#pragma once

#include <string>

namespace ext

{

enum Month

{

January = 1,

February,

March,

April,

May,

June,

July,

August,

September,

October,

Novemver,

December

};

enum DateFormat

{

MonthAsInt,

MonthAsString

};

enum Season

{

Winter,

Spring,

Summer,

Autumn

};

struct Date

{

int year;

Month month;

int day;

};

/\*

Возвращает Юлианскую дату

https://ru.wikipedia.org/wiki/%D0%AE%D0%BB%D0%B8%D0%B0%D0%BD%D1%81%D0%BA%D0%B0%D1%8F\_%D0%B4%D0%B0%D1%82%D0%B0

раздел "Вычисление номера юлианского дня (JDN) по дате григорианского календаря"

Тестовые данные Ожидаемый результат

1.12.2018 2458454

1.1.2018 2458120

1.6.2000 2451697

21.12.2012 2456283

\*/

unsigned int countJND(Date date)

{

int a;

a = (14 - date.month) / 12;

int y;

y = date.year + 4800 - a;

int m;

m = date.month + 12 \* a - 3;

return date.day + ((153 \* m + 2) / 5) + (365 \* y) + (y / 4) - (y / 100) + (y / 400) - 32045;

}

/\*

Рассчитывает количество дней между двумя датами.

При реализвации используйте CountJND

\*/

int countDistance(Date from, Date to)

{

return countJND(from) - countJND(to);

}

/\*

Выводит в консоль дату в формате: dd.mm.yyyy (2.10.2001)

Если параметр DateFormat равен MonthAsString, то вывести дату в формате: dd monthName yyyy (2 октября 2001)

\*/

void print(Date data, DateFormat format = DateFormat::MonthAsInt);

/\*

Возвращает сезон (зима, весна, лето, осень) передаваемой даты

\*/

Season getSeason(Date date);

/\*

Сравнивает даты на эквивалентность

\*/

bool isSame(Date lhs, Date rhs);

/\*

Меняет даты местами

\*/

void swap(Date \* lhs, Date \* rhs);

/\*

Возвращает указатель на большую дату

\*/

Date \* max(Date \* lhs, Date \* rsh);

/\*

Возвращает указатель на меньшую дату

\*/

Date \* min(Date \* lhs, Date \* rsh);

/\*

Возвращает указатель на минимальную дату в переданном массиве дат

\*/

Date \* getMinDate(Date dates[], int size);

/\*

Возвращает указатель на максимальную дату в переданном массиве дат

\*/

Date \* getMaxDate(Date dates[], int size);

/\*

Проверяет может ли существовать дата с передаваемыми значениями.

В случае успеха возвращает true, и заполняет дату по переданному адресу.

Если дата с переданными параметрами существовать не может,

возвращает false.

\*/

bool tryFillDate(Date \* date, int year, int month, int day);

}