1. Lý thuyết

**biến static** là 1 biến có vùng nhớ riêng, vùng nhớ không bị thay đồi trong qt chạy ct,

nó không thuộc vào bất kỳ project nào, nên project nào cũng có thể gọi nó ra. hàm static tương tự

**class static** là 1 class chỉ gồm các hàm và các biến (bắt buộc) là static, 1 class static thì

không cần tạo đối tượng kiểu class

1. Hướng dẫn
2. Cấu trúc chương trình khi chạy lệnh dung BackGroundWorker

Khi người dùng nhấn bắt đầu

void batdaubtn\_Click(object sender, EventArgs e)

{

isRunning = true;

Enable\_Disable\_Control(isRunning);

Reset\_CountTime();

timer1.Start();

backgroundWorker1.RunWorkerAsync(); // goi ham do work

}

Enable\_Disable\_Control(isRunning):Vô hiệu hóa các button như bắt đầu,Chọn tang giảm,Chọn loại thuật toán, vì khi đang chạy trên GUI, nếu chọn lại loại thực toán chương trình sẽ gây lỗi.

timer1.Start():Bắt đầu đếm thời gian, ngừng khi Complete

Reset\_CountTime(): Trường hợp người dùng chạy nhiều lần, mỗi lần nhấn bắt đầu thời gian sẽ đặt lại là 0:0, nếu k reset thời gian sẽ cộng dồn.

backgroundWorker1.RunWorkerAsync():Phần chính của chương trình, gọi sự kiện backgroundWorker1.DoWork.

private void backgroundWorker1\_DoWork(object sender, DoWorkEventArgs e)

{

//Do work xu ly du lieu nhung khong the cap nhat len Gui

// Ham dowork thuc hien xong goi ham complete

HienThiThuatToan.ChayCodeC(1);

if (interchangerdbtn.Checked==true)

{

InterchangeSort();

}

if (bubblerdbtn.Checked==true)

{

BubbleSort();

}

if (selectionrdbtn.Checked==true)

{

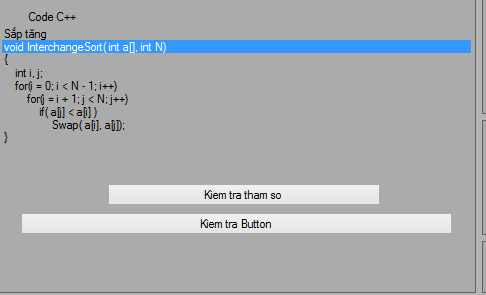
SelectionSort();

}

}

Hàm Doword có thể xữ lý dữ liệu nhưng lại không thể cập nhật GUI(Giao diện chương trình, trường hợp này nói đơn giản là mấy cái Node chạy qua chạy lại).

HienThiThuatToan.ChayCodeC(1):Chọn dòng đầu tiên trên cái Codelistbox:Lúc này nó là Tên của thuật toán,vì dòng đầu tiên của mấy cái thuật toán là y hệt nhau nên để v luôn cũng đc.



Tiếp theo là mấy cái điều kiện kiểm tra xem người dùng chọn thuật toán nào, nó sẽ chạy thuật toán đó.

Ví dụ ở đây sẽ là interchangeSoft.

private void InterchangeSort()

{

int i, j;

bool flag;

HienThiThuatToan.ChayCodeC(3);

//status st = new status();

for (i = 0; i < SoLuongNode - 1; i++)

{

HienThiThuatToan.ChayCodeC(4);

for (j = i + 1; j < SoLuongNode; j++)

{

HienThiThuatToan.ChayCodeC(5);

flag = false;

if (tangrdbtn.Checked==true)

{

HienThiThuatToan.ChayCodeC(6);

if (DanhSachThamSo[j] < DanhSachThamSo[i]) flag = true;

}

if (giamrdbtn.Checked==true)

{

HienThiThuatToan.ChayCodeC(6);

if (DanhSachThamSo[j] > DanhSachThamSo[i]) flag = true;

