labo_07_schaufelberger_yannick_gallay_david

Generated by Doxygen 1.8.13

Contents

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

Chapter 2

File Documentation

2.1 constants.h File Reference

Enumerations

- enum MONTHS {
  JANUAR = 1, FEBRUAR, MARCH, APRIL,
  MAY, JUNE, JULY, AUGUST,
  SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER }

Variables

- const int MIN_DAY = 1
- const int MIN_MONTH = JANUAR
- const int MIN_YEAR = 1900
- const int MAX_DAY = 31
- const int MAX_MONTH = DECEMBER
- const int MAX_YEAR = 2300
- const char DATE_SEPARATOR = '-'

2.1.1 Enumeration Type Documentation

2.1.1.1 MONTHS

```
enum MONTHS
```

Enumerator

JANUAR	
FEBRUAR	
MARCH	
APRIL	
MAY	

Enumerator

JUNE	
JULY	
AUGUST	
SEPTEMBER	
OCTOBER	
NOVEMBER	
DECEMBER	

2.1.2 Variable Documentation

2.1.2.1 DATE_SEPARATOR

```
const char DATE_SEPARATOR = '-'
```

2.1.2.2 MAX_DAY

```
const int MAX_DAY = 31
```

2.1.2.3 MAX_MONTH

```
const int MAX_MONTH = DECEMBER
```

2.1.2.4 MAX_YEAR

```
const int MAX_YEAR = 2300
```

2.1.2.5 MIN_DAY

```
const int MIN_DAY = 1
```

2.1.2.6 MIN_MONTH

```
const int MIN_MONTH = JANUAR
```

2.1.2.7 MIN_YEAR

```
const int MIN_YEAR = 1900
```

2.2 delta_date.h File Reference

```
#include <string>
```

Include dependency graph for delta_date.h:

2.3 interface.h File Reference

Functions

- bool ask_for_restart ()
  This function keeps asking as long as the user enters anything else than RESTART_CHAR or STOP_CHAR.

Variables

- const char RESTART_CHAR = 'O'
- const char STOP_CHAR = 'N'

2.3.1 Function Documentation

2.3.1.1 ask_for_restart()

```
bool ask_for_restart ( )
```

This function keeps asking as long as the user enters anything else than RESTART_CHAR or STOP_CHAR.

Returns

true if the user has entered RESTART_CHAR and false if the user has entered STOP_CHAR

2.3.2 Variable Documentation

2.3.2.1 RESTART_CHAR

```
const char RESTART_CHAR = 'O'
```

2.3.2.2 STOP_CHAR

```
const char STOP_CHAR = 'N'
```

2.4 utilities.h File Reference

```
#include <string>
#include <iostream>
#include <limits>
```

Include dependency graph for utilities.h:

Macros

- #define CLEAR_BUFFER std::cin.ignore(std::numeric_limits<std::streamsize>::max(),'\n')

Functions

- bool is_leap_year (int year)
- bool is_date_valid (int day, int month, int year)

  This function checks that year is between a range of date given by constants, that month is between 1 and 12 and that day is between 1 and 28/29/30/31, depending on the month.
- bool check_date_order (int start_day, int start_month, int start_year, int end_day, int end_month, int end_year)
- int days_between_dates (int startDay, int startMonth, int startYear, int endDay, int endMonth, int endYear)

  This returns a negative number if the start date is after the end date.
- int get_days_since_reference_day (int day, int month, int year)

  This function is heavily inspired by the Julian Day calculation found on this page : `http://www.cs.utsa.edu/~cs1063/projects/Spring2011/Project1/jdn-explanation.html` *It has been slightly modified to fit the current project.*

2.4.1 Macro Definition Documentation

2.4.1.1 CLEAR_BUFFER

```
#define CLEAR_BUFFER std::cin.ignore(std::numeric_limits<std::streamsize>::max(),'\n')
```

2.4.2 Function Documentation

2.4.2.1 check_date_order()

```
bool check_date_order (
             int start_day,
             int start_month,
             int start_year,
             int end_day,
             int end_month,
             int end_year )
```

Parameters

start_day	a number between 1 and 31
start_month	a number between 1 and 12
start_year	a number between 1900 and 2300
end_day	a number between 1 and 31
end_month	a number between 1 and 12
end_year	a number between 1900 and 2300

Returns

true if the date composed of start_day, start_month and start_year is before the date composed of end_day, end_month and end_year

2.4.2.2 days_between_dates()

```
int days_between_dates (
             int startDay,
             int startMonth,
             int startYear,
             int endDay,
             int endMonth,
             int endYear )
```

This returns a negative number if the start date is after the end date.

Parameters

startDay	
startMonth	
startYear	
endDay	
endMonth	
endYear	

Returns

the number of days between the start date and the end date

2.4.2.3 get_days_since_reference_day()

```
int get_days_since_reference_day (
            int day,
            int month,
            int year )
```

This function is heavily inspired by the Julian Day calculation found on this page : http://www.cs.utsa.edu/~cs1063/projects/Spring2011/Project1/jdn-explanation.html It has been slightly modified to fit the current project.

Parameters

day	
month	
year	

Returns

the number of days between January 1st 1900 and the date defined by day, month and year

2.4.2.4 is_date_valid()

```
bool is_date_valid (
            int day,
            int month,
            int year )
```

This function checks that year is between a range of date given by constants, that month is between 1 and 12 and that day is between 1 and 28/29/30/31, depending on the month.

Parameters

day	any number
month	any number
year	any number

Returns

true if the date composed of day, month and year is a valid date

2.4.2.5 is_leap_year()

```
bool is_leap_year (
            int year )
```

Parameters

year	any given year

Returns

true if year is a leap year, else false

Index