

GUI PROGRAM FOR CALENDAR FORMULAS

BERNDT E. SCHWERDTFEGER

ABSTRACT. This paper describes my implementation of the *Julian* day function in a GUI program using Qt.

PREFACE

In June 2010 I published the command line program `cal` [2] that implements the *Julian* day number j introduced by *John Herschel* in 1849. In this paper I describe my implementation with a graphical user interface (GUI), making use of the open source software Qt (pronounced *cute*).

In 2024 I updated it for Qt6 – in fact I am using Qt 6.6.1 right now.

Berlin, 27 January 2024

© 2010–2024 Berndt E. Schwerdtfeger

version 1.3, 27 January 2024

1. JULIAN DAY NUMBER, JULIAN AND GREGORIAN DATES

Julian dates proper describe all calendar dates since the calendar reform of *Julius Caesar*, valid as of Friday -44-1-1 (in modern ISO 8601 notation) until Thursday 1582-10-4. The dates October 5-14 in 1582 have been dismissed by the *Gregorian* calendar reform and the dates continued with Friday 1582-10-15.

We will assume so called *proleptic* dates: as if the *Julian* numbering of years had been in use ever since until 1582, and as if the *Gregorian* numbering will be used forever since 1582. This infinite set has to be limited somehow for practical purposes; I have chosen the *year* range from -9,999,999 to +9,999,999.

The *Julian* day number to be used in Astronomy was proposed in 1849 by astronomer *John Herschel*, based on the *Julian period* suggested by *Joseph Scaliger* in 1583. This number denotes the number of days passed since -4712-1-1. The formula expressing *Julian* day number $j(y, m, d)$ in terms of calendar dates y - m - d is given in my article *On calendar formulas* [2]. Some values are

$$\begin{array}{ll} j(-4712, 1, 1) = 0 & j(1, 1, 1) = 1,721,424 \\ j(1582, 10, 4) = 2,299,160 & j(1582, 10, 15) = 2,299,161 \\ j(2001, 1, 1) = 2,451,911 & j(2010, 12, 24) = 2,455,555 \end{array}$$

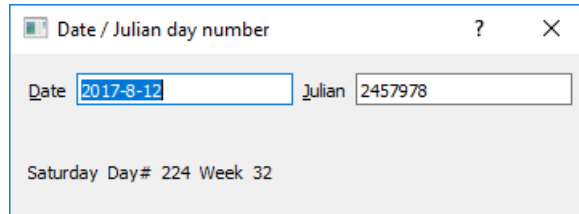
2. THE GUI VERSION OF THE CALENDAR PROGRAM

2.1. **GUI and main program.** The user interface is straightforward with two data entry fields *date* and *julian* day number, switching between them is done by the *Tab* key. Starting with a *date* the *julian* day number will be displayed and vice versa.

2020 *Mathematics Subject Classification*. Primary 68-04; Secondary 68N15.

Key words and phrases. Calendar GUI program, Julian day number, Julian and Gregorian date.

The weekday, the day of the year and the week of the year is shown as well. A regular expression validator checks for meaningful entry data. The validator allows for the 29th of February and the program corrects it silently to March 1st in non-leap years. The dismissed dates from the *Gregorian* calendar reform are *invalid*. The user interface is described in the appendix A.3.



The calendar formulas are identical to those in [2]. As I make use of the open source software Qt [1], the program structure is completely different and will only look familiar if you are used to Qt programming. The main program is the standard trivial Qt application, the dialog logic hidden in the class `CalgDialog`: //

```

/* -----
Module:   main.cpp, 2011-04-05
          Calendar formulas, GUI version
-----

Copyright 2010 Berndt E. Schwerdtfeger

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-----*/

#include <QApplication>
#include "calgdialog.h"

// -----
// main function
// -----

int main(int argc, char* argv[]){

    QApplication app(argc, argv);
    CalgDialog* dialog = new CalgDialog;
    dialog->show();
    return app.exec();
}
//

```

2.2. Date validator. The regular expression used in the validator for allowed dates looks complicated

```

[+|-]?[0-9]{1,7}-(((0?[1-9]|1[0-2])-(0?[1-9]|[1-2]\\d))\
|((0?[13-9]|1[0-2])-(30|((0?[13578]|1[02])-(31))

```

so let us look at the details.

