa elementu

*qName* nazwa wykorzystywana, jeśli *localName* jest pustym ciągiem znaków

#### 3.8.2.4 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXParser.characters (char buf[], int offset, int len) throws SAXException`

Metoda obsługuje odczytane przez parser dane znakowe; jeśli obiekt ma zaznaczoną flagę `isAlt` (czy odczytano właśnie znacznik wysokości punktu), to odczytane znaki podają wysokość ostatniego punktu; punkt ten jest usuwany z listy i wstawiany ponownie, z uzupełnioną wysokością n.p.m.



**Parametry:**

- buf* dane znakowe dokumentu XML
- offset* numer pierwszego znaku w tablicy
- len* dlugosc odczytanego lanucha znakow

**3.8.2.5 ArrayList pl.umk.mat.pz2006.nebuchadrezzar.marduk.GPXParser.getPoints ()**

Funkcja zwraca punkty sciezki (szerokosc, dlugosc geograficzna i wysokosc nad poziomem morza, o ile zostala podana) w formie, ktora przy uzyciu dostepnych funkcji mozna wprowadzic do bazy danych

**Zwraca:**

lista punktow

Dokumentacja dla tej klasy zostala wygenerowana z pliku:

- GPXParser.java

**3.9 Dokumentacja klasy pl.umk.mat.pz2006.nebuchadrezzar.marduk.KMLExporter**

Diagram dziedziczenia dla pl.umk.mat.pz2006.nebuchadrezzar.marduk.KMLExporterDiagram wspolpracy dla pl.umk.mat.pz2006.nebuchadrezzar.marduk.KMLExporter:

**Metody publiczne**

- byte[] [export](#) (ArrayList pts, ArrayList sp, double[] brd)

**Metody chronione**

- void [writeMetadata](#) ()
- void [writePathes](#) ()
- void [writeSpecial](#) ()

**Metody prywatne**

- void [startPath](#) (int col)
- void [endPath](#) ()

**Atrybuty prywatne**

- TransformerHandler [hd](#) = null  
*obiekt odpowiedzialny za zapis do pliku XML*
- AttributesImpl [atts](#) = new AttributesImpl()  
*atrybuty znacznika w XML*

### 3.9.1 Opis szczegółowy

Klasa sluzaca do eksportowania tras do KML. Pliki w tym formacie mozna wyswietlac w Google Earth, Google Maps czy Google Maps for Mobile. Eksport obejmuje zarys tras z okreslonego zakresu, znajdujace sie na nim punkty markowane. Kolory szciezek odpowiadaja oznaczeniom rodzaju nawierzchni stosowanym w aplikacji.

**Autor:**

Agnieszka Holka

### 3.9.2 Dokumentacja funkcji składowych

#### 3.9.2.1 `byte [] pl.umk.mat.pz2006.nebuchadrezzar.marduk.KMLExporter.export (ArrayList pts, ArrayList sp, double[] brd) [virtual]`

Główna metoda klasy, która tworzy dokument XML i wywołuje metody, tworzące w pliku strukturę informacji o szkiechach, punktach markowanych i stylach wyświetlania (kolory linii, ładowanie obrazków)

**Parametry:**

*pts* lista szciezek w postaci list punktów

*sp* lista punktów specjalnych

**Zwraca:**

tablica bajtów zawierająca treść wynikowego pliku

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.Exporter](#).

#### 3.9.2.2 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.KMLExporter.writeMetadata () [protected, virtual]`

Metoda definiuje kolory linii dla poszczególnych rodzajów nawierzchni używanych w aplikacji oraz ikony (użyte linki do plików na serwerze) do zaznaczania sklepów, pol namiotowych itp.

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.XMLExporter](#).

#### 3.9.2.3 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.KMLExporter.writePathes () [protected, virtual]`

Metoda przechodzi wszystkie szkiezki, a w ich obrebie kolejne punkty. Dla każdego elementu wywoływane są metody `startElement` oraz `endElement`, które powiadamiają o zmianach w XML i pozwalają na ich serializację. Szkiezki są dodatkowo dzielone ze względu na zmiany rodzajów nawierzchni, wysokość punktów jest ustalona na 10 metrów nad ziemią, aby można było oglądać zaznaczenie trasy w trójwymiarze.

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.XMLExporter](#).

#### 3.9.2.4 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.KMLExporter.startPath (int col) [private]`

Metoda wprowadza do pliku XML informacje o rozpoczęciu nowej szkiezki i wizualne ustawienia dla niej: kolor w zależności od rodzaju nawierzchni, grubość linii, przeciąganie punktów do ziemi, aby szkiezka wyglądała jak 'mur', wysokość rysowanych punktów w odniesieniu do lokalnej wysokości n.p.m.

### 3.9.2.5 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.KMLExporter.endPath ()` [private]

Metoda zaznacza w dokumencie koniec sciezki

### 3.9.2.6 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.KMLExporter.writeSpecial ()` [protected, virtual]

Eksport punktow specjalnych: dla standardowo przewidzianych przez program dodawane sa ikony takie jak te wystepujace na Semiramidzie. Punkty oznaczane opisowo zyskuja domyslony znak graficzny.