}

if (flag)

{

HienThiThuatToan.ChayCodeC(7);

Thread.Sleep(ThamSo.ThoiGianDoi);

DanhSachButton[i].BackColor = DanhSachButton[j].BackColor = Color.Green;

CapNhatThamSo(i, j);

DichuyenCacNode(j, i);

}

}

}

}

HienThiThuatToan.ChayCodeC: Chạy dòng tương ứng trên CodeListBox, tự test nhé

if (tangrdbtn.Checked==true)

{

HienThiThuatToan.ChayCodeC(6);

if (DanhSachThamSo[j] < DanhSachThamSo[i]) flag = true;

}

if (giamrdbtn.Checked==true)

{

HienThiThuatToan.ChayCodeC(6);

if (DanhSachThamSo[j] > DanhSachThamSo[i]) flag = true;

}

Phần này có nghĩa là xem người dùng chọn tăng hay giảm, vì chỉ có 1 trường hợp duy nhất ở 1 lần chạy là tăng hoặc giảm, nên ta dùng kỹ thuật đặt cờ hiệu, để cho chương trình biết

Cứ mỗi khi điều kiện sắp xếp đúng (flag==true)nó sẽ thực hiện dòng sau.

if (flag)

{

HienThiThuatToan.ChayCodeC(7);

Thread.Sleep(ThamSo.ThoiGianDoi);

DanhSachButton[i].BackColor = DanhSachButton[j].BackColor = Color.Green;

CapNhatThamSo(i, j);

DichuyenCacNode(j, i);

}

DanhSachButton[i].BackColor = DanhSachButton[j].BackColor = Color.Green;: Đổi màu 2 cái Node Chuẩn bị đổi vị trí thành xanh cho người dùng thấy.

CapNhatThamSo(i, j);:Dòng này rất quan trọng, nó sẽ cập nhật lại mãng, tức là nếu có 2 cái thay đổi vị trí vd: 2 3 1 , 1 và 2 đỗi vị trí, cập nhật lại mãng: 1 3 2.Nếu k cập nhật chương trình sẽ chạy sai

DichuyenCacNode(j, i): Gọi hàm đổi chỗ Node j và Node i.

private void DichuyenCacNode(int vt1, int vt2)

{

int Height = ThamSo.DoCaoDiChuyen + 10;

int \_width = Math.Abs(vt1 - vt2) \* (ThamSo.KichCoNode + ThamSo.KhoangCachCacNode);

status st = new status();

st.vt1 = vt1;

st.vt2 = vt2;

st.Type = LoaiDiChuyen.Di\_LEN\_DI\_XUONG;

for (int i = 0 ; i < Height; i=i+1)

{

// goiham progresschanged de cap nhat lai giao dien

// Khi tao brackgroundworker nho set reportprogree la true k se bi

backgroundWorker1.ReportProgress(0,st);

Thread.Sleep(ThamSo.ThoiGianDoi);

}

st.Type = LoaiDiChuyen.QUA\_PHAI\_QUA\_TRAI;

for (int i = 0; i < \_width; i=i+2)

{

backgroundWorker1.ReportProgress(0, st);

Thread.Sleep(ThamSo.ThoiGianDoi);

}

st.Type = LoaiDiChuyen.DI\_XUONG\_DI\_LEN;

for (int i = 0; i < Height; i=i+1)

{

backgroundWorker1.ReportProgress(0, st);

Thread.Sleep(ThamSo.ThoiGianDoi);

}

st.Type = LoaiDiChuyen.DUNG;

backgroundWorker1.ReportProgress(0, st);

Thread.Sleep(ThamSo.ThoiGianDoi);

}

Chú ý Khi tao brackgroundworker nho set reportprogree la true nếu k k thể truyền tham số vào hàm

backgroundWorker1.ReportProgress(0, st);

ý nghĩa Thread.Sleep(ThamSo.ThoiGianDoi) ở mỗi dòng, nó sẽ cho thread chính đi ngủ, tức là dừng lại để cho người dùng thấy hiệu ứng các Node đổi chỗ với nhau, bỏ dòng này lỗi toàn CT.

10 cái thuật toán (Trừ cái Mờ sọt) có cách di chuyển giống nhau(Đổi chỗ hai button).

Đầu tiên, ta khai báo cái status là cái class mình viết trong loại di chuyển, vt1,vt2 là cái thứ tự của I và J đang xét khi chạy thuật toán sắp xếp. vậy muốn đổi chỗ 2 Node đầu tiên phải di chuyển Node đầu lên Node sau Xuống.

st.Type = LoaiDiChuyen.Di\_LEN\_DI\_XUONG;

for (int i = 0 ; i < Height; i=i+1)

{

// goiham progresschanged de cap nhat lai giao dien

// Khi tao brackgroundworker nho set reportprogree la true k se bi

backgroundWorker1.ReportProgress(0,st);

Thread.Sleep(ThamSo.ThoiGianDoi);

}

Nó sẽ chạy lên xuống cho đến khi tới giới hạn Height

Mỗi lần chạy sẽ gọi hàm

backgroundWorker1.ReportProgress(0,st); để cập nhật giao diện, cập nhật xong chạy tiếp cho đến khi hết vòng lập, sẽ di chuyển ngang, rồi đi xuống đi lên, rồi đến trạng thái cuối cùng là dừng.

private void backgroundWorker1\_ProgressChanged(object sender, ProgressChangedEventArgs e)

{

/// Cap nhat Gui sau do goi lai ham do work

status st = e.UserState as status;// lay gia tri st truyen vao ham trn

if (st==null)

{

return;

}

if (st.Type==LoaiDiChuyen.DUNG)

{

DanhSachButton[st.vt1].BackColor = DanhSachButton[st.vt2].BackColor = Color.Honeydew;

CapNhatDanhSachNode(st.vt2, st.vt1);

return;

}

if (st.Type==LoaiDiChuyen.Di\_LEN\_DI\_XUONG)

{

DanhSachButton[st.vt1].Location = new Point(DanhSachButton[st.vt1].Location.X,DanhSachButton[st.vt1].Location.Y+1);

DanhSachButton[st.vt2].Location = new Point(DanhSachButton[st.vt2].Location.X, DanhSachButton[st.vt2].Location.Y - 1);

}

if (st.Type==LoaiDiChuyen.QUA\_PHAI\_QUA\_TRAI)

{

DanhSachButton[st.vt1].Location = new Point(DanhSachButton[st.vt1].Location.X-2, DanhSachButton[st.vt1].Location.Y);

DanhSachButton[st.vt2].Location = new Point(DanhSachButton[st.vt2].Location.X+2, DanhSachButton[st.vt2].Location.Y);

}

if (st.Type==LoaiDiChuyen.DI\_XUONG\_DI\_LEN)

{

DanhSachButton[st.vt1].Location = new Point(DanhSachButton[st.vt1].Location.X, DanhSachButton[st.vt1].Location.Y -1);

DanhSachButton[st.vt2].Location = new Point(DanhSachButton[st.vt2].Location.X, DanhSachButton[st.vt2].Location.Y +1);

}

}

status st = e.UserState as status; Dòng này rất quan trọng nó sẽ lấy dữ liệu cái st mà mình khai báo ở hàm DichuyenCacNode, để biết nó phải di chuyển như thế nào, đến khi Dung sẽ return.

Khi hàm sắp xếp chạy hết , nó sẽ chạy tới hàm Complete vậy là xong.

NÓi chung Gặp thuật toán nào có dòng Swap(a[x],a[y]) Ta thay bằng dòng

{

Thread.Sleep(ThamSo.ThoiGianDoi);

DanhSachButton[x].BackColor = DanhSachButton[y].BackColor = Color.Green;

CapNhatThamSo(x, y );

DichuyenCacNode(x, y);

}

Nhớ thêm cái cờ hiệu để biến xếp tăng hay giảm