

Organizer

Generated by Doxygen 1.9.1

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 Class Documentation	5
3.1 Organizer Class Reference	5
3.1.1 Detailed Description	6
3.1.2 Constructor & Destructor Documentation	6
3.1.2.1 Organizer() [1/2]	6
3.1.2.2 Organizer() [2/2]	7
3.1.3 Member Function Documentation	7
3.1.3.1 printAll()	7
3.1.3.2 printListToFile()	7
3.1.3.3 addTask()	7
3.1.3.4 updateTask()	8
3.1.3.5 removeTask()	8
3.1.3.6 removeFinished()	8
3.1.3.7 printHelp()	8
3.1.3.8 removeAll()	8
3.1.3.9 countTasks()	9
3.1.3.10 getName()	9
3.1.3.11 setName()	9
3.1.3.12 interactiveMode()	9
3.2 Product Class Reference	10
3.2.1 Detailed Description	10
3.3 ShoppingList Class Reference	11
3.3.1 Detailed Description	11
3.3.2 Member Function Documentation	12
3.3.2.1 printListToFile()	12
3.3.2.2 addTask()	12
3.3.2.3 printAll()	12
3.3.2.4 removeTask()	12
3.3.2.5 removeFinished()	13
3.3.2.6 updateTask()	13
3.3.2.7 removeAll()	13
3.3.2.8 countTasks()	13
3.4 Task Class Reference	14
3.4.1 Detailed Description	14
3.4.2 Member Function Documentation	15
3.4.2.1 set() [1/2]	15
3.4.2.2 set() [2/2]	15

3.4.3 Friends And Related Function Documentation	15
3.4.3.1 operator<<	15
Index	17

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Organizer	5
ShoppingList	11
Task	14
Product	10

Chapter 2

Class Index

2.1 Class List

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

Organizer	Base class used as a list of tasks to do	5
Product	Derived class from class Task that becomes a product for purchase. Methods and overloaded operator work the same as in base class but are slightly changed	10
ShoppingList	Derived class from class Organizer that is a shopping list. Instead of vector of objects of type Task , it holds objects of type Product . Its methods work the same as in class Organizer but were changed to work with Products and not Tasks	11
Task	Base class that represents a task	14

Chapter 3

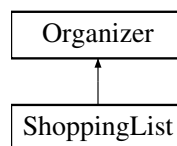
Class Documentation

3.1 Organizer Class Reference

Base class used as a list of tasks to do.

```
#include <organizer.h>
```

Inheritance diagram for Organizer:



Public Member Functions

- virtual void `removeAll ()`
A method for removing all tasks.
- virtual int `countTasks ()`
A method for calculating number of tasks in the list.
- std::string `getName ()`
A method for getting a name of the list.
- void `setName ()`
A method that sets a new name for the list.
- void `interactiveMode ()`
A method that calls other methods depending on user's input.
- `Organizer (std::string_view _listName)`
One-argument constructor.
- `Organizer ()`
Default constructor.

Protected Member Functions

- virtual void `printAll ()`
A method for printing the list.
- virtual void `printListToFile ()`
A method for printing the list to file.
- virtual void `addTask ()`
A method for pushing a new task on the list.
- virtual void `updateTask ()`
A method for changing task's status.
- virtual void `removeTask ()`
A method for removing a specific task from the list.
- virtual void `removeFinished ()`
A method for removing finished tasks.
- void `printHelp ()`
A method that prints available options for the user.

Protected Attributes

- `std::string listName`

Private Attributes

- `std::vector< Task > list`
- `Task _task`

3.1.1 Detailed Description

Base class used as a list of tasks to do.

Parameters

<i>list</i>	A vector of objects of type <code>Task</code>
<i>listName</i>	Name of the list
<i>_task</i>	An object that is being modified with its copy being pushed on the list

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Organizer() [1/2]

```
Organizer::Organizer (
    std::string_view _listName )
```

One-argument constructor.

Parameters

<code>_listName</code>	A name for the list
------------------------	---------------------

3.1.2.2 Organizer() [2/2]

```
Organizer::Organizer ( )
```

Default constructor.

3.1.3 Member Function Documentation

3.1.3.1 printAll()

```
void Organizer::printAll ( ) [protected], [virtual]
```

A method for printing the list.

Reimplemented in [ShoppingList](#).

3.1.3.2 printListToFile()

```
void Organizer::printListToFile ( ) [protected], [virtual]
```

A method for printing the list to file.

Reimplemented in [ShoppingList](#).

