GIẢI THÍCH CHI TIẾT

BÀI TOÁN:

- Xây dựng: DaThuc là một danh sách liên kết cấu trúc Head không vòng dùng để chứa một đa thức, mỗi phần tử chứa một số hạng và có ba vùng: vùng HeSo chứa một số thực là hệ số của số hạng, vùng SoMu chứa một số nguyên là số mũ của số hạng và vùng KeTiep chỉ đến số hạng kế tiếp.

- Cấu trúc lớp xây dựng cho toàn bộ các yêu cầu của bài toán:

+ Lớp SoHang:

public class SoHang

{

public double HeSo { get; set; }

public int SoMu { get; set; }

public SoHang KeTiep { get; set; }

}

+ Lớp DaThuc:

public class DaThuc

{

public SoHang Head;

public DaThuc()

{

Head = null;

}

}

+ Có bổ sung phương thức InDaThuc() để có cái nhìn trực quan:

public void InDaThuc()

{

SoHang p = Head;

bool isFirstTerm = true;

while (p != null)

{

if (p.HeSo != 0)

{

if (isFirstTerm)

{

Console.Write(p.HeSo);

isFirstTerm = false;

}

else if (p.HeSo > 0)

{

Console.Write(" + " + p.HeSo);

}

else

{

Console.Write(" - " + (-p.HeSo));

}

if (p.SoMu != 0)

{

Console.Write("x^" + p.SoMu);

}

}

p = p.KeTiep;

}

if (isFirstTerm)

{

Console.Write(0);

}

}

1. Phương thức daThuc.Them(heso, somu):

public void Them(double heso, int somu)

{

var newSoHang = new SoHang { HeSo = heso, SoMu = somu };

if (Head == null)

{

Head = newSoHang;

Head.KeTiep = Head;

}

else

{

SoHang prev = null, curr = Head;

while (curr != Head && curr.SoMu > somu)

{

prev = curr;

curr = curr.KeTiep;

}

if (curr == Head && curr.SoMu <= somu)

{

newSoHang.KeTiep = Head;

Head = newSoHang;

}

else

{

prev.KeTiep = newSoHang;

newSoHang.KeTiep = curr;

}

}

}

Main:

DaThuc daThuc = new DaThuc();

daThuc.Them(3.2, 5);

daThuc.Them(-1.7, 3);

daThuc.Them(0.8, 1);

Console.WriteLine("Cac so hang cua da thuc la:");

SoHang current = daThuc.Head;

while (current != null)

{

Console.WriteLine("{0}x^{1}", current.HeSo, current.SoMu);

current = current.KeTiep;

}

Console.WriteLine();

Output:

A black screen with white text

Description automatically generated with low confidence

2. Phương thức daThuc.RutGon( ):

public void RutGon()

{

SoHang prev = null, curr = Head;

while (curr != null)

{

if (curr.HeSo == 0)

{

if (prev != null)

{

prev.KeTiep = curr.KeTiep;

}

else

{

Head = curr.KeTiep;

}

}

else

{

SoHang temp = curr.KeTiep;

while (temp != null && temp.SoMu == curr.SoMu)

{

curr.HeSo += temp.HeSo;

temp = temp.KeTiep;

}

prev = curr;

curr = curr.KeTiep;

}

}

}

Main:

DaThuc daThuc = new DaThuc();

daThuc.Them(3.2, 5);

daThuc.Them(-1.7, 3);

daThuc.Them(0.8, 1);

Console.WriteLine("Da thuc truoc khi rut gon: ");

daThuc.InDaThuc();

daThuc.RutGon();

Console.WriteLine("Da thuc sau khi rut gon: ");

daThuc.InDaThuc();

Console.WriteLine();

Output:

A screenshot of a computer screen

Description automatically generated with low confidence

3. phương thức daThuc1.Cong(dathuc2):

public static DaThuc Cong(DaThuc dathuc1, DaThuc dathuc2)

{

DaThuc result = new DaThuc();

SoHang p1 = dathuc1.Head;

SoHang p2 = dathuc2.Head;

while (p1 != null || p2 != null)

{

if (p1 == null)

{

result.Them(p2.HeSo, p2.SoMu);

p2 = p2.KeTiep;

}

else if (p2 == null)

{

result.Them(p1.HeSo, p1.SoMu);

p1 = p1.KeTiep;

}

else if (p1.SoMu > p2.SoMu)

{

result.Them(p1.HeSo, p1.SoMu);

p1 = p1.KeTiep;

}

else if (p1.SoMu < p2.SoMu)

{

result.Them(p2.HeSo, p2.SoMu);

p2 = p2.KeTiep;

}

else

{

double heso = p1.HeSo + p2.HeSo;

if (heso != 0)

{

result.Them(heso, p1.SoMu);

}

p1 = p1.KeTiep;

p2 = p2.KeTiep;

}

}

result.RutGon();

return result;

}

Main:

DaThuc dathuc1 = new DaThuc();

dathuc1.Them(2, 3);

dathuc1.Them(3, 2);

dathuc1.Them(4, 1);

DaThuc dathuc2 = new DaThuc();

dathuc2.Them(5, 3);

dathuc2.Them(3, 2);

dathuc2.Them(1, 1);

DaThuc dathucTong = dathuc1.Cong(dathuc2);

dathuc1.InDaThuc();

Console.WriteLine("+");

dathuc2.InDaThuc();

Console.WriteLine("=");

dathucTong.RutGon();

dathucTong.InDaThuc();

Console.WriteLine();

Output:

A picture containing text, font, screenshot, typography

Description automatically generated

4. phương thức daThuc.DoiDau( ):

public void DoiDau()

{

SoHang current = Head;

while (current != null)

{

current.HeSo \*= -1;

current = current.KeTiep;