The first part is clear

```
[+|-]?[0-9]{1,7}-
```

all years in the range $-9,999,999$ to $+9,999,999$ are allowed. The *plus*-sign $+$ is optional. After the year a *dash*-sign $-$ is obligatory.

The next part

```
((0?[1-9]|1[0-2])-(0?[1-9]|1[1-2]\\d))
```

describes $mm-dd$ with month values $01 \leq mm \leq 12$ and day values $01 \leq dd \leq 29$, with leading 0 maybe omitted.

The last two parts

```
((0?[13-9]|1[0-2])-30)|((0?[13578]|1[02])-31))
```

add the 30^{th} of all months except for February and the 31^{st} for the months 1, 3, 5, 7, 8, 10, 12.

REFERENCES

- [1] Jasmin Blanchette and Mark Summerfield, *C++ GUI Programming with Qt 4*, 2nd ed., Prentice Hall, 2008.
- [2] Berndt E. Schwerdtfeger, *On calendar formulas* (2010), available at <http://berndt-schwerdtfeger.de/wp-content/uploads/pdf/calj.pdf>.

APPENDIX A. PROGRAM LISTINGS OF CALGDIALOG

This appendix gathers the program listings for `calgdiallog` of

- the implementation of class `CalgDialog`
- the header file `calgdiallog.h`
- the user interface `calgdiallog.ui`

A.1. Implementation of class `CalgDialog`. The class `CalgDialog` implements the complete program logic, including the input validator and the user interface. The input fields emit a `editingFinished` signal in case the validator returns `Acceptable` and line editing loses focus (*Tab* key).

The signals trigger the corresponding slots: `slot on_dateEdit_editingFinished` for *date* resp. `slot on_julianEdit_editingFinished` for *julian* day number. The processing in both slots is similar (and some optimization should be done - lazy me!), more or less stolen from `cal.c`, so I do not need to explain the details.

The other functions `date`, `leap` and `quot` are well-known by now. //

```
/* -----
Module:  calgdiallog.cpp, 2011-04-05
         dialog part for calendar GUI program
         updated for Qt 6.6.1 on 27 January 2024
-----*/
```

Copyright 2010 Berndt E. Schwerdtfeger

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```

-----*/

#include <QWidget>
#include <QRegularExpressionValidator>// has changed in Qt version 6.6.1
#include "calgdialog.h"
int em[13] = {0,0,31,59,90,120,151,181,212,243,273,304,334};
QString weekdays = "Monday Tuesday Wednesday Thursday Friday Saturday Sunday";
QStringList wday = weekdays.split(" ");

// -----
CalgDialog::CalgDialog(QWidget* parent) : QDialog(parent) {

    setupUi(this);
    QRegularExpression datefmt("[+|-]?[0-9]{1,7}-(((0?[1-9]|1[0-2])-(0?[1-9]|1[1-2])\\d))\\
|((0?[13-9]|1[0-2])-(0?[13578]|1[02])-(31))");
    dateEdit->setValidator(new QRegularExpressionValidator(datefmt, this));
    QRegularExpression julianfmt("[+|-]?[0-9]{1,10}");
    julianEdit->setValidator(new QRegularExpressionValidator(julianfmt, this));

// connect(quitButton, SIGNAL(clicked()), this, SLOT(reject()));

}

// -----
void CalgDialog::on_dateEdit_editingFinished() {

    ErrMsg->setText(" "); // clear error message
    QString text = dateEdit->text(); // pull out the string
    QStringList datelist = text.split("-",Qt::SkipEmptyParts);
    long long y = datelist[0].toLongLong(); // set year
    if (text.left(1) == "-") y=-y; // correct if negative
    int m = datelist[1].toInt(); // set month
    int d = datelist[2].toInt(); // set day
    long long j = 0; // julian

// the following code comes from cal.c

    if (y==1582 && m==10 && d>4 && d<15) {
        ErrMsg->setText("Date invalid!");
        Week->setText(" ");
        Day->setText(" ");
        weekday->setText(" ");
    }
}

```

