My Project

Generated by Doxygen 1.9.1

Chapter 1

README

Place your project here

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

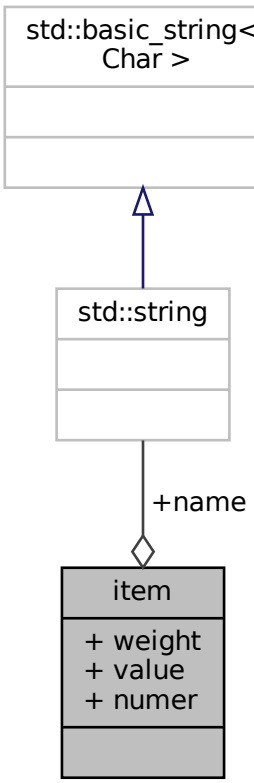
Class Documentation

4.1 item Struct Reference

Represents an item in the backpack problem.

`#include <structures.h>`

Collaboration diagram for item:



Public Attributes

- std::string **name**
- double **weight**
- double **value**
- int **numer**

4.1.1 Detailed Description

Represents an item in the backpack problem.

The item structure contains the following properties:

Parameters

name	Name of the item
weight	Weight of the item
value	Value of the item
numer	Number of the item

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

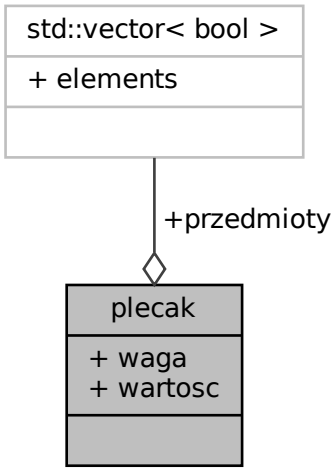
- structures.h

4.2 plecak Struct Reference

Represents a backpack in the backpack problem.

`#include <structures.h>`

Collaboration diagram for plecak:



Public Attributes

- std::vector< bool > **przedmioty**
- double **waga**
- double **wartosc**

4.2.1 Detailed Description

Represents a backpack in the backpack problem.

The plecak structure contains the following properties:

Parameters

przedmioty	Vector of booleans representing whether an item is in the backpack or not
waga	Total weight of the items in the backpack
wartosc	Total value of the items in the backpack

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

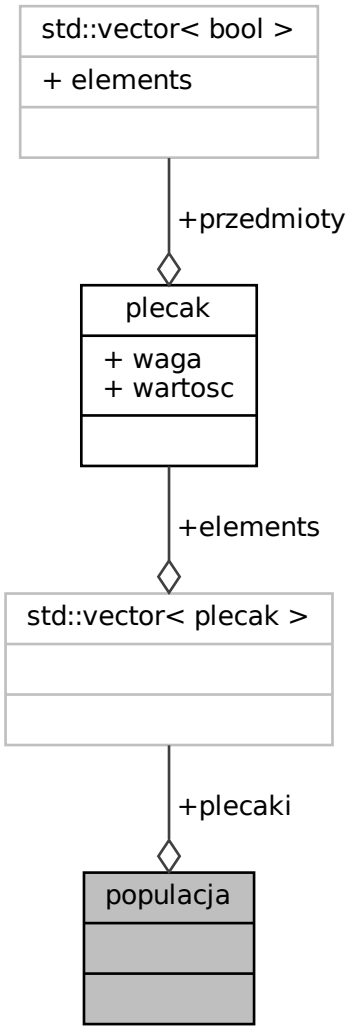
- structures.h

4.3 populacja Struct Reference

Represents a population in the backpack problem.

```
#include <structures.h>
```

Collaboration diagram for populacja:



Public Attributes

- std::vector< plecak > **plecaki**

4.3.1 Detailed Description

Represents a population in the backpack problem.

The populacja structure contains the following properties:

Parameters

plecaki	Vector of backpacks in the population

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

- structures.h

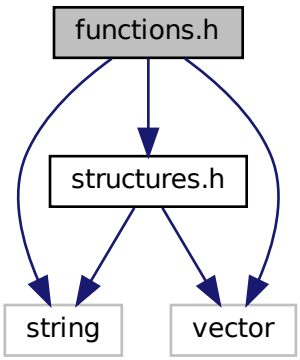
Chapter 5

File Documentation

5.1 functions.h File Reference

```
#include <string>
#include <vector>
#include "structures.h"
```

Include dependency graph for functions.h:



Functions

- std::map< std::string, std::string > paramcheck (int ile, char *param[])

  Parameter checker.
- std::deque< item > dataread (const std::string &filename)

  Data reader.
- int randomint (const int &limit)

  Random number generator.
- plecak generate_knapsack (const std::deque< item > &items, const double &capacity)

  Knapsack generator.

