|  |  |
| --- | --- |
| Министерство образования Республики Беларусь | |
| Учреждение образования | |
| БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ | |
| ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ | |
|  | |
|  | |
| Факультет компьютерных систем и сетей | |
| Кафедра программного обеспечения информационных технологий | |
| Дисциплина: **Название дисциплины (АББРЕВИАТУРА)** | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
| **ОТЧЁТ** | |
| по лабораторной работе № **X** | |
|  | |
| Тема работы: **Название темы** | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
| Выполнил: | Фамилия И.О. |
|  | гр. **XXXXXX** |
|  | Вариант **X** |
|  |  |
| Проверил: | Фамилия И.О. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Минск **202X** | |

содержание

[1 Постановка задачи 3](#_Toc91701377)

[2 Методика решения 4](#_Toc91701378)

[2.1 Пользовательские типы данных 4](#_Toc91701379)

[2.2 Использование условия високосности 4](#_Toc91701380)

[2.3 Использование массивов 4](#_Toc91701381)

[3 Текстовый алгоритм решения задачи 5](#_Toc91701382)

[4 Структура данных 7](#_Toc91701383)

[5 Схема алгоритма решения задачи по ГОСТ 19.701-90 8](#_Toc91701384)

[6 Результаты расчетов 10](#_Toc91701385)

[Приложение А 11](#_Toc91701386)

[Приложение Б 14](#_Toc91701387)

# Постановка задачи

Даны: число, месяц, год. На какой день недели приходится эта дата? («Вечный календарь»);

При выполнении работы считать, что Григорианский календарь был введён с 1 января 1 года.

# Методика решения

## Пользовательские типы данных

Для облегчения работы можно использовать два типа данных: месяц и день недели, которые принимают 12 и 7 значений соответственно.

## Использование условия високосности

Чтобы определить день недели по дате, нужно вычислить, сколько всего прошло дней, начиная с 1 января 1 года.

Общее количество дней можно найти, если:

1. Взять количество дней в последнем месяце;
2. Добавить количество дней в уже прошедших месяцах, начиная с 1 января и заканчивая текущим месяцем невключительно;
3. Добавить количество дней в уже прошедших годах, начиная с 1 года и заканчивая текущим годом невключительно;
4. Добавить количество високосных дней, равное количеству високосных лет среди уже прошедших годов, начиная с 1 года и и заканчивая текущим годом невключительно;
5. Если текущий год – високосный, а введённая дата – 1 марта или позже, то добавить один високосный день текущего года.

В результате можно получить формулу:

*N:= N + (Year - 1) \* 365 + ((Year-1) Div 4) - ((Year-1) Div 100) +*

*+ ((Year-1) Div 400) + DaysBefore[NMon]*

Полученное количество необходимо разделить с остатком на 7, полученное число будет номером дня в неделе. Номером воскресенья будет 0, т.к. остаток при делении на 7 не может быть равен 7.

## Использование массивов

Количество дней в месяцах постоянно, если 29 февраля добавляется отдельно. Названия дней недели также постоянны. Поэтому можно создать 2 массива: первый для хранения количества дней, прошедших от начала года, а второй – для хранения названий дней недели.

Значение прошедших от начала года дней находится по индексу массива, равному одному из значений перечисления месяцев в пользовательском типе данных

Вывод дня недели осуществляется по индексу массива, равному одному из значений перечисления дней недели в пользовательском типе данных.