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.XMLExporter](#).

Dokumentacja dla tej klasy zostala wygenerowana z pliku:

- `KMLExporter.java`

## 3.10 Dokumentacja klasy `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Main`

### Statyczne metody publiczne

- static void `main` (`String[] args`)

### 3.10.1 Opis szczegolowy

Klasa uruchamiajaca modul bazodanowy

#### Autor:

Anna Gogolinska

### 3.10.2 Dokumentacja funkcji skladowych

#### 3.10.2.1 `static void pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.Main.main (String[] args)` [static]

Metoda czyta z pliku konfiguracyjnego, otwiera `Socket` i oczekuje na klientow. Jesli zaakceptuje polaczenie, tworzy nowy watek i przekazuje mu parametry konfiguracyjne. tworzy watek do jego obsluzenia.

#### Parametry:

*args* nieuzywany

Dokumentacja dla tej klasy zostala wygenerowana z pliku:

- `Main.java`

## 3.11 Dokumentacja klasy `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.MySQL-Base`

Diagram dziedziczenia dla `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.MySQLBase`Diagram wspolpracy dla `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.MySQLBase`:

### Metody publiczne

- [MySQLBase](#) (String ip, int num1, int what1, String name, String user, String pass)

#### 3.11.1 Opis szczegółowy

##### Autor:

Ania Gogolinska

#### 3.11.2 Dokumentacja konstruktora i destruktora

##### 3.11.2.1 `pl.umk.mat.pz2006.nebuchadrezzar.hamurabi.MySQLBase.MySQLBase` (String ip, int num1, int what1, String name, String user, String pass)

Stworzenie obiektu odpowiedzialnego za komunikacje z MySQLowa baza danych. Nawiązanie polaczenia z baza.

##### Parametry:

*ip* Adres ip komutera na ktory znajduje sie baza.

*num1* Numer oznaczajacy rodzaj punktu z dodatkowym opisem.

*what1* Numer oznaczajacy ostatni rodzaj punktu, okreslajacy rodzaj drogi.

*name* Nazwa bazy danych.

*user* Nazwa uzytkownika bazy danych.

*pass* Haslo uzytkownia bazy danych.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `MySQLBase.java`

## 3.12 Dokumentacja klasy `pl.umk.mat.pz2006.nebuchadrezzar.marduk.NmeaParser`

Diagram współpracy dla `pl.umk.mat.pz2006.nebuchadrezzar.marduk.NmeaParser`:

### Metody publiczne

- [NmeaParser](#) ([GisInfo](#) gisInfo)
- void [setFeature](#) (String name, boolean value)
- void [reset](#) ()
- void [append](#) (String what)
- void [append](#) (char[] what, int length)

### Metody chronione

- Integer [parseFixed](#) (String fixed, int precision)

### Metody prywatne

- void `parseLine` (String line)
- Integer `parseInteger` (String integer, int minSize)

### Statyczne metody prywatne

- static String[] `split` (String str, char ch)

#### 3.12.1 Opis szczegółowy

Parser NMEA 0183 oparty na zdarzeniach.

Program chcący korzystać z `NmeaParser` musi posiadać klasę implementującą interfejs `GisInfo` - jest ona przekazywana parserowi podczas jego tworzenia. W zależności od napływających danych parser wywołuje odpowiednie metody z interfejsu.

Parser jest wzorowany na standardzie SAX.

Realizuje abstrakcję źródła danych. Nie zwraca informacji o odczytanym poleceniu NMEA lecz o danych jakie ona zawierała. Ułatwia to obsługę różnorodnych odbiorników, gdyż nie wszystkie muszą obsługiwać dane polecenia.

Obsługuje najpopularniejsze, występujące niemal we wszystkich odbiornikach polecenia NMEA:

- GGA - fix information
- GLL - lat/lon data
- GSA - overall satellite data
- GSV - detailed satellite data
- RMC - recommended minimum data for gps

Jest odporny na uszkodzone dane, sprawdza sumy kontrolne, może zostać podłączony do istniejącej już transmisji. Jeśli natrafi na jakiegokolwiek błąd po cichu je omija, nie próbując odzyskiwać.

#### Autor:

Krzysztof Kosyl

#### 3.12.2 Dokumentacja konstruktora i destruktor

##### 3.12.2.1 `pl.umk.mat.pz2006.nebuchadrezzar.marduk.NmeaParser.NmeaParser` (`GisInfo` `gisInfo`)

Tworzy nowy parser.

#### Parametry:

`gisInfo` obiekt klasy implementującej interfejs `GisInfo` do której będą przekazywane wyniki parsowania

### 3.12.3 Dokumentacja funkcji składowych

#### 3.12.3.1 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.NmeaParser.setFeature (String name, boolean value)`

Ustawia dodatkowe własności parsera.

W chwili obecnej obsługuje tylko jedna możliwość: "allowSimpleNewLine" która jeśli jest ustawiona na true powoduje mniej restrykcyjne parsowanie znaków nowej linii (tzn nie traktuje jako błędnych linii zakończonych "\r" lub "\n"; standard wymaga dwuznakowego "\r\n").