```

        julianEdit->setText(" ");
        return;
    }
    if (y < 1582 || (y == 1582 && m*100+d < 1005)) // Julian era
        j = 1721423 + d + em[m] + (y-1)*365 + quot(y-1,4);
    else { // Gregorian era
        long long x = y - 2001;
        j = 2451910 + d + em[m] + x*365 + quot(x,4) - quot(x,100) + quot(x,400);
    }
    if (m>2) j+=leap(y); // correction for leap year
    if (j < 2299161) { // Julian era
        y=1582;
        date(j-2298883,&y,&m,&d); // calculate the date
    }
    else { // Gregorian era
        y=1583;
        date(j-2299238,&y,&m,&d); // calculate the date
    }
    int n = d + em[m]; // offset into this year
    if (m > 2)
        n += leap(y); // adjust for leap day
    if (y==1582 && n>278) n-=10; // ten days dropped in 1582
    int w = j%7; // set weekday
    if (w<0) w+=7;

    datelist[0]=QString::number(y);
    datelist[1]=QString::number(m);
    datelist[2]=QString::number(d);
    text = datelist.join("-");
    dateEdit->setText(text);
    julianEdit->setText(QString::number(j));
    Week->setText(QString::number((n-w+9)/7));
    Day->setNum(n);
    weekday->setText(wday[w]);
}

// -----
void CalgDialog::on_julianEdit_editingFinished() {
    QString text = julianEdit->text(); // pull out the string
    long long j = text.toLongLong();
    long long y = 0;
    int m,d;
    if (j < 2299161) { // Julian era
        y=1582;
        date(j-2298883,&y,&m,&d); // calculate the date
    }
    else { // Gregorian era
        y=1583;
        date(j-2299238,&y,&m,&d); // calculate the date
    }
    int n = d + em[m]; // offset into this year
    if (m > 2)

```

```

        n += leap(y);                // adjust for leap day
    if (y==1582 && n>278) n-=10;      // ten days dropped in 1582
    int w = j%7;                    // set weekday
    if (w<0) w+=7;

    QStringList datelist;
    datelist << QString::number(y) << QString::number(m) << QString::number(d);
    text = datelist.join("-");
    dateEdit->setText(text);
    julianEdit->setText(QString::number(j));
    Week->setText(QString::number((n-w+9)/7));
    Day->setNum(n);
    weekday->setText(wday[w]);
}

// -----
void CalgDialog::date(long long n, long long* y, int* m, int* d){

    int i;
    while (n > 365 + leap(*y)){      // if offset larger than # of
        n -= 365 + leap(*y);        // ... days in a year
        *y += 1;                    // ... find the correct year
    }
    while (n <= 0){                 // if offset is negative
        *y -= 1;                    // ... find the correct year
        n += 365 + leap(*y);        // ... and offset
    }
    i = leap(*y);                  // adjust for leap day
    *m = 12;
    while (n <= em[*m] + i){        // searching for the month
        *m -= 1;
        if (*m<3) i=0;
    }
    *d = n - em[*m] - i;           // setting the day
}

// -----
int CalgDialog::leap(long long y){

    int i = 0;
    if (y%4 == 0)
        i = 1;
    if (y > 1582 && y%100 == 0 && y%400 != 0)
        i = 0;
    return i;
}

// -----
long long CalgDialog::quot(long long a, long long b){

    long long x = a/b;
    if (a < 0 )                // for negative numerator ..

```

```

        x -= (a%b != 0);           // .. if remainder, subtract 1
    return x;
}

//

```

A.2. **Header file** calgdialog.h. The header file contains the declarations and includes the user interface header ui_calgdialog.h, which is automatically compiled by the user interface compiler at build time of the project. //

```

/* -----
Module:   calgdialog.h, 2011-04-05
          header file for calendar GUI program
----- */

Copyright 2010 Berndt E. Schwerdtfeger

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

-----*/
#ifndef CALGDIALOG_H
#define CALGDIALOG_H

#include <QDialog>
#include "ui_calgdialog.h"

class CalgDialog : public QDialog, public Ui::CalgDialog
{
    Q_OBJECT

public:
    CalgDialog(QWidget* parent=0);

private slots:
    void on_dateEdit_editingFinished();
    void on_julianEdit_editingFinished();

private:
    void date(long long, long long*, int*, int*);
    long long quot(long long, long long); // = [a/b] (corrected: works if a < 0)
    int leap(long long); // = 1 if year is a leap year
};

```