# Текстовый алгоритм решения задачи

Таблица – Алгоритм решения

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | Ввод Year, Month |
|  | Flag:= (((Year Mod 4 = 0) And (Year Mod 100 <> 0)) Or (Year Mod 400 = 0)) |
|  | NMon:= Jan |
|  | I:= 1 |
|  | Начало цикла А1. Проверка выполнения условия (I<=11). Если условие истинно, идти к шагу 6, иначе – к шагу 10 |
|  | Проверка выполнения условия Month > I. Если условие истинно, перейти к шагу 7, иначе – к шагу 8 |
|  | Inc(NMon) |
|  | Inc(I) |
|  | Конец цикла А1. Идти к шагу 5 |
|  | Проверка выполнения условия NMon = Feb. Если условие истинно, перейти к шагу 11, иначе – к шагу 15 |
|  | Max:=28 |
|  | Проверка выполнения условия Flag = True. Если условие истинно, перейти к шагу 13, иначе – к шагу 14 |
|  | Inc(Max) |
|  | Идти к шагу 19 |
|  | Проверка выполнения условия (NMon = Jun) or (NMon = Sep) or (NMon = Nov) or (NMon = Apr). Если условие истинно, перейти к шагу 16, иначе – к шагу 18 |
|  | Max:=30 |
|  | Идти к шагу 19 |
|  | Max:=31 |
|  | Начало цикла А2. Идти к шагу 20 |
|  | Ввод N |
|  | Конец цикла А2. Проверка выполнения условия (N>0) And (N<=Max). Если условие истинно, идти к шагу 22, иначе – к шагу 19 |
|  | N:= N + (Year - 1) \* 365 + ((Year-1) Div 4) - ((Year-1) Div 100) + ((Year-1) Div 400) + AQuas[NMon] |
|  | Проверка выполнения условия ((Flag = True) And (NMod > Feb)). Если условие истинно, перейти к шагу 24, иначе – к шагу 25 |
|  | Inc(N) |
|  | NWds:=Sun |

Продолжение Таблицы 1

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | I:=0 |
|  | Начало цикла А3. Проверка выполнения условия (I<=11). Если условие истинно, идти к шагу 28, иначе – к шагу 32 |
|  | Проверка выполнения условия NWds > I. Если условие истинно, перейти к шагу 29, иначе – к шагу 30 |
|  | Inc(NWds) |
|  | Inc(I) |
|  | Конец цикла А3. Идти к шагу 27 |
|  | Вывод Weekday |
|  | Останов. |

# Структура данных

Таблица – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| NMon | Тип пользователя  «Перечисление». Базовый тип – String | Месяц. Принимает следующие значения: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec |
| NWds | Тип пользователя  «Перечисление». Базовый тип – String | День недели. Принимает следующие значения: Mon, Tue, Wed, Thi, Fri, Sat, Sun. |
| AQuas | Array[TMon] Of  Integer | Массив, в котором хранится количество дней за прошлые месяцы |
| ADays | Array[TWds] Of String | Массив, в котором хранятся названия дней недели |
| Year, Month | Integer | Год, месяц |
| N | Integer | Количество дней |
| I | Integer | Счётчик цикла |
| Max | Integer | Лимит дней в месяце |
| Flag | Boolean | Наличие или отсутствие 29 февраля |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

|  |
| --- |
|  |
| Рисунок 1 – Схема алгоритма решения задачи по ГОСТ 19.701-90 |

|  |
| --- |
|  |
| Рисунок 2 – Схема алгоритма решения задачи по ГОСТ 19.701-90 |

# Результаты расчетов

Вследствие результатов программы на экран выводятся следующие результаты расчетов:

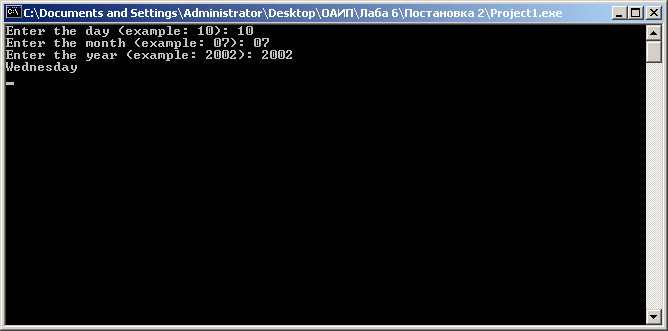


Рисунок 3 – Результаты расчетов

Приложение А

(обязательное)

Исходный код программы (постановка 1)

Пользовательские типы данных

Program Lab6;

{This program calculates the day of the week from the date}

//Use app

{$APPTYPE CONSOLE}

//Declare modules

Uses

SysUtils;

//Declare types

Type

TMon = (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec);

TWds = (Sun, Mon, Tue, Wed, Thi, Fri, Sat);

//TMon - type for months

//TWds - type for weekdays

//Declare vars

Var

NMon:TMon;

NWds:TWds;

//NMon - name of month

//NWds - name of weekday

AQuas:Array[TMon] Of Integer = (0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334);

ADays:Array[TWds] Of String = ('Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thirsday', 'Friday', 'Saturday');

//AQuas - quantities of days since 1 jan

//ADays - names of weekdays

Year, Month, N, I, Error, Max:Integer;

//Year - current year

//N - quantity of days

//I - cycle parameter

//Error - parameter for checking input

//Max - limit of month days

Input:String;

//Input - input string

Flag:Boolean;