##### Parametry:

*name* nazwa własności

*value* jeśli true aktywuje własność, w przeciwnym wypadku dezaktywuje

#### 3.12.3.2 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.NmeaParser.reset ()`

Resetuje parser.

Przywraca parser do stanu pierwotnego, czyli takiego w jakim znajdował się zaraz po stworzeniu. Jest to o wiele szybsze rozwiązanie od tworzenia nowego obiektu.

#### 3.12.3.3 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.NmeaParser.append (String what)`

Przesyła do parsera porcję danych.

Do parsera można wysyłać porcję danych dowolnej wielkości, zarówno pojedyncze znaki jak i wielolinio-  
we ciągi. Porcja danych nie musi się zaczynać w początku linii - parser po prostu pominię niepoprawny  
fragment linii.

##### Parametry:

*what* porcja danych

#### 3.12.3.4 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.NmeaParser.append (char[] what, int length)`

Przesyła do parsera porcję danych.

Do parsera można wysyłać porcję danych dowolnej wielkości, zarówno pojedyncze znaki jak i wielolinio-  
we ciągi. Porcja danych nie musi się zaczynać w początku linii - parser po prostu pominię niepoprawny  
fragment linii.

Ta wersja metody obsługuje dane wejściowe w postaci tablicy znaków, aby ułatwić korzystanie z niej nie  
wymaga aby porcja danych zajmowała całą tablicę.

##### Parametry:

*what* porcja danych

*length* ile początkowych znaków wysłać do parsera.

**3.12.3.5 void `pl.umk.mat.pz2006.nebuchadrezzar.marduk.NmeaParser.parseLine` (String *line*)**  
[private]

Parse single line of input.

**Parametry:**

*line* Line of text beginning with "\$", without "\n", "\r\n" on end. Maximal length is 78.

**3.12.3.6 Integer `pl.umk.mat.pz2006.nebuchadrezzar.marduk.NmeaParser.parseInt` (String *integer*, int *minSize*)**  
[private]

Przetwarza napis do postaci liczby całkowitej.

**3.12.3.7 Integer `pl.umk.mat.pz2006.nebuchadrezzar.marduk.NmeaParser.parseFixed` (String *fixed*, int *precision*)**  
[protected]

Przetwarza napis do postaci liczby stałoprzecinkowej zapisanej w Integer. Przykłady dla `precision = 3`

- "" -> 0
- "." -> 0
- "1" -> 1000
- "1." -> 1000
- "123456" -> 123456000
- ".5" -> 500
- "1.5" -> 1500
- "1.543" -> 1543
- "1.54321" -> 1543
- "1.23456ala" -> null

**Parametry:**

*fixed* napis zawierający liczbę zmiennoprzecinkową postaci "[0-9]\*([0-9]\*.?)?"

*precision* ilość cyfr po przecinku jakie mają być uwzględniane

**Zwraca:**

Float(`fixed`) \* 10 ^ `precision` jeśli `fixed` było w podanym formacie, null w przeciwnym wypadku

**3.12.3.8 static String [] `pl.umk.mat.pz2006.nebuchadrezzar.marduk.NmeaParser.split` (String *str*, char *ch*)**  
[static, private]

Własna prosta wersja `split()`.

Własna wersja `split()` znacząco uproszczona względem standardowej. Potrafi dzielić łańcuch na tablicę łańcuchów stosując jako separator podany znak (a NIE wyrażenie regularne). Została stworzona ponieważ klasa String w JavaME nie zawiera odpowiedniej metody.

**Parametry:**

- str* lancuch zrodlowy
- ch* znak bedacy separatorem

**Zwraca:**

tablica lancuchow

Dokumentacja dla tej klasy została wygenerowana z pliku:

- NmeaParser.java

### 3.13 Dokumentacja klasy pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter

Diagram dziedziczenia dla pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporterDiagram współpracy dla pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter:

**Metody publiczne**

- byte[] [export](#) (ArrayList pts, ArrayList sp, double[] brd)

**Metody chronione**

- void [paintLatLng](#) ()
- void [paintPathes](#) ()
- void [markSpecialPoints](#) ()
- void [paintTitle](#) ()
- void [createLegend](#) ()

**Metody prywatne**

- void [clipMap](#) ()
- PdfPCell [createCell](#) (String s, Font font)
- void [showText](#) (String text, float x, float y)
- byte[] [getImageByte](#) (String name)

**Atrybuty prywatne**

- Document [doc](#)  
*obiekt reprezentujacy dokument w formacie PDF*
- PdfWriter [writer](#)  
*obiekt odpowiedzialny za zapis wygenerowanych elementow do pliku PDF*
- PdfContentByte [cb](#)  
*zapisuje dane bajtowo (wykorzystywane glownie do rysowania linii)*



### 3.13.1 Opis szczegółowy

Klasa eksportująca podana listę ścieżek i punktów specjalnych z danego obszaru do pliku w formacie PDF

**Autor:**

Agnieszka Holka

### 3.13.2 Dokumentacja funkcji składowych

#### 3.13.2.1 `byte [] pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter.export (ArrayList pts, ArrayList sp, double[] brd)` [virtual]

Główna metoda klasy, odpowiadająca za cały przebieg procesu eksportowania danych do pliku. Tworzy plik dokumentu wyników, zapisuje do niego podstawowe informacje i wywołuje wyspecyfikowane metody, odpowiedzialne za nanoszenie ścieżek, punktów specjalnych, siatki oraz legendy.

