Plecak

Generated by Doxygen 1.9.8

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Class Documentation

3.1 Backpack Struct Reference

Struktura pelniaca role plecaka z przedmiotami.

`#include <plecak.h>`

Public Attributes

- std::list< Item > items = std::list<Item>()
- float capacity = 0
- int value = 0

3.1.1 Detailed Description

Struktura pelniaca role plecaka z przedmiotami.

Parameters

items	Lista typu Items przechowujaca przedmioty znajdujace sie w plecaku.
capacity	Zmienna float okreslajaca pojemnosc plecaka.
value	Zmienna int okreslajaca wartosc plecaka.

3.1.2 Member Data Documentation

3.1.2.1 capacity

```
float Backpack::capacity = 0
```

3.1.2.2 items

```
std::list<Item> Backpack::items = std::list<Item>()
```

3.1.2.3 value

```
int Backpack::value = 0
```

The documentation for this struct was generated from the following file:

- plecak.h

3.2 Item Struct Reference

struktura przechowujaca dane z pliku powiazanego z przelacznikiem -i.

```
#include <plecak.h>
```

Public Attributes

- std::string name = ""
- float weight = 0
- float price = 0

3.2.1 Detailed Description

struktura przechowujaca dane z pliku powiazanego z przelacznikiem -i.

Parameters

name	Zmienna typu string przyjmujaca nazwe przedmiotu.
weight	Zmienna typu float przyjmujaca wage przedmiotu.
price	Zmienna typu float przyjmujaca cene przedmiotu.

3.2.2 Member Data Documentation

3.2.2.1 name

```
std::string Item::name = ""
```

3.2.2.2 price

```
float Item::price = 0
```

3.2.2.3 weight

```
float Item::weight = 0
```

The documentation for this struct was generated from the following file:

- plecak.h

3.3 Params Struct Reference

Struktura przyjmujaca wprowadzone dane - odpowiednie dla przelacznikow nazwy plikow i wartosci - do programu.

`#include <plecak.h>`

Public Attributes

- std::string itemsFile
- std::string solutionFile
- float capacity
- int generations
- int amountOfSpecimens

3.3.1 Detailed Description

Struktura przyjmujaca wprowadzone dane - odpowiednie dla przelacznikow nazwy plikow i wartosci - do programu.

Parameters

itemsFile	Zmienna typu string przyjmujaca nazwe pliku z przedmiotami.
solutionFile	Zmienna typu string przyjmujaca nazwe pliku z zapisem wyniku.
float	Zmienna typu float przyjmujaca pojemnosc plecaka.
generations	Zmienna typu int przyjmujaca ilosc generacji.
amountOfSpecimens	Zmienna typu int przyjmujaca ilosc osobnikow w pokoleniu.

3.3.2 Member Data Documentation

3.3.2.1 amountOfSpecimens

```
int Params::amountOfSpecimens
```

3.3.2.2 capacity

```
float Params::capacity
```

3.3.2.3 generations

```
int Params::generations
```

3.3.2.4 itemsFile

```
std::string Params::itemsFile
```

3.3.2.5 solutionFile

```
std::string Params::solutionFile
```

The documentation for this struct was generated from the following file:

- plecak.h

Chapter 4

File Documentation

4.1 main.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <list>
#include "plecak.h"
```

Functions

- int main (int argc, char ∗argv[ ])

4.1.1 Function Documentation

4.1.1.1 main()

```
int main (
            int argc,
            char * argv[] )
```

4.2 plecak.cpp File Reference

```
#include <iostream>
#include <vector>
#include <string>
#include <fstream>
#include <iomanip>
#include "plecak.h"
#include <random>
#include <sstream>
```

Functions

- bool Load_parameters (Params &param, const int &Liczba_parametrow, char *parametry[])

  Funkcja sprawujaca kontrole nad wlasciwym dostarczeniem przelacznikow do programu.
- void move_items (std::list< Item > &items, std::ifstream &input)

  Funkcja dostarczajaca dane z przelacznika - i do Listy typu Items w strukturze Backpack.
- void to_file (Backpack &backpack, std::ofstream &output_o, Params &params)

  Funkcja zapisujaca wynik programu do pliku o nazwie podanej w przelaczniku -o.
- std::list< Item > findFittingItems (std::list< Item > &items, float backpackCapacity)

  Funkcja odrzucajaca przedmioty o wadze wiekszej niz pojemnosc plecaka - przygotowanie poczatkowych plecakow.
- Item drawItem (std::list< Item > &items)

  Funkcja losujaca przedmioty do plecakow initial.
- std::vector< Backpack > generateInitialBackpacks (std::list< Item > &items, int amountOfBackpacks, float backpackMaxWeight)

  Funkcja odpowiadajaca za przygotowanie wstepnych plecakow - populacja poczatkowa.
- Backpack mixBackpacks (Backpack &first, Backpack &second, float backpackCapacity)

  Funkcja odpowiadajaca za krzyzowanie plecakow.
- Backpack findBest (std::vector< Backpack > &generationToSearch)

  Funkcja odpowiadajaca za wybieranie najlepszego plecaka z poprzedniego pokolenia, szukanie najlepszych opcji.
- void createGeneration (std::vector< Backpack > &generationToMix, float backpackCapacity)

  Funkcja odpowiadajaca za tworzenie nowych plecakow.
- Backpack findSolution (std::vector< Backpack > &generation, int generations, float backpackCapacity)

  Funkcja odpowiadajaca za tworzenie nowych, coraz lepszych pokolen.

4.2.1 Function Documentation

4.2.1.1 createGeneration()

```
void createGeneration (
            std::vector< Backpack > & generationToMix,
            float backpackCapacity )
```

Funkcja odpowiadajaca za tworzenie nowych plecakow.

Parameters

generationToMix	Vector zawierajacy osobniki - plecaki do krzywowania.
backpackCapacity	Zmienna typu float okreslajaca pojemnosc plecaka.

4.2.1.2 drawItem()

```
Item drawItem (
            std::list< Item > & items )
```

Funkcja losujaca przedmioty do plecakow initial.

Parameters

items	Lista typu Items przechowujaca przedmioty znajdujace sie w plecaku.

Returns

zwraca losowo wybrany przedmiot z listy rzeczy pasujacych do plecaka.

4.2.1.3 findBest()

```
Backpack findBest (
            std::vector< Backpack > & generationToSearch )
```

Funkcja odpowiadajaca za wybieranie najlepszego plecaka z poprzedniego pokolenia, szukanie najlepszych opcji.

Parameters

Vector	zawierajacy przygotowane plecaki - osobniki.

Returns

zwraca plecak o najwiekszej wartosci - algorytm max.

4.2.1.4 findFittingItems()

```
std::list< Item > findFittingItems (
            std::list< Item > & items,
            float backpackCapacity )
```

Funkcja odrzucajaca przedmioty o wadze wiekszej niz pojemnosc plecaka - przygotowanie poczatkowych plecakow.

Parameters

items	Lista typu Items przechowujaca przedmioty znajdujace sie w plecaku.
backpackCapacity	Zmienna typu float okreslajaca pojemnosc plecaka.

Returns

zwraca liste przedmiotow pasujacych do plecaka

4.2.1.5 findSolution()

```
Backpack findSolution (
            std::vector< Backpack > & generation,
            int generations,
            float backpackCapacity )
```

Funkcja odpowiadajaca za tworzenie nowych, coraz lepszych pokolen.

Parameters

generation	Vector zawierajacy pokolenie plecakow do krzyzowania - pokolenie initial.
generations	Zmienna typu int okreslajaca ilosc pokolen.
backpackCapacity	Zmienna typu float okreslajaca pojemnosc plecaka.

Returns

zwraca najlepsze rozwiazanie dla uzyskanych generacji.

4.2.1.6 generateInitialBackpacks()

```
std::vector< Backpack > generateInitialBackpacks (
            std::list< Item > & items,
            int amountOfBackpacks,
            float backpackWeight )
```

Funkcja odpowiadajaca za przygotowanie wstepnych plecakow - populacja poczatkowa.

Parameters

items	Lista typu Items przechowujaca przedmioty znajdujace sie w plecaku.
amountOfBackpack	Zmienna typu int okreslajaca ilosc osobnikow w pokoleniu.
backpackWeight	Zmienna typu float okreslajaca wage przedmiotow w plecaku.

Returns

zwraca vector plecakow przygotowanych pod krzyzowanie.

4.2.1.7 Load_parameters()

```
bool Load_parameters (
            Params & param,
            const int & Liczba_parametrow,
            char * parametry[] )
```

Funkcja sprawujaca kontrole nad wlasciwym dostarczeniem przelacznikow do programu.

Parameters

param	Zmienna struktury pozwalajaca na poslugiwanie sie plikami i wartosciami, ktore byly dostarczone przez przelaczniki.
Liczba_parametrow	Zmienna przyjmujaca wartosc od argc - ilosc parametrow.
parametry	Zmienna przyjmujaca wartosc od argv - wartosci parametrow.

Returns

false blad przy czytaniu przelacznikow - niewlasciwy format / kolejnosc.

true brak bledow

4.2.1.8 mixBackpacks()

```
Backpack mixBackpacks (
            Backpack & first,
```

```
            Backpack & second,
            float backpackCapacity )
```

Funkcja odpowiadajaca za krzyzowanie plecakow.

Parameters

first	Zmienna struktury Backpack stworzona w celu porownywania 2 plecakww.
second	Zmienna struktury Backpack stworzona w celu porownywania 2 plecakow.
backpackCapacity	Zmienna typu float okreslajaca ilosc wolnego miejsca w plecaku temporary.

Returns

zwraca nowy plecak, otrzymany w wyniku krzyzowania 2 sasiednich.

4.2.1.9 move_items()

```
void move_items (
            std::list< Item > & items,
            std::ifstream & input )
```

Funkcja dostarczajaca dane z przelacznika - i do Listy typu Items w strukturze Backpack.

Parameters

items	Lista typu Items przechowujaca przedmioty znajdujace sie w plecaku.
input	Strumien ifstream pobierajacy dane z pliku z przelacznika -i.

4.2.1.10 to_file()

```
void to_file (
            Backpack & backpack,
            std::ofstream & output_o,
            Params & params )
```

Funkcja zapisujaca wynik programu do pliku o nazwie podanej w przelaczniku -o.

Parameters

backpack	Zmienna typu Backpack umozliwiajaca obsluge struktury Backpack.
output_o	Strumien typu ofstream zapsujacy zmienne do pliku.
params	Zmienna typu Param - zapis danych plecaka do pliku.

4.3 plecak.h File Reference

```
#include <iostream>
```

```
#include <vector>
#include <string>
#include <list>
```

Classes

- struct Params

  Struktura przyjmujaca wprowadzone dane - odpowiednie dla przelacznikow nazwy plikow i wartosci - do programu.
- struct Item

  struktura przechowujaca dane z pliku powiazanego z przelacznikiem -i.
- struct Backpack

  Struktura pelniaca role plecaka z przedmiotami.

Functions

- bool Load_parameters (Params &param, const int &Liczba_parametrow, char ∗parametry[ ])

  Funkcja sprawujaca kontrole nad wlasciwym dostarczeniem przelacznikow do programu.
- void move_items (std::list< Item > &items, std::ifstream &input)

  Funkcja dostarczajaca dane z przelacznika - i do Listy typu Items w strukturze Backpack.
- void to_file (Backpack &backpack, std::ofstream &output_o, Params &params)

  Funkcja zapisujaca wynik programu do pliku o nazwie podanej w przelaczniku -o.
- std::list< Item > findFittingItems (std::list< Item > &items, float backpackCapacity)

  Funkcja odrzucajaca przedmioty o wadze wiekszej niz pojemnosc plecaka - przygotowanie poczatkowych plecakow.
- Item drawItem (std::list< Item > &items)

  Funkcja losujaca przedmioty do plecakow initial.
- std::vector< Backpack > generateInitialBackpacks (std::list< Item > &items, int amountOfBackpacks, float backpackWeight)

  Funkcja odpowiadajaca za przygotowanie wstepnych plecakow - populacja poczatkowa.
- Backpack mixBackpacks (Backpack &first, Backpack &second, float backpackCapacity)

  Funkcja odpowiadajaca za krzyzowanie plecakow.
- Backpack findBest (std::vector< Backpack > &generationToSearch)

  Funkcja odpowiadajaca za wybieranie najlepszego plecaka z poprzedniego pokolenia, szukanie najlepszych opcji.
- void createGeneration (std::vector< Backpack > &generationToMix, float backpackCapacity)

  Funkcja odpowiadajaca za tworzenie nowych plecakow.
- Backpack findSolution (std::vector< Backpack > &generation, int generations, float backpackCapacity)

  Funkcja odpowiadajaca za tworzenie nowych, coraz lepszych pokolen.

4.3.1 Function Documentation

4.3.1.1 createGeneration()

```
void createGeneration (
            std::vector< Backpack > & generationToMix,
            float backpackCapacity )
```

Funkcja odpowiadajaca za tworzenie nowych plecakow.

Parameters

generationToMix	Vector zawierajacy osobniki - plecaki do krzywowania.
backpackCapacity	Zmienna typu float okreslajaca pojemnosc plecaka.

4.3.1.2 drawItem()

```
Item drawItem (
            std::list< Item > & items )
```

Funkcja losujaca przedmioty do plecakow initial.

Parameters

items	Lista typu Items przechowujaca przedmioty znajdujace sie w plecaku.

Returns

zwraca losowo wybrany przedmiot z listy rzeczy pasujacych do plecaka.

4.3.1.3 findBest()

```
Backpack findBest (
            std::vector< Backpack > & generationToSearch )
```

Funkcja odpowiadajaca za wybieranie najlepszego plecaka z poprzedniego pokolenia, szukanie najlepszych opcji.

Parameters

Vector	zawierajacy przygotowane plecaki - osobniki.

Returns

zwraca plecak o najwiekszej wartosci - algorytm max.

4.3.1.4 findFittingItems()

```
std::list< Item > findFittingItems (
            std::list< Item > & items,
            float backpackCapacity )
```

Funkcja odrzucajaca przedmioty o wadze wiekszej niz pojemnosc plecaka - przygotowanie poczatkowych plecakow.

Parameters

items	Lista typu Items przechowujaca przedmioty znajdujace sie w plecaku.
backpackCapacity	Zmienna typu float okreslajaca pojemnosc plecaka.

Returns

zwraca liste przedmiotow pasujacych do plecaka

4.3.1.5 findSolution()

```
Backpack findSolution (
            std::vector< Backpack > & generation,
            int generations,
            float backpackCapacity )
```

Funkcja odpowiadajaca za tworzenie nowych, coraz lepszych pokolen.

Parameters

generation	Vector zawierajacy pokolenie plecakow do krzyzowania - pokolenie initial.
generations	Zmienna typu int okreslajaca ilosc pokolen.
backpackCapacity	Zmienna typu float okreslajaca pojemnosc plecaka.

Returns

zwraca najlepsze rozwiazanie dla uzyskanych generacji.

4.3.1.6 generateInitialBackpacks()

```
std::vector< Backpack > generateInitialBackpacks (
            std::list< Item > & items,
            int amountOfBackpacks,
            float backpackWeight )
```

Funkcja odpowiadajaca za przygotowanie wstepnych plecakow - populacja poczatkowa.

Parameters

items	Lista typu Items przechowujaca przedmioty znajdujace sie w plecaku.
amountOfBackpack	Zmienna typu int okreslajaca ilosc osobnikow w pokoleniu.
backpackWeight	Zmienna typu float okreslajaca wage przedmiotow w plecaku.

Returns

zwraca vector plecakow przygotowanych pod krzyzowanie.

4.3.1.7 Load_parameters()

```
bool Load_parameters (
            Params & param,
            const int & Liczba_parametrow,
            char * parametry[] )
```

Funkcja sprawujaca kontrole nad wlasciwym dostarczeniem przelacznikow do programu.

Parameters

param	Zmienna struktury pozwalajaca na poslugiwanie sie plikami i wartosciami, ktore byly dostarczone przez przelaczniki.
Liczba_parametrow	Zmienna przyjmujaca wartosc od argc - ilosc parametrow.
parametry	Zmienna przyjmujaca wartosc od argv - wartosci parametrow.

Returns

false blad przy czytaniu przelacznikow - niewlasciwy format / kolejnosc.

true brak bledow

4.3.1.8 mixBackpacks()

```
Backpack mixBackpacks (
            Backpack & first,
            Backpack & second,
            float backpackCapacity )
```

Funkcja odpowiadajaca za krzyzowanie plecakow.

Parameters

first	Zmienna struktury Backpack stworzona w celu porownywania 2 plecakww.
second	Zmienna struktury Backpack stworzona w celu porownywania 2 plecakow.
backpackCapacity	Zmienna typu float okreslajaca ilosc wolnego miejsca w plecaku temporary.

Returns

zwraca nowy plecak, otrzymany w wyniku krzyzowania 2 sasiednich.

4.3.1.9 move_items()

```
void move_items (
            std::list< Item > & items,
            std::ifstream & input )
```

Funkcja dostarczajaca dane z przelacznika - i do Listy typu Items w strukturze Backpack.

Parameters

items	Lista typu Items przechowujaca przedmioty znajdujace sie w plecaku.
input	Strumien ifstream pobierajacy dane z pliku z przelacznika -i.

4.3.1.10 to_file()

```
void to_file (
```

```
            Backpack & backpack,
            std::ofstream & output_o,
            Params & params )
```

Funkcja zapisujaca wynik programu do pliku o nazwie podanej w przelaczniku -o.

Parameters

backpack	Zmienna typu Backpack umozliwiajaca obsluge struktury Backpack.
output_o	Strumien typu ofstream zapsujacy zmienne do pliku.
params	Zmienna typu Param - zapis danych plecaka do pliku.

4.4 plecak.h

Go to the documentation of this file.

```
00001 #ifndef PLECAK_H
00002 #define PLECAK_H
00003
00004 #include <iostream>
00005 #include <vector>
00006 #include<string>
00007 #include <list>
00008
00017 struct Params
00018 {
00019     std::string itemsFile;
00020     std::string solutionFile;
00021     float capacity;
00022     int generations;
00023     int amountOfSpecimens;
00024 };
00025
00032 struct Item
00033 {
00034     std::string name = "";
00035     float weight = 0;
00036     float price = 0;
00037 };
00038
00039
00046 struct Backpack
00047 {
00048     std::list<Item> items = std::list<Item>();
00049     float capacity = 0;
00050     int value = 0;
00051 };
00052
00053
00062 bool Load_parameters(Params& param, const int& Liczba_parametrow, char* parametry[]);
00063
00064
00070 void move_items(std::list<Item>& items, std::ifstream& input);
00071
00072
00079 void to_file(Backpack& backpack, std::ofstream& output_o, Params& params);
00080
00081
00088 std::list<Item> findFittingItems(std::list<Item>& items, float backpackCapacity);
00089
00090
00096 Item drawItem(std::list<Item>& items);
00097
00098
00106 std::vector<Backpack> generateInitialBackpacks(std::list<Item>& items, int amountOfBackpacks, float
      backpackWeight);
00107
00108
00116 Backpack mixBackpacks(Backpack& first, Backpack& second, float backpackCapacity);
00117
00118
00124 Backpack findBest(std::vector<Backpack>& generationToSearch);
00125
```

```
00126
00132 void createGeneration(std::vector<Backpack>& generationToMix, float backpackCapacity);
00133
00134
00142 Backpack findSolution(std::vector<Backpack>& generation, int generations, float backpackCapacity);
00143
00144
00145 #endif  // !PLECAK_H
```

Index