3.1.3.3 addTask()

```
void Organizer::addTask ( ) [protected], [virtual]
```

A method for pushing a new task on the list.

Reimplemented in [ShoppingList](#).

3.1.3.4 updateTask()

```
void Organizer::updateTask ( ) [protected], [virtual]
```

A method for changing task's status.

Reimplemented in [ShoppingList](#).

3.1.3.5 removeTask()

```
void Organizer::removeTask ( ) [protected], [virtual]
```

A method for removing a specific task from the list.

Reimplemented in [ShoppingList](#).

3.1.3.6 removeFinished()

```
void Organizer::removeFinished ( ) [protected], [virtual]
```

A method for removing finished tasks.

Reimplemented in [ShoppingList](#).

3.1.3.7 printHelp()

```
void Organizer::printHelp ( ) [protected]
```

A method that prints available options for the user.

3.1.3.8 removeAll()

```
void Organizer::removeAll ( ) [virtual]
```

A method for removing all tasks.

Reimplemented in [ShoppingList](#).

3.1.3.9 countTasks()

```
int Organizer::countTasks ( ) [virtual]
```

A method for calculating number of tasks in the list.

Returns

Number of tasks

Reimplemented in [ShoppingList](#).

3.1.3.10 getName()

```
std::string Organizer::getName ( )
```

A method for getting a name of the list.

Returns

Name of the list

3.1.3.11 setName()

```
void Organizer::setName ( )
```

A method that sets a new name for the list.

3.1.3.12 interactiveMode()

```
void Organizer::interactiveMode ( )
```

A method that calls other methods depending on user's input.

The documentation for this class was generated from the following files:

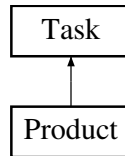
- organizer.h
- organizer.cpp

3.2 Product Class Reference

Derived class from class [Task](#) that becomes a product for purchase. Methods and overloaded operator work the same as in base class but are slightly changed.

```
#include <task.h>
```

Inheritance diagram for Product:



Public Member Functions

- void **print** ()
- void **set** (std::string_view _name, double _cost, bool _status)
- void **set** (std::string_view _name, double _cost)
- double **getCost** ()

Private Attributes

- double **cost**

Friends

- std::ostream & **operator**<< (std::ostream &os, [Product](#) const &prod)

Additional Inherited Members

3.2.1 Detailed Description

Derived class from class [Task](#) that becomes a product for purchase. Methods and overloaded operator work the same as in base class but are slightly changed.

Parameters

<i>cost</i>	A cost of the product
-------------	-----------------------

The documentation for this class was generated from the following files:

- task.h
- task.cpp

3.3 ShoppingList Class Reference

Derived class from class [Organizer](#) that is a shopping list. Instead of vector of objects of type [Task](#), it holds objects of type [Product](#). Its methods work the same as in class [Organizer](#) but were changed to work with Products and not Tasks.

```
#include <organizer.h>
```

Inheritance diagram for ShoppingList:



Public Member Functions

- **ShoppingList** (std::string_view _listName)
- void [removeAll](#) ()
A method for removing all tasks.
- int [countTasks](#) ()
A method for calculating number of tasks in the list.

Private Member Functions

- void [printListToFile](#) ()
A method for printing the list to file.
- void [addTask](#) ()
A method for pushing a new task on the list.
- void [printAll](#) ()
A method for printing the list.
- void [removeTask](#) ()
A method for removing a specific task from the list.
- void [removeFinished](#) ()
A method for removing finished tasks.
- void [updateTask](#) ()
A method for changing task's status.

Private Attributes

- [Product](#) _product
- std::vector< [Product](#) > list

Additional Inherited Members

3.3.1 Detailed Description

Derived class from class [Organizer](#) that is a shopping list. Instead of vector of objects of type [Task](#), it holds objects of type [Product](#). Its methods work the same as in class [Organizer](#) but were changed to work with Products and not Tasks.

Parameters

<i>list</i>	A vector of objects of type Product
<i>_product</i>	An object that is being modified with its copy being pushed on the list

3.3.2 Member Function Documentation

3.3.2.1 printListToFile()

```
void ShoppingList::printListToFile ( ) [private], [virtual]
```

A method for printing the list to file.

Reimplemented from [Organizer](#).

3.3.2.2 addTask()

```
void ShoppingList::addTask ( ) [private], [virtual]
```

A method for pushing a new task on the list.

Reimplemented from [Organizer](#).

3.3.2.3 printAll()

```
void ShoppingList::printAll ( ) [private], [virtual]
```

A method for printing the list.