**Parametry:**

*pts* lista ścieżek w postaci list punktów

*sp* lista punktów specjalnych

*brd* granice tworzonej mapy

**Zwraca:**

tablica bajtów zawierająca treść wynikowego pliku

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.Exporter](#).

#### 3.13.2.2 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter.paintLatLng` () [protected, virtual]

Metoda nanosząca na mapę siatkę południków i równoleżników wraz z wartościami szerokości i długości geograficznej

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter](#).

#### 3.13.2.3 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter.paintPathes` () [protected, virtual]

Metoda, która rysuje na mapie ścieżki podane w postaci listy punktów

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter](#).

#### 3.13.2.4 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter.markSpecialPoints` () [protected, virtual]

Zaznacza na mapie wszystkie punkty specjalne i dopisuje im etykiety

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter](#).

#### 3.13.2.5 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter.paintTitle` () [protected, virtual]

Tytuł i logo projektu jako nagłówek strony

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter](#).

**3.13.2.6** `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter.createLegend ()`  
[protected, virtual]

Tworzy legende do mapy i wyswietla ja z boku strony

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter](#).

**3.13.2.7** `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter.clipMap ()` [private]

Metoda ogranicza drukowana mape, aby sciezki nie wychodzily na tytul czy legende

**3.13.2.8** `PdfPCell pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter.createCell (String s, Font font)` [private]

Tworzenie nowej komorki tabeli (wykorzystywane w legendzie)

**3.13.2.9** `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter.showText (String text, float x, float y)` [private]

Wyswietla tekst w okreslonych wspolrzednych, jednoczesnie zachpuwujac obramowanie tekstu (w celu przejrzystosci wydruku)

**3.13.2.10** `byte [] pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter.getImageByte (String name)` [private]

Metoda czyta obraz i zapisuje go do tablicy bajtow

Dokumentacja dla tej klasy została wygenerowana z pliku:

- PDFExporter.java

## 3.14 Dokumentacja klasy `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineMerger`

### Metody publiczne

- boolean [merge](#) (ArrayList [points1](#), ArrayList [points2](#))

### Metody prywatne

- void [join](#) ()

### Atrybuty prywatne

- ArrayList [points1](#) = null  
*punkty dodawanej sciezki*

- `ArrayList points2 = null`  
*punkty sciezki, ktora juz jest w bazie*
- `double tol = 0.0002 * 0.0002`  
*tolerancja okreslajaca, jaka maksymalna odleglosc miedzy punktami sciezek podlega usrednianiu*
- `double[ ][ ] pdist = null`  
*odleglosci punktow pierwszej sciezki od punktow drugiej*
- `boolean changed = false`  
*flaga okreslajaca, czy jakies punkty ulegly scaleniu (jesli tak, to trzeba zmienic sciezke przebywajaca juz w bazie)*

### 3.14.1 Opis szczegółowy

Klasa odpowiedzialna za scalanie pobliskich sciezek tak, by na mapie wygladaly jak jedna i nie zaciemniały obrazu

#### Autor:

Agnieszka Holka

### 3.14.2 Dokumentacja funkcji składowych

#### 3.14.2.1 `boolean pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineMerger.merge (ArrayList points1, ArrayList points2)`

Główna metoda klasy `PolylineMerger`. Odpowiada za przebieg procesu usredniania sciezek, poprzez wywoływanie wyspecyfikowanych metod

#### Zwraca:

`true` jesli sciezki zostaly usrednione  
`false` jeli sciezki nie sa podobne i nie wywieraja na siebie plywu

#### 3.14.2.2 `void pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineMerger.join ()` [private]

Metoda przeglada wszystkie punkty pierwszej sciezki; szuka dla kazdego najblizszego sasiasda w drugiej lamanej, jesli odleglosc jest ponizej zadanego progu `tol`, to punkty zostaja scalone, aby na mapie czy wydruku pojawic sie w dokladnie tym samym miejscu i nie wplywac negatywnie na wizualizacje danych

Dokumentacja dla tej klasy zostala wygenerowana z pliku:

- `PolylineMerger.java`

## 3.15 Dokumentacja klasy `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSimplifier`

### Metody publiczne

- `ArrayList simplify (ArrayList al, ArrayList specialList)`

- void `reduceVertices` (ArrayList pts)
- void `dividePath` ()
- void `DouglasPeucker` (ArrayList pts)

#### Metody prywatne

- void `convertSpecial` (ArrayList specialList)
- void `markSpecialPoints` ()
- void `mediumDistance` (ArrayList al, double[] dist, double[] mdist)