```
#endif
//
```

A.3. **User Interface file** calgdialog.ui. The user interface (the GUI form) was designed with Qt Designer, which stores its information in an XML file, the user interface file calgdialog.ui. The user interface compiler uic then compiles the header file ui_calgdialog.h, which I have not included.

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>CalgDialog</class>
  <widget class="QWidget" name="CalgDialog">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>360</width>
        <height>100</height>
      </rect>
    </property>
    <property name="windowTitle">
      <string>Date / Julian day number</string>
    </property>
    <widget class="QWidget" name="">
      <property name="geometry">
        <rect>
          <x>11</x>
          <y>12</y>
          <width>339</width>
          <height>71</height>
        </rect>
      </property>
      <layout class="QVBoxLayout" name="verticalLayout">
        <item>
          <layout class="QHBoxLayout" name="horizontalLayout">
            <item>
              <widget class="QLabel" name="dateLabel">
                <property name="text">
                  <string>&Date</string>
                </property>
                <property name="indent">
                  <number>1</number>
                </property>
                <property name="buddy">
                  <cstring>dateEdit</cstring>
                </property>
              </widget>
            </item>
            <item>
              <widget class="QLineEdit" name="dateEdit">
                <property name="whatsThis">
                  <string>enter date in format year-mm-dd</string>
                </property>
              </widget>
            </item>
          </layout>
        </item>
      </layout>
    </widget>
  </widget>
</ui>
```



```

    </widget>
  </item>
  <item>
    <widget class="QLabel" name="julianLabel">
      <property name="text">
        <string>&amp;Julian</string>
      </property>
      <property name="buddy">
        <cstring>julianEdit</cstring>
      </property>
    </widget>
  </item>
  <item>
    <widget class="QLineEdit" name="julianEdit">
      <property name="whatsThis">
        <string>julian day number</string>
      </property>
    </widget>
  </item>
</layout>
</item>
<item>
  <widget class="QLabel" name="ErrMsg">
    <property name="enabled">
      <bool>true</bool>
    </property>
    <property name="whatsThis">
      <string>error message</string>
    </property>
    <property name="text">
      <string/>
    </property>
    <property name="indent">
      <number>0</number>
    </property>
  </widget>
</item>
<item>
  <layout class="QHBoxLayout" name="horizontalLayout_2">
    <item>
      <widget class="QLabel" name="weekday">
        <property name="whatsThis">
          <string>weekday</string>
        </property>
        <property name="text">
          <string>Wednesday</string>
        </property>
        <property name="indent">
          <number>0</number>
        </property>
      </widget>
    </item>
  </layout>
</item>

```

```
<item>
  <widget class="QLabel" name="dayLabel">
    <property name="text">
      <string>Day#</string>
    </property>
  </widget>
</item>
<item>
  <widget class="QLabel" name="Day">
    <property name="whatsThis">
      <string>day number of the year</string>
    </property>
    <property name="text">
      <string>000</string>
    </property>
  </widget>
</item>
<item>
  <widget class="QLabel" name="weekLabel">
    <property name="text">
      <string>Week</string>
    </property>
  </widget>
</item>
<item>
  <widget class="QLabel" name="Week">
    <property name="whatsThis">
      <string>week number of the year</string>
    </property>
    <property name="text">
      <string>00</string>
    </property>
  </widget>
</item>
<item>
  <spacer name="horizontalSpacer">
    <property name="orientation">
      <enum>Qt::Horizontal</enum>
    </property>
    <property name="sizeHint" stdset="0">
      <size>
        <width>40</width>
        <height>20</height>
      </size>
    </property>
  </spacer>
</item>
</layout>
</item>
</layout>
</widget>
</widget>
```

```
<tabstops>
  <tabstop>dateEdit</tabstop>
  <tabstop>julianEdit</tabstop>
</tabstops>
<resources/>
<connections/>
</ui>
```

A.4. Changes made for new versions of Qt. After installing Qt5 in October 2013 I noticed that several classes have been moved from *QtGui* to the new module *QtWidgets*. Therefore, the project file has to be adapted.

Again in 2024 after migrating to Qt6 more modifications where necessary. These changes have been applied to `calgdialog.cpp`:

- `QRegExpValidator` changed to `QRegularExpressionValidator`
- `QRegExp` changed to `QRegularExpression`
- `QString::SkipEmptyParts` changed to `Qt::SkipEmptyParts`

The steps for make are as follows:

- create a directory `calg` and put the project files into it
- create the project file `calg.pro`: `qmake6 -project`
- edit the project file and add the string `QT += widgets`
- create the makefiles and subdirs (debug, release): `qmake6`
- make the executable: `make release`