Reimplemented from [Organizer](#).

3.3.2.4 removeTask()

```
void ShoppingList::removeTask ( ) [private], [virtual]
```

A method for removing a specific task from the list.

Reimplemented from [Organizer](#).

3.3.2.5 removeFinished()

```
void ShoppingList::removeFinished ( ) [private], [virtual]
```

A method for removing finished tasks.

Reimplemented from [Organizer](#).

3.3.2.6 updateTask()

```
void ShoppingList::updateTask ( ) [private], [virtual]
```

A method for changing task's status.

Reimplemented from [Organizer](#).

3.3.2.7 removeAll()

```
void ShoppingList::removeAll ( ) [virtual]
```

A method for removing all tasks.

Reimplemented from [Organizer](#).

3.3.2.8 countTasks()

```
int ShoppingList::countTasks ( ) [virtual]
```

A method for calculating number of tasks in the list.

Returns

Number of tasks

Reimplemented from [Organizer](#).

The documentation for this class was generated from the following files:

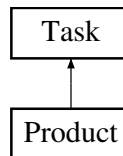
- organizer.h
- organizer.cpp

3.4 Task Class Reference

Base class that represents a task.

```
#include <task.h>
```

Inheritance diagram for Task:



Public Member Functions

- virtual void **print** ()
- virtual void **set** (std::string_view _name, std::string_view _desc, bool _status)
A three-argument method that changes object's values.
- virtual void **set** (std::string_view _name, std::string_view _desc)
An overloaded two-argument method that changes object's values. Works the same as its original version, but sets status value as false by default.

Public Attributes

- std::string **name**
- bool **status**

Private Attributes

- std::string **desc**

Friends

- std::ostream & **operator<<** (std::ostream &os, **Task** const &task)
An overloaded operator for outputting object's values.

3.4.1 Detailed Description

Base class that represents a task.

Parameters

<i>desc</i>	A description of the task
<i>name</i>	A name of the task
<i>status</i>	A status of the task represented by a boolean value

3.4.2 Member Function Documentation

3.4.2.1 set() [1/2]

```
void Task::set (
    std::string_view _name,
    std::string_view _desc,
    bool _status ) [virtual]
```

A three-argument method that changes object's values.

Parameters

<code>_name</code>	The new name for the task
<code>_desc</code>	The new description for the task
<code>_status</code>	The new status of the task

3.4.2.2 set() [2/2]

```
void Task::set (
    std::string_view _name,
    std::string_view _desc ) [virtual]
```

An overloaded two-argument method that changes object's values. Works the same as its original version, but sets status value as false by default.

3.4.3 Friends And Related Function Documentation

3.4.3.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    Task const & task ) [friend]
```

An overloaded operator for outputting object's values.

Returns

A string that is ready for being put either on screen or to file

The documentation for this class was generated from the following files:

- task.h
- task.cpp

Index

addTask
 Organizer, [7](#)
 ShoppingList, [12](#)

countTasks
 Organizer, [8](#)
 ShoppingList, [13](#)

getName
 Organizer, [9](#)

interactiveMode
 Organizer, [9](#)

operator<<
 Task, [15](#)
Organizer, [5](#)
 addTask, [7](#)
 countTasks, [8](#)
 getName, [9](#)
 interactiveMode, [9](#)
 Organizer, [6](#), [7](#)
 printAll, [7](#)
 printHelp, [8](#)
 printListToFile, [7](#)
 removeAll, [8](#)
 removeFinished, [8](#)
 removeTask, [8](#)
 setName, [9](#)
 updateTask, [7](#)

printAll
 Organizer, [7](#)
 ShoppingList, [12](#)

printHelp
 Organizer, [8](#)

printListToFile
 Organizer, [7](#)
 ShoppingList, [12](#)

Product, [10](#)

removeAll
 Organizer, [8](#)
 ShoppingList, [13](#)

removeFinished
 Organizer, [8](#)
 ShoppingList, [12](#)

removeTask
 Organizer, [8](#)
 ShoppingList, [12](#)

set
 Task, [15](#)
setName
 Organizer, [9](#)
ShoppingList, [11](#)
 addTask, [12](#)
 countTasks, [13](#)
 printAll, [12](#)
 printListToFile, [12](#)
 removeAll, [13](#)
 removeFinished, [12](#)
 removeTask, [12](#)
 updateTask, [13](#)

Task, [14](#)
 operator<<, [15](#)
 set, [15](#)

updateTask
 Organizer, [7](#)
 ShoppingList, [13](#)