#### Atrybuty prywatne

- ArrayList `points` = null  
*punkty pobrane z odczytu GPS, rozdzielone w razie potrzeby na kilka scieze (points przechowuje obiekty typu ArrayList)*
- HashMap `special` = null  
*punkty specjalne, informujace o waznych obiektach (wsp. geograficzne i znacznik)*
- int `local` = 20  
*liczba odcinkow po obu stronach majacych wplyw na badanie odleglosci miedzy konkretnymi dwoma punktami*
- int `minLength` = 3  
*minimalna ilosc punktow w pojedynczej sciezce*
- double `ratio` = 20.0  
*tolerowany stosunek odleglosci miedzy punktami do odleglosci sredniej*
- double `tv` = 0.000001 \* 0.000001  
*tolerancja wierzchołkow - kwadrat odleglosci, ponizej ktorej punkty sprowadzane sa do jednego*
- double `tl` = 0.00004 \* 0.00004  
*tolerancja linii - kwadrat odleglosci, ponizej ktorej punkt staje sie nowym rozdzielajacym (w metodzie DouglasPeucker)*

#### Komponenty

- class `SpecialTable`

##### 3.15.1 Opis szczegółowy

Klasa odpowiadajaca za algorytm upraszczania danych pobranych z aplikacji mobilnej, aby mozna bylo je wprowadzic do bazy danych.

#### Autor:

Agnieszka Holka

### 3.15.2 Dokumentacja funkcji składowych

#### 3.15.2.1 `ArrayList` `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSimplifier.simplify` (`ArrayList al`, `ArrayList specialList`)

Metoda poddaje uproszczeniu pobrana ścieżkę, wywołując odpowiednie algorytmy

**Parametry:**

*al* lista punktów składających się na ścieżkę

*specialList* lista punktów, w których zmienia się rodzaj nawierzchni

**Zwraca:**

lista ścieżek zawierających listy punktów (po przefiltrowaniu)

#### 3.15.2.2 `void` `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSimplifier.convertSpecial` (`ArrayList specialList`) [`private`]

Metoda zamienia listę punktów specjalnych (markujących jakość ścieżki) na mapę hashowana, aby dla dowolnego punktu ścieżki za pomocą prostego odwołania do obiektu klasy `HashMap` można było określić, czy jest punktem specjalnym.

**Parametry:**

*specialList* lista punktów specjalnych (współrzędnych i znaczników)

#### 3.15.2.3 `void` `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSimplifier.markSpecialPoints` () [`private`]

Przeglądane są punkty trasy i jeśli któryś znajduje się również wśród punktów specjalnych, znacznik jest dodawany do podstawowych danych o punkcie i przechowywany w liście `points`

#### 3.15.2.4 `void` `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSimplifier.reduceVertices` (`ArrayList pts`)

Sprawdzone są odległości pomiędzy kolejnymi punktami ścieżki i usunięte zostają wszystkie, które pojawiają się zbyt blisko siebie (dokładność z góry ustalona)

**Parametry:**

*pts* lista punktów pojedynczej ścieżki

#### 3.15.2.5 `void` `pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSimplifier.dividePath` ()

Jeśli ścieżka zawiera nieproporcjonalnie duże odległości pomiędzy niektórymi punktami, to prawdopodobnie wystąpiły przerwy w odczytach GPS; w takim wypadku metoda `dividePath` dzieli dane na więcej ścieżek, aby osobno poddać je uproszczeniu itp. i nie doprowadzić do przekłaman w bazie danych; dodatkowo usuwane są ścieżki, które miałyby mieć mniej niż `minLength` punktów.

**3.15.2.6 void pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSimplifier.DouglasPeucker (ArrayList pts)**

Algorytm upraszczania pojedynczej sciezki metoda Douglasa i Peuckera: poczatkowy i koncowy punkt trasy laczone sa odcinkiem i zostaja obliczone odleglosci pozostalych punktow od niego. Jesli niektore nie mieszczą sie w zadanym przedziale, to najdalszy staje sie nowym punktem podzialu; wowczas powstale dwa przedzialy odkladane sa na stos do powtornej obrobki; algorytm dziala dopoki na stosie znajduja sie odcinki.

**Parametry:**

*pts* pojedyncza sciezka

**Zwraca:**

lista punktow w uproszczonej sciezce

**3.15.2.7 void pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSimplifier.mediumDistance (ArrayList al, double[] dist, double[] mdist) [private]**

Metoda okresla odelglosci pomiedzy kolejnym punktami w odczycie i oblicza lokalne srednie

**Parametry:**

*al* lista punktow

*dist* tablica odleglosci pomiedzy kolejnymi punktami

*mdist* tablica lokalnych srednich odleglosci pomiedzy kolejnymi punktami

Dokumentacja dla tej klasy zostala wygenerowana z pliku:

- PolylineSimplifier.java

**3.16 Dokumentacja klasy pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSimplifier.SpecialTable****3.16.1 Opis szczegółowy**

Klasa sluzaca do przechowywania informacji o punktach specjalnych

Dokumentacja dla tej klasy zostala wygenerowana z pliku:

- PolylineSimplifier.java

**3.17 Dokumentacja klasy pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSmoother****Metody publiczne**

- ArrayList [smooth](#) (ArrayList pts)
- ArrayList [McMaster](#) ()

### Atrybuty prywatne

- `ArrayList points` = null  
*ścieżka pobrana z bazy danych*
- `ArrayList smoothed` = null  
*lista punktów po wygładzeniu*
- `int neighbour` = 5  
*liczba punktów (nieparzysta) brana do usredniania pojedynczego punktu*
- `double attr` = 0.75f  
*sila przyciągania srodkowego punktu podczas wygładzania*

#### 3.17.1 Opis szczegółowy

Klasa odpowiedzialna za wygładzanie ścieżek z bazy danych przed przesłaniem ich do serwisu internetowego lub do eksportu; służy do lepszego wyświetlania ścieżek, natomiast nie ma wpływu na kształt tras przechowywanych w bazie

#### Autor:

Agnieszka Holka

#### 3.17.2 Dokumentacja funkcji składowych

##### 3.17.2.1 `ArrayList pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSmoother.smooth (ArrayList pts)`

Metoda wygładzająca ścieżki, aby lepiej wyglądały na mapie; w bazie danych ścieżki mają usunięte niepotrzebne punkty, są usrednione (jeśli opisują tę samą trasę), ale wygładzanie jest wykonywane jedynie podczas pobierania danych do serwisu internetowego / eksportu do formatów zewnętrznych; ma to zapobiec niedokładnościom w bazie danych

#### Parametry:

*pts* lista punktów do wygładzenia

#### Zwraca:

punkty w postaci listy współrzędnych

##### 3.17.2.2 `ArrayList pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PolylineSmoother.McMaster ()`

Algorytm McMastera wygładzania łamanych

#### Zwraca:

wygładzone punkty w postaci listy

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `PolylineSmoother.java`

## 3.18 Dokumentacja klasy pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PostgresBase

Diagram dziedziczenia dla pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PostgresBaseDiagram współpracy dla pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PostgresBase:

### Metody publiczne

- [PostgresBase](#) (String ip, int num1, int what1, String name, String user, String pass)

#### 3.18.1 Opis szczegółowy

##### Autor:

Ania Gogolinska

#### 3.18.2 Dokumentacja konstruktora i destruktora

##### 3.18.2.1 pl.umk.mat.pz2006.nebuchadrezzar.hammurabi.PostgresBase.PostgresBase (String ip, int num1, int what1, String name, String user, String pass)

Stworzenie obiektu odpowiedzialnego za komunikacje z postgresowa baza danych. Nawiązanie polaczenia z baza.

##### Parametry:

- ip* Adres ip komputera na ktory znajduje sie baza.
- num1* Numer oznaczajacy rodzaj punktu z dodatkowym opisem.
- what1* Numer oznaczajacy ostatni rodzaj punktu, okreslajacy rodzaj drogi.
- name* Nazwa bazy danych.
- user* Nazwa uzytkownika bazy danych.
- pass* Haslo uzytkownia bazy danych.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- PostgresBase.java

## 3.19 Dokumentacja klasy pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter

Diagram dziedziczenia dla pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporterDiagram współpracy dla pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter:

### Metody chronione

- abstract void [paintLatLng](#) ()
- abstract void [paintPathes](#) ()
- abstract void [markSpecialPoints](#) ()
- abstract void [paintTitle](#) ()
- abstract void [createLegend](#) ()
- void [countMapParams](#) ()



### Atrybuty chronione

- double `lngCenter`  
*dlugosc i szerokosc geograficzna srodka mapy*
- double `span`  
*rozpietosc mapy (w stopniach)*
- float `xcenter` = 367  
*stosunek pikseli na papierze do rozpietosci na mapie dla szerokosci i dlugosci geograficznej wspolrzeczna X srodka mapy (w pikselach)*
- float `ycenter` = 295  
*wspolrzeczna Y srodka mapy (w pikselach)*
- float `xspan` = 313  
*rozpietosc wspolrzecznej X na mapie mapy (w pikselach)*
- float `yspan` = 241  
*rozpietosc wspolrzecznej Y na mapie mapy (w pikselach)*
- String `logo` = "images/logo.png"  
*sciezka do pliku zawierajacego logo projektu*

#### 3.19.1 Opis szczegółowy

Klasa abstrakcyjna reprezentujaca eksport sciezek do plikow drukowalnych (postscript, PDF)

##### Autor:

Agnieszka Holka

#### 3.19.2 Dokumentacja funkcji składowych

##### 3.19.2.1 `abstract void pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter.paintLatLng ()` [protected, pure virtual]

Metoda nanoszaca na mapie siatke poludnikow i rownoleznikow wraz z wartosciami szerokosci i dlugosci geograficznej

Implementowany w [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter](#) i [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporter](#).

##### 3.19.2.2 `abstract void pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter.paintPathes ()` [protected, pure virtual]

Metoda, ktora rysuje na mapie sciezki podane w postaci listy punktow

Implementowany w [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter](#) i [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporter](#).

**3.19.2.3 abstract void `pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter.markSpecialPoints ()`** [`protected`, `pure virtual`]

Zaznacza na mapie wszystkie punkty specjalne i dopisuje im etykiety

Implementowany w [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter](#) i [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporter](#).

**3.19.2.4 abstract void `pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter.paintTitle ()`** [`protected`, `pure virtual`]

Tytuł i logo projektu jako nagłówek strony

Implementowany w [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter](#) i [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporter](#).

**3.19.2.5 abstract void `pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter.createLegend ()`** [`protected`, `pure virtual`]

Tworzy legende do mapy i wyswietla ja z boku strony

Implementowany w [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PDFExporter](#) i [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporter](#).

**3.19.2.6 void `pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter.countMapParams ()`** [`protected`]

Metoda oblicza wspolrzedne geograficzne srodka mapy oraz wspolczynniki potrzebne do poprawnego i proporcjonalnego rysowania obiektow na mapie

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `PrintableExporter.java`

**3.20 Dokumentacja klasy `pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporter`**

Diagram dziedziczenia dla `pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporterDiagram` współpracy dla `pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporter`:

**Metody publiczne**

- `byte[] export (ArrayList pts, ArrayList sp, double[] brd)`

**Metody chronione**

- void `paintLatLng ()`
- void `paintPathes ()`
- void `markSpecialPoints ()`
- void `paintTitle ()`
- void `createLegend ()`

**Metody prywatne**

- void `showText (String text, String x, String y, int loop)`

### Atrybuty prywatne

- String `ps`

*lancuch znakow zawierajacy program postscript, zapisywany do pliku wynikowego*

#### 3.20.1 Opis szczegółowy

Klasa eksportujaca podana liste sciezek i punktow specjalnych z danego obszaru do formatu postscript

#### Autor:

Agnieszka Holka

#### 3.20.2 Dokumentacja funkcji składowych

##### 3.20.2.1 `byte [] pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporter.export (ArrayList pts, ArrayList sp, double[] brd)` [virtual]

Metoda odpowiada za inicjalizacje zmiennych potrzebnych do wyeksportowania konkretnego zestawu sciezek. Wprowadza podstawowe ustawienia do pliku postscriptowego (podstawowy schemat dokumentu, podzial na sekcje (naglowek, wlasciwa mapa, legenda), formatowanie czcionki itp.). Wywołuje wyspecyfikowane metody, odpowiedzialne za nanoszenie sciezek, punktow specjalnych, siatki oraz legendy. Ostatecznie zapisuje ustalony kod postscript do pliku.

#### Parametry:

*pts* lista sciezek w postaci list punktow

*sp* lista punktow specjalnych

*brd* granice tworzonej mapy

#### Zwraca:

tablica bajtow zawierajaca tresc wynikowego pliku

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.Exporter](#).

##### 3.20.2.2 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporter.paintLatLng` () [protected, virtual]

Metoda nanoszaca na mapie siatke poludnikow i rownoleznikow wraz z wartosciami szerokosci i dlugosci geograficznej

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter](#).

##### 3.20.2.3 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporter.paintPathes` () [protected, virtual]

Metoda, ktora rysuje na mapie sciezki podane w postaci listy punktow

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter](#).

**3.20.2.4** void `pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporter.markSpecialPoints` ()  
[protected, virtual]

Zaznacza na mapie wszystkie punkty specjalne i dopisuje im etykiety

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter](#).

**3.20.2.5** void `pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporter.paintTitle` ()  
[protected, virtual]

Tytuł i logo projektu jako nagłówek strony

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter](#).

**3.20.2.6** void `pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporter.createLegend` ()  
[protected, virtual]

Tworzy legende do mapy i wyswietla ja z boku strony

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.PrintableExporter](#).

**3.20.2.7** void `pl.umk.mat.pz2006.nebuchadrezzar.marduk.PSExporter.showText` (String *text*,  
String *x*, String *y*, int *loop*) [private]

Wyswietla tekst w okreslonych wspolrzednych, jednoczesnie zachpuwujac obramowanie tekstu (w celu przejrzystosci wydruku)

Dokumentacja dla tej klasy zostala wygenerowana z pliku:

- `PSExporter.java`

## 3.21 Dokumentacja klasy `pl.umk.mat.pz2006.nebuchadrezzar.marduk.TXTExporter`

Diagram dziedziczenia dla `pl.umk.mat.pz2006.nebuchadrezzar.marduk.TXTExporterDiagram` współpracy dla `pl.umk.mat.pz2006.nebuchadrezzar.marduk.TXTExporter`:

### Metody publiczne

- `byte[] export` (ArrayList *pts*, ArrayList *sp*, double[] *brd*)

#### 3.21.1 Opis szczegółowy

Klasa obsługująca eksport do pliku tekstowego, zachowującego format wymagany przez aplikacje mobilna.

#### Autor:

Agnieszka Holka

#### 3.21.2 Dokumentacja funkcji składowych

**3.21.2.1** `byte[] pl.umk.mat.pz2006.nebuchadrezzar.marduk.TXTExporter.export` (ArrayList *pts*,  
ArrayList *sp*, double[] *brd*) [virtual]

Metoda wprowadza na początek pliku znacznik `<Nabuchodonozor>` , dalej wprowadza liste punktow nalezacych do kolejnych sciezek. Pierwszy punkt kazdej nowej sciezki (poza pierwsza) poprzedzony jest znakiem `&` . Na koncu pliku pojawiaja sie wszystkie marki.