- populacja generatefirstgen (const std::deque< item > &items, const double &count, const double &capacity)

  First generation generator.
- populacja VWcalcualator (const std::deque< item > &items, populacja &generation)

  Value-Weight calculator.
- populacja crossing (const std::deque< item > &items, const populacja &generation)

  Crossing operation.
- populacja selection (const populacja &generation, const double &count, const double &capacity)

  Select operation.
- plecak topplecak (const populacja &generation, const double &capacity)

  Best solution calculator.
- void datawrite (const populacja &generation, const double &gen, const std::deque< item > &items, const double &capacity, std::ofstream &file)

  Data output operation.
- void genetic (const std::deque< item > &items, const double &count, const double &capacity, const double &gennumber, const std::string &output)

  Man genetic function.

5.1.1 Function Documentation

5.1.1.1 crossing()

```
populacja crossing (
            const std::deque< item > & items,
            const populacja & generation )
```

Crossing operation.

Creates new knapsacks by choosing and merging two individuals taken from generation.

Parameters

generation	The current population of knapsacks
items	The set of items

Returns

New population based on previus one

5.1.1.2 dataread()

```
std::deque<item> dataread (
            const std::string & filename )
```

Data reader.

This function reads and checks the data from a file and stores it in a deque of struct items.

Parameters

filename	The name of the file containing the data

Returns

A deque of struct items

5.1.1.3 datawrite()

```
void datawrite (
            const populacja & generation,
            const double & gen,
            const std::deque< item > & items,
            const double & capacity,
            std::ofstream & file )
```

Data output operation.

Writes the best solution of a generation to the text file with the information about items, total value and weight of an knapsack.

Parameters

generation	The current population of knapsacks
output	The name of the output file
gen	The current generation number
items	The set of items
capacity	The capacity of each knapsack
file	The output file stream

5.1.1.4 generate_knapsack()

```
plecak generate_knapsack (
            const std::deque< item > & items,
            const double & capacity )
```

Knapsack generator.

This function generates a struct variable plecak given a set of items and a capacity.

Parameters

items	The set of items
capacity	The capacity of the knapsack

Returns

A knapsack

5.1.1.5 generatefirstgen()

```
populacja generatefirstgen (
            const std::deque< item > & items,
            const double & count,
            const double & capacity )
```

First generation generator.

This function generates the first generation of knapsacks which is made out of structure populacja.

Parameters

items	The set of items
count	The number of knapsacks in the generation
capacity	The capacity of each knapsack

Returns

The first generation of knapsacks

5.1.1.6 genetic()

```
void genetic (
            const std::deque< item > & items,
            const double & count,
            const double & capacity,
            const double & gennumber,
            const std::string & output )
```

Man genetic function.

Generates best solution for knapsack problem by combining other sub-functions.

Parameters

items	The set of items
count	The number of knapsacks in the generation
capacity	The capacity of each knapsack
gennumber	The number of generations to run the algorithm for
output	The name of the output file

5.1.1.7 paramcheck()

```
std::map<std::string, std::string> paramcheck (
            int ile,
            char * param[] )
```

Parameter checker.

Checks the parameters passed to the program and stores them in a map, where the keys are the names of the parameters and the map values are their values.

Parameters

ile	The number of parameters passed to the program
param	The array of parameters passed to the program

Returns

A map containing the parameters and their values

5.1.1.8 randomint()

```
int randomint (
            const int & limit )
```

Random number generator.

This function generates a random integer in the range [0, a].

Parameters

limit	The upper bound of the range

Returns

A random integer

5.1.1.9 selection()

```
populacja selection (
            const populacja & generation,
```

```
            const double & count,
            const double & capacity )
```

Select operation.

Selects the best individuals from the current generation based on their fitness value.

Parameters

generation	The current population of knapsacks
capacity	The capacity of each knapsack

Returns

New population based on previus one

5.1.1.10 topplecak()

```
plecak topplecak (
            const populacja & generation,
            const double & capacity )
```

Best solution calculator.

Finds best knapsack by comparing all candidates in generation.

Parameters

generation	The current population of knapsacks
capacity	The capacity of each knapsack

Returns

The best solution

5.1.1.11 VWcalcualator()

```
populacja VWcalcualator (
            const std::deque< item > & items,
            populacja & generation )
```

Value-Weight calculator.

This function calculates the value and weight of each knapsack in a generation.

Parameters

items	The set of items
generation	The current population of knapsacks

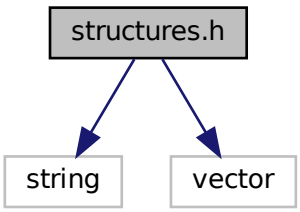
Returns

The updated generation of knapsacks with their values and weights calculated

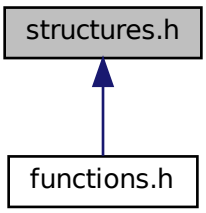
5.2 structures.h File Reference

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

Include dependency graph for structures.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct item

  Represents an item in the backpack problem.
- struct plecak

  Represents a backpack in the backpack problem.
- struct populacja

  Represents a population in the backpack problem.

Index