//Flag - has 29 feb

Begin

//Checking for the correct input

Repeat

Write('Enter the year (example: 2002): ');

ReadLn(Input);

//First incorrect symbol is Error

Val(Input, Year, Error);

If (Year <= 0) Or (Error <> 0) Then

WriteLn('Invalid input. Enter another number.');

Until (Error = 0) And (Year >= 0);

//Checking for the correct input

Repeat

Write('Enter the month (example: 07): ');

ReadLn(Input);

//First incorrect symbol is Error

Val(Input, Month, Error);

If (Month <= 0) Or (Month > 12) Or (Error <> 0) Then

WriteLn('Invalid input. Enter another number.');

Until (Error = 0) And (Month > 0) And (Month <= 12);

//If has 29 feb

Flag:= (((Year Mod 4 = 0) And (Year Mod 100 <> 0)) Or (Year Mod 400 = 0));

//Convert month

NMon:= Jan;

For I:= 1 To 11 Do

If Month > I Then

Inc(NMon);

//Limit of days

Case (NMon) Of

Apr, Jun, Sep, Nov:

Max:= 30;

Feb:

Begin

Max:= 28;

If Flag Then

Inc(Max);

End;

Else

Max:= 31;

End;

//Checking for the correct input

Repeat

Write('Enter the day (example: 10): ');

ReadLn(Input);

//First incorrect symbol is Error

Val(Input, N, Error);

If (N <= 0) Or (N > Max) Or (Error <> 0) Then

WriteLn('Invalid input. Enter another number.');

Until (Error = 0) And (N > 0) And (N <= Max);

//Add all days of previous years, months + 29 febs

N:= N + (Year - 1) \* 365 + ((Year-1) Div 4) - ((Year-1) Div 100) + ((Year-1) Div 400) + AQuas[NMon];

//If there was 29 feb

If Flag And (NMon > Feb) Then

Inc(N);

//Convert weekday

NWds:= Sun;

For I:= 0 To 5 Do

If N Mod 7 > I Then

Inc(NWds);

WriteLn;

WriteLn(ADays[NWds]);

ReadLn;

End.

Приложение Б

(обязательное)

Тестовые наборы

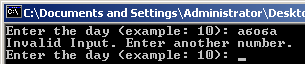
Тестовая ситуация 1: проверка ввода данных

Тестовая ситуация: проверка, сообщит ли программа о некорректном вводе, если введено буквенное, отрицательное или дробное значение.

Исходные данные: Day = абоба.

Ожидаемый результат: Invalid input. Enter another number.

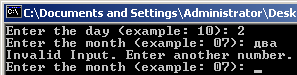
Полученный результат:



Исходные данные: Month = два.

Ожидаемый результат: Invalid input. Enter another number.

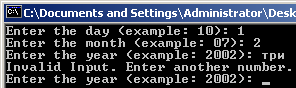
Полученный результат:



Исходные данные: Year = три.

Ожидаемый результат: Invalid input. Enter another number.

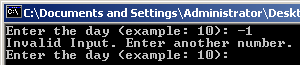
Полученный результат:



Исходные данные: Day = -1.

Ожидаемый результат: Invalid input. Enter another number.

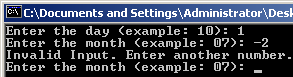
Полученный результат:



Исходные данные: Month = -2.

Ожидаемый результат: Invalid input. Enter another number.

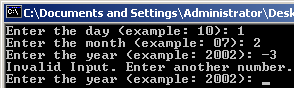
Полученный результат:



Исходные данные: Year = -3.

Ожидаемый результат: Invalid input. Enter another number.

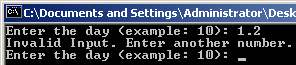
Полученный результат:



Исходные данные: Day = 1.2.

Ожидаемый результат: Invalid input. Enter another number.

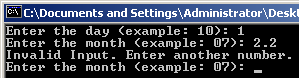
Полученный результат:



Исходные данные: Month = 2.2.

Ожидаемый результат: Invalid input. Enter another number.

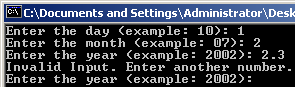
Полученный результат:



Исходные данные: Year = 2.3.

Ожидаемый результат: Invalid input. Enter another number.

Полученный результат:



Тестовая ситуация 2: проверка вычислений

Исходные данные: Day = 10, Month = 07, Year = 2002.

Ожидаемый результат: Wednesday.

Полученный результат:

