# 程序说明

我使用C语言完成算法的编写，使用Qt 5.7.0来完成图形界面设计布局等，Qt是一个1991年由奇趣科技开发的跨平台[C++](http://baike.baidu.com/item/C%2B%2B)[图形用户界面](http://baike.baidu.com/item/%E5%9B%BE%E5%BD%A2%E7%94%A8%E6%88%B7%E7%95%8C%E9%9D%A2)应用程序开发框架。开发使用的工具QtCreator。

整个项目的相关文档我都传到了<https://github.com/805436085/math_course.git>上。

所有功能的代码都在widget.cpp里，可以用记事本打开它。

[**math\_course**](https://github.com/805436085/math_course)/[course](https://github.com/805436085/math_course/tree/master/course)/[course](https://github.com/805436085/math_course/tree/master/course/course)/[qt\_course](https://github.com/805436085/math_course/tree/master/course/course/qt_course)/[qt\_course](https://github.com/805436085/math_course/tree/master/course/course/qt_course/qt_course)/**demo**目录下的[qt\_course.exe](https://github.com/805436085/math_course/blob/master/course/course/qt_course/qt_course/demo/qt_course.exe)可以双击打开使用，注意同级目录里的文件不能少。

#include "widget.h"

#include "ui\_widget.h"

#include <cmath>

int num = 0;//工件总数

double z, s, c, v, a, b;//alpha,beta,delta,gamma,a,b六个参数

int piece\_time[MAX] = { 0 };//各工件时间：3，4，6，9，14，18，20

double w[MAX] = { 0 };//各工件的小w

double W[MAX] = { 0 };//各工件的大W

double P[MAX] = { 0 };//各工件的P

double T[MAX] = { 0 };//各工件完成时的时间

double t = 0;//客观时间

double sum = 0;//累加用的变量

double F = 0;//总时间成本

int S[MAX] = {-1};

//提前完工权数

double before(double r)

{

double ret = num \* c + (r - 1) \* z;

return ret;

}

//窗口完工权数

double normal()

{

double ret = num \* v;

return ret;

}

//延迟完工权数

double delay(double r)

{

double ret = (num + 1 - r) \* s;

return ret;

}

//位置权数值

double getmin(double d1, double d2, double d3)

{

double ret = (d1 < d2) ? d1 : d2;

ret = (ret < d3) ? ret : d3;

return ret;

}

//位置函数值

double GetW(int r)

{

if (r == num)

{

return pow(r, a) \* w[r - 1];

}

else

{

double xx = 0;

for (int j = r; j < v; j++)

{

xx += GetW(j + 1);

}

return pow(r, a) \* (w[r - 1] + b \* xx);

}

}

//总的时间成本

double GetF()

{

double sum = 0;

for (int i = 0; i < num; ++i)

{

sum += W[i] \* piece\_time[i];

}

return sum;

}

//工件的实际完工时间P

double Getp(int r)

{

return (piece\_time[r] + b \* t) \* pow(r + 1, a);

}

Widget::Widget(QWidget \*parent) :

QWidget(parent),

ui(new Ui::Widget)

{

ui->setupUi(this);

setWindowTitle("基于窗口指派的准时制生产调度算法模型构建");

}

Widget::~Widget()

{

delete ui;

}

//当点击“计算”按钮，开始执行此函数，提取参数，计算结果并显示在相关窗口

void Widget::on\_pushButton\_clicked()

{

//取值

num = ui->num\_lineedit->text().toInt();

if (!num)

{

//如果num是0则弹出一个对话框并退出“计算”

QMessageBox::information(nullptr, nullptr, "num is 0");

return;

}

QString piece\_time\_str = ui->piece\_time\_textedit->toPlainText();

QStringList piece\_time\_list = piece\_time\_str.split(",");

for (int i = 0; i < num; ++i)

{

piece\_time\_str = piece\_time\_list.at(i);

piece\_time[i] = piece\_time\_str.toInt();

}

z = ui->alpha\_lineEdit->text().toDouble();

s = ui->beta\_lineEdit->text().toDouble();

c = ui->delta\_lineEdit->text().toDouble();

v = ui->gamma\_lineEdit->text().toDouble();

a = ui->a\_lineEdit->text().toDouble();

b = ui->b\_lineEdit->text().toDouble();

//循环计算w

for (int r = 0; r < num; ++r)

{

double d1 = before(r + 1);

double d2 = normal();

double d3 = delay(r + 1);

w[r] = getmin(d1, d2, d3);

if(w[r] == d1)

{

S[r] = 0;

ui->before\_listWidget->addItem(QString::number(r+1));

}

else if(w[r] == d2)

{

S[r] = 1;

ui->on\_listWidget->addItem(QString::number(r+1));

}

else if(w[r] == d3)

{

S[r] = 2;

ui->delay\_listWidget->addItem(QString::number(r+1));

}

}

//循环计算W

for (int r = 0; r < num; ++r)

{

W[r] = GetW(r + 1);

}

F = GetF();

ui->F\_lineEdit->setText(QString::number(F, 'g', 8));

for (int r = 0; r < num; ++r)

{

P[r] = Getp(r);

t += P[r];

T[r] = t;

}

//计算d、D

double d = 0;

double D = 0;

for(int r = 0; r < num; ++r)

{

if(S[r] == 0)

{

d += P[r];

}

else if(S[r] == 1)

{

D += P[r];

}

}

ui->d\_lineEdit->setText(QString::number(d));

ui->D\_lineEdit->setText(QString::number(D));

//最优排序

for(int i = 0; i < num - 1; ++i)

{

for(int j = 0; j < num - i - 1; j++)

{

if(piece\_time[j] > piece\_time[j+1])

{

int t = piece\_time[j+1];

piece\_time[j+1] = piece\_time[j];

piece\_time[j] = t;

}

}

}

QString good;

for(int i = 0; i < num; ++i)

{

good.append(QString::number(piece\_time[i]));

good.append(",");

}

ui->good\_textEdit->setText(good);

}