Implementuje [pl.umk.mat.pz2006.nebuchadrezzar.marduk.Exporter](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `TXTEporter.java`

## 3.22 Dokumentacja klasy `pl.umk.mat.pz2006.nebuchadrezzar.marduk.TXTParser`

### Metody publiczne

- `TXTParser ()`
- `void parse (BufferedReader br, int num, int what) throws Exception`
- `ArrayList getPoints ()`
- `ArrayList getSpecial ()`

### Atrybuty pakietu

- `ArrayList points`
- `ArrayList special`

#### 3.22.1 Opis szczegółowy

##### Autor:

Ania Gogolinska

#### 3.22.2 Dokumentacja konstruktora i destruktoru

##### 3.22.2.1 `pl.umk.mat.pz2006.nebuchadrezzar.marduk.TXTParser.TXTParser ()`

Stworzenia nowego parsera, zainicjalizowanie list.

#### 3.22.3 Dokumentacja funkcji składowych

##### 3.22.3.1 `void pl.umk.mat.pz2006.nebuchadrezzar.marduk.TXTParser.parse (BufferedReader br, int num, int what) throws Exception`

Metoda pasujaca dane czytanie ze strumienia. Dane wpisywane sa do dwoch list points i special.

##### Parametry:

- br* `BufferedReader` podpiety do strumienia z ktorego sa czytane dane.
- num* Numer ktory oznacza typ punktu z opisem.

### 3.22.3.2 `ArrayList` `pl.umk.mat.pz2006.nebuchadrezzar.marduk.TXTParser.getPoints ()`

Metoda zwracająca listę punktów `points`.

**Zwraca:**

Lista `points`.

### 3.22.3.3 `ArrayList` `pl.umk.mat.pz2006.nebuchadrezzar.marduk.TXTParser.getSpecial ()`

Metoda zwracająca listę punktów specjalnych.

**Zwraca:**

Lista `special`.

### 3.22.4 Dokumentacja atrybutów składowych

#### 3.22.4.1 `ArrayList` `pl.umk.mat.pz2006.nebuchadrezzar.marduk.TXTParser.points` [package]

Lista punktów, które mają być wstawione do bazy. Zawiera ona trzelementowe tablice zmiennych typu `double`, zawierające szerokość geograficzną, długość geograficzną, wysokość.

#### 3.22.4.2 `ArrayList` `pl.umk.mat.pz2006.nebuchadrezzar.marduk.TXTParser.special` [package]

Lista punktów specjalnych, zawiera ona w kolejności szerokość geograficzną (`double`), długość (`double`), typ punktu oraz ewentualnie opis punktu (`String`).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `TXTParser.java`

## 3.23 Dokumentacja klasy `pl.umk.mat.pz2006.nebuchadrezzar.marduk.XMLExporter`

Diagram dziedziczenia dla `pl.umk.mat.pz2006.nebuchadrezzar.marduk.XMLExporterDiagram` współpracy dla `pl.umk.mat.pz2006.nebuchadrezzar.marduk.XMLExporter`:

### 3.23.1 Opis szczegółowy

Klasa abstrakcyjna zawierająca nagłówki metod wymaganych podczas eksportu danych do typów GPX i KLM

**Autor:**

Agnieszka Holka

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `XMLExporter.java`

## 3.24 Dokumentacja klasy pl.umk.mat.pz2006.nebuchadrezzar.marduk.XMLMarkers

### Statyczne atrybuty publiczne

- static final String `gpxPathMarker` = "trk"  
*znacznik trasy*
- static final String `gpxSegMarker` = "trkseg"  
*znacznik fragmentu trasy*
- static final String `gpxPointMarker` = "trkpt"  
*znacznik punktu nalezacego do trasy*
- static final String `gpxPath2Marker` = "rte"  
*znacznik trasy (inny sposob)*
- static final String `gpxPoint2Marker` = "rtept"  
*znacznik punktu nalezacego do trasy*
- static final String `gpxLatMarker` = "lat"  
*znacznik szerokosci geograficznej*
- static final String `gpxLngMarker` = "lon"  
*znacznik wysokosci geograficznej*
- static final String `gpxAltMarker` = "ele"  
*znacznik wysokosci n.p.m.*
- static final String `gpxSpecialMarker` = "wpt"  
*znacznik okreslajacy punkt specjalny*
- static final String `gpxNameMarker` = "name"  
*znacznik nazwy punktu*
- static final String `gpxDescMarker` = "desc"  
*znacznik komentarza do punktu*

### 3.24.1 Opis szczegółowy

Klasa zawierajaca definicje znacznikow wykorzystywanych w formatach XML

**Autor:**

Agnieszka Holka

### 3.24.2 Dokumentacja atrybutów składowych

**3.24.2.1** `final String pl.umk.mat.pz2006.nebuchadrezzar.marduk.XMLMarkers.gpxPathMarker = "trk"` [static]

znacznik trasy

Znaczniki wykorzystywane w formacie GPX

Dokumentacja dla tej klasy została wygenerowana z pliku:

- XMLMarkers.java