

Epidemia

Wygenerowano przez Doxygen 1.8.11

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 CommonDisease	5
3.1.1	Dokumentacja funkcji składowych	5
3.1.1.1	tryAttack()	5
3.2	Dokumentacja klasy CommonPerson	6
3.2.1	Dokumentacja funkcji składowych	6
3.2.1.1	attack()	6
3.2.1.2	attackChance()	6
3.2.1.3	description()	6
3.2.1.4	dodgeChance()	6
3.2.1.5	intelligence()	7
3.2.1.6	who()	7
3.3	Dokumentacja klasy Country	7
3.3.1	Opis szczegółowy	7
3.3.2	Dokumentacja funkcji składowych	7
3.3.2.1	cure()	7
3.3.2.2	cureSomeInfectedPersons(fstream &output)	8
3.3.2.3	description()	8

3.3.2.4	infectedPersons()	8
3.3.2.5	InfectPerson(std::vector< Person * >::iterator personOpponent)	8
3.3.2.6	persons()	8
3.3.2.7	searchForCure(fstream &output)	8
3.3.2.8	showEveryone()	8
3.3.2.9	spreadVirus(int i, fstream &output)	8
3.4	Dokumentacja klasy Disease	9
3.4.1	Opis szczegółowy	9
3.4.2	Dokumentacja funkcji składowych	9
3.4.2.1	tryAttack()=0	9
3.5	Dokumentacja klasy Mutagen	9
3.5.1	Opis szczegółowy	10
3.5.2	Dokumentacja funkcji składowych	10
3.5.2.1	tryAttack()	10
3.6	Dokumentacja klasy Person	10
3.6.1	Opis szczegółowy	11
3.6.2	Dokumentacja funkcji składowych	11
3.6.2.1	attack()=0	11
3.6.2.2	attackChance()=0	11
3.6.2.3	damage(int dmg)	11
3.6.2.4	description()=0	11
3.6.2.5	dodgeChance()=0	11
3.6.2.6	intelligence()=0	11
3.6.2.7	isDead()	11
3.6.2.8	personHP()	12
3.6.2.9	who()=0	12
3.7	Dokumentacja klasy Scientist	12
3.7.1	Opis szczegółowy	12
3.7.2	Dokumentacja funkcji składowych	12
3.7.2.1	description()	12

3.7.2.2	intelligence()	13
3.7.2.3	who()	13
3.8	Dokumentacja klasy UltraVegan	13
3.8.1	Opis szczegółowy	13
3.8.2	Dokumentacja funkcji składowych	13
3.8.2.1	tryAttack()	13
3.9	Dokumentacja klasy Vegan	14
3.9.1	Opis szczegółowy	14
3.9.2	Dokumentacja funkcji składowych	14
3.9.2.1	attack()	14
3.9.2.2	attackChance()	14
3.9.2.3	description()	14
3.9.2.4	dodgeChance()	15
3.9.2.5	intelligence()	15
3.9.2.6	who()	15
Indeks		17

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

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

Country	7
Disease	9
CommonDisease	5
Mutagen	9
UltraVegan	13
Person	10
CommonPerson	6
Scientist	12
Vegan	14

Rozdział 2

Indeks klas

2.1 Lista klas

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

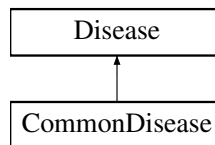
CommonDisease	5
CommonPerson	6
Country	7
Disease	9
Mutagen	9
Person	10
Scientist	12
UltraVegan	13
Vegan	14

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja klasy CommonDisease

Diagram dziedziczenia dla CommonDisease



Metody publiczne

- int **attackChance** ()
- virtual int **who** ()
- virtual bool **tryAttack** ()
- virtual string **description** ()

3.1.1 Dokumentacja funkcji składowych

3.1.1.1 `bool CommonDisease::tryAttack () [virtual]`

*0-CommonDisease 1-UltraVegan 2-Mutagen

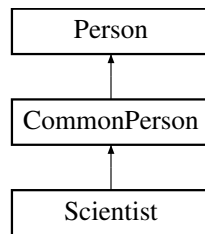
Implementuje [Disease](#).

Dokumentacja dla tej klasy została wygenerowana z plików:

- Disease.h
- Disease.cpp

3.2 Dokumentacja klasy CommonPerson

Diagram dziedziczenia dla CommonPerson



Metody publiczne

- int `attack` ()
- int `attackChance` ()
- int `dodgeChance` ()
- int `intelligence` ()
- virtual int `who` ()
- virtual string `description` ()

3.2.1 Dokumentacja funkcji składowych

3.2.1.1 int `CommonPerson::attack` () [`inline`],[`virtual`]

*Zadaje dmg osobom.

Implementuje `Person`.

3.2.1.2 int `CommonPerson::attackChance` () [`inline`],[`virtual`]

*Atak wyprowadzany przez osobę

Implementuje `Person`.

3.2.1.3 string `CommonPerson::description` () [`virtual`]

*0-Common 1-Scientist 2-Vegans

Implementuje `Person`.

Reimplementowana w `Scientist`.

3.2.1.4 int `CommonPerson::dodgeChance` () [`inline`],[`virtual`]

*Szansa na atak

Implementuje `Person`.

3.2.1.5 `int CommonPerson::intelligence () [inline],[virtual]`

*Szansa na unik

Implementuje [Person](#).

Reimplementowana w [Scientist](#).

3.2.1.6 `virtual int CommonPerson::who () [inline],[virtual]`

*Inteligencja

Implementuje [Person](#).

Reimplementowana w [Scientist](#).

Dokumentacja dla tej klasy została wygenerowana z plików:

- [Person.h](#)
- [Person.cpp](#)

3.3 Dokumentacja klasy Country

```
#include <Country.h>
```

Metody publiczne

- **Country** (unsigned int population, int sl, int vl, int zl)
- int [persons](#) ()
- int [infectedPersons](#) ()
- int [cure](#) ()
- void [InfectPerson](#) (std::vector< [Person](#) * >::iterator personOpponent)
- void [spreadVirus](#) (int i, fstream &output)
- bool [searchForCure](#) (fstream &output)
- void [cureSomeInfectedPersons](#) (fstream &output)
- string [description](#) ()
- string [showEveryone](#) ()

3.3.1 Opis szczegółowy

*Klasa zarządzająca przebiegiem zdarzeń w ciągu jednego dnia, posiada ona parametry determinujące szanse człowieka na wygraną w walce z chorobą

3.3.2 Dokumentacja funkcji składowych

3.3.2.1 `int Country::cure ()`

*Zwraca rozmiar wektora osób zainfekowanych

3.3.2.2 void Country::cureSomeInfectedPersons (fstream & output)

*Naukowcy szukają lekarstwa. Powodzenie poszukiwań zależne jest od wartości ich inteligencji, a jego wartość jest losowana pod koniec każdego dnia i odejmowana od puli (10000 to początkowa wartość puli)

3.3.2.3 string Country::description ()

*Jeśli wartość `_cure` będzie równa 0 funkcja ta zaczyna leczyć ludzi. Lekarstwo jest losowo rozpraszane na kilka (ilość losowa) jednostek, a każda z nich ma 50% szans na powrót do zdrowia

3.3.2.4 int Country::infectedPersons ()

*Zwraca rozmiar wektora osób

3.3.2.5 void Country::InfectPerson (std::vector< Person * >::iterator personOpponent)

*Zwraca wartość postępu szukania lekarstwa (od 10000 do 0)

3.3.2.6 int Country::persons ()

*Przeprowadza całą symulację trwania epidemii, zależną od wartości początkowych jej zmiennych.

3.3.2.7 bool Country::searchForCure (fstream & output)

*Zaraża losowo wybraną osobę co 4 dni. Czyli symulacja naturalnego rozprzestrzeniania się wirusa

3.3.2.8 string Country::showEveryone ()

*Zwraca ilość osób zdrowych, chorych, postęp szukania lekarstwa itp. używana do wyświetlenia wartości początkowych na początku trwania programu

3.3.2.9 void Country::spreadVirus (int i, fstream & output)

*W przypadku walki wygranej przez chorobę, funkcja ta zaraża człowieka chorobą z którą przegrał walkę. `void fight(fstream &output, int outputInfo);` Symuluje walkę pomiędzy osobą a chorobą. Choroba nie zadaje osobie żadnych obrażeń, w przypadku wygranej po prostu zaraża osobę, natomiast w przypadku przegranej jednostka tej choroby ginie. Jeśli wygrywa osoba, to nie zostaje zarażona. Wynik walki jest uzależniony od parametrów zarówno choroby, jak i osoby, wybieranych losowo, jednak z innym przedziałem losowości w przypadku różnych klas chorób i osób.

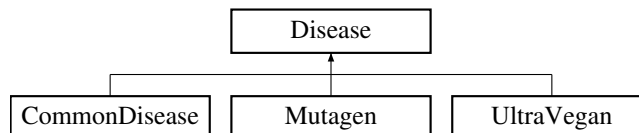
Dokumentacja dla tej klasy została wygenerowana z plików:

- Country.h
- Country.cpp

3.4 Dokumentacja klasy Disease

```
#include <Disease.h>
```

Diagram dziedziczenia dla Disease



Metody publiczne

- int **diseaseID** ()
- int **diseaseHP** ()
- void **damage** (int dmg)
- bool **isDead** ()
- virtual int **attackChance** ()=0
- virtual int **who** ()=0
- virtual bool **tryAttack** ()=0
- virtual string **description** ()=0

3.4.1 Opis szczegółowy

*Klasa choroby, posiada parametry takie jak HP i ID choroby oraz cechy choroby od których zależy przebieg walki między nią a człowiekiem.

3.4.2 Dokumentacja funkcji składowych

3.4.2.1 virtual bool Disease::tryAttack () [pure virtual]

*0-CommonDisease 1-UltraVegan 2-Mutagen

Implementowany w [Mutagen](#), [UltraVegan](#) i [CommonDisease](#).

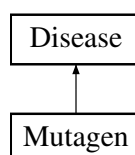
Dokumentacja dla tej klasy została wygenerowana z plików:

- Disease.h
- main.cpp

3.5 Dokumentacja klasy Mutagen

```
#include <Disease.h>
```

Diagram dziedziczenia dla Mutagen



Metody publiczne

- int **attackChance** ()
- int **attackCount** ()
- virtual int **who** ()
- virtual bool **tryAttack** ()
- virtual string **description** ()

3.5.1 Opis szczegółowy

*Klasa wirusa który sier... ucieka w te pędy, gdy spotka vegana (tak na prawdę to każda osobę którą spotyka) zarażając po drodze kilka innych osób.

3.5.2 Dokumentacja funkcji składowych

3.5.2.1 bool Mutagen::tryAttack() [virtual]

*0-CommonDisease 1-UltraVegan 2-Mutagen

Implementuje [Disease](#).

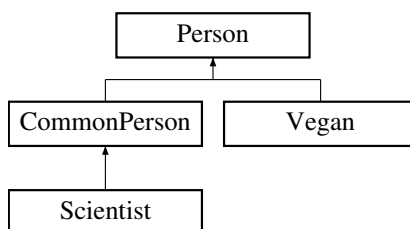
Dokumentacja dla tej klasy została wygenerowana z plików:

- Disease.h
- Disease.cpp

3.6 Dokumentacja klasy Person

```
#include <Person.h>
```

Diagram dziedziczenia dla Person



Metody publiczne

- int **personID** ()
- int **personHP** ()
- int **isDead** ()
- void **damage** (int dmg)
- virtual int **attack** ()=0
- virtual int **attackChance** ()=0
- virtual int **dodgeChance** ()=0
- virtual int **intelligence** ()=0
- virtual int **who** ()=0
- virtual string **description** ()=0

3.6.1 Opis szczegółowy

*Klasa osób. Posiada wartości ID, HP oraz kilka przydatnych metod.

3.6.2 Dokumentacja funkcji składowych

3.6.2.1 `virtual int Person::attack () [pure virtual]`

*Zadaje dmg osobom.

Implementowany w [Vegan](#) i [CommonPerson](#).

3.6.2.2 `virtual int Person::attackChance () [pure virtual]`

*Atak wyprowadzany przez osobę

Implementowany w [Vegan](#) i [CommonPerson](#).

3.6.2.3 `void Person::damage (int dmg) [inline]`

*Sprawdza, czy osoba jest martwa.

3.6.2.4 `virtual string Person::description () [pure virtual]`

*0-Common 1-Scientist 2-Vegans

Implementowany w [Vegan](#), [Scientist](#) i [CommonPerson](#).

3.6.2.5 `virtual int Person::dodgeChance () [pure virtual]`

*Szansa na atak

Implementowany w [Vegan](#) i [CommonPerson](#).

3.6.2.6 `virtual int Person::intelligence () [pure virtual]`

*Szansa na unik

Implementowany w [Vegan](#), [Scientist](#) i [CommonPerson](#).

3.6.2.7 `int Person::isDead () [inline]`

*Zwraca HP osoby.

3.6.2.8 `int Person::personHP () [inline]`

*Zwraca ID osoby.

3.6.2.9 `virtual int Person::who () [pure virtual]`

*Inteligencja

Implementowany w [Vegan](#), [Scientist](#) i [CommonPerson](#).

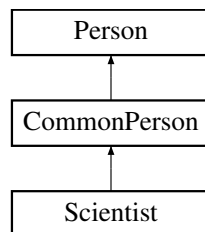
Dokumentacja dla tej klasy została wygenerowana z plików:

- [Person.h](#)
- [main.cpp](#)

3.7 Dokumentacja klasy Scientist

```
#include <Person.h>
```

Diagram dziedziczenia dla Scientist



Metody publiczne

- `int intelligence ()`
- `virtual int who ()`
- `virtual string description ()`

3.7.1 Opis szczegółowy

*Klasa przeciętnej osoby. Taka osoba ma mniejsze szanse na wygraną walkę z chorobą.

3.7.2 Dokumentacja funkcji składowych

3.7.2.1 `string Scientist::description () [virtual]`

*0-Common 1-Scientist 2-Vegans

Reimplementowana z [CommonPerson](#).

3.7.2.2 `int Scientist::intelligence () [inline],[virtual]`

*Szansa na unik

Reimplementowana z [CommonPerson](#).

3.7.2.3 `virtual int Scientist::who () [inline],[virtual]`

*Inteligencja

Reimplementowana z [CommonPerson](#).

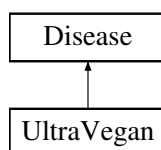
Dokumentacja dla tej klasy została wygenerowana z plików:

- Person.h
- Person.cpp

3.8 Dokumentacja klasy UltraVegan

```
#include <Disease.h>
```

Diagram dziedziczenia dla UltraVegan



Metody publiczne

- `int attackChance ()`
- `virtual int who ()`
- `virtual bool tryAttack ()`
- `virtual string description ()`

3.8.1 Opis szczegółowy

*Klasa przeciętnego rodzaju choroby, nie boi się ona vegan

3.8.2 Dokumentacja funkcji składowych

3.8.2.1 `bool UltraVegan::tryAttack () [virtual]`

*0-CommonDisease 1-UltraVegan 2-Mutagen

Implementuje [Disease](#).

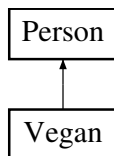
Dokumentacja dla tej klasy została wygenerowana z plików:

- Disease.h
- Disease.cpp

3.9 Dokumentacja klasy Vegan

```
#include <Person.h>
```

Diagram dziedziczenia dla Vegan



Metody publiczne

- int [attack](#) ()
- int [attackChance](#) ()
- int [dodgeChance](#) ()
- int [intelligence](#) ()
- virtual int [who](#) ()
- virtual string [description](#) ()

3.9.1 Opis szczegółowy

*Klasa naukowców. Są oni zupełnie bezbronni, służą tylko znalezieniu lekarstwa oraz jako jedyni posiadają inteligencję.

3.9.2 Dokumentacja funkcji składowych

3.9.2.1 int [Vegan::attack](#) () [inline],[virtual]

*Zadaje dmg osobom.

Implementuje [Person](#).

3.9.2.2 int [Vegan::attackChance](#) () [inline],[virtual]

*Atak wyprowadzany przez osobę

Implementuje [Person](#).

3.9.2.3 string [Vegan::description](#) () [virtual]

*0-Common 1-Scientist 2-Vegans

Implementuje [Person](#).

3.9.2.4 `int Vegan::dodgeChance () [inline],[virtual]`

*Szansa na atak

Implementuje [Person](#).

3.9.2.5 `int Vegan::intelligence () [inline],[virtual]`

*Szansa na unik

Implementuje [Person](#).

3.9.2.6 `virtual int Vegan::who () [inline],[virtual]`

*Inteligencja

Implementuje [Person](#).

Dokumentacja dla tej klasy została wygenerowana z plików:

- [Person.h](#)
- [Person.cpp](#)

Skorowidz

attack

CommonPerson, 6
Person, 11
Vegan, 14

attackChance

CommonPerson, 6
Person, 11
Vegan, 14

CommonDisease, 5

tryAttack, 5

CommonPerson, 6

attack, 6
attackChance, 6
description, 6
dodgeChance, 6
intelligence, 6
who, 7

Country, 7

cure, 7
cureSomeInfectedPersons, 7
description, 8
InfectPerson, 8
infectedPersons, 8
persons, 8
searchForCure, 8
showEveryone, 8
spreadVirus, 8

cure

Country, 7

cureSomeInfectedPersons

Country, 7

damage

Person, 11

description

CommonPerson, 6
Country, 8
Person, 11
Scientist, 12
Vegan, 14

Disease, 9

tryAttack, 9

dodgeChance

CommonPerson, 6
Person, 11
Vegan, 14

InfectPerson

Country, 8

infectedPersons

Country, 8

intelligence

CommonPerson, 6
Person, 11
Scientist, 12
Vegan, 15

isDead

Person, 11

Mutagen, 9

tryAttack, 10

Person, 10

attack, 11
attackChance, 11
damage, 11
description, 11
dodgeChance, 11
intelligence, 11
isDead, 11
personHP, 11
who, 12

personHP

Person, 11

persons

Country, 8

Scientist, 12

description, 12
intelligence, 12
who, 13

searchForCure

Country, 8

showEveryone

Country, 8

spreadVirus

Country, 8

tryAttack

CommonDisease, 5
Disease, 9
Mutagen, 10
UltraVegan, 13

UltraVegan, 13

tryAttack, 13

Vegan, 14

attack, 14
attackChance, 14

description, [14](#)
dodgeChance, [14](#)
intelligence, [15](#)
who, [15](#)

who

CommonPerson, [7](#)
Person, [12](#)
Scientist, [13](#)
Vegan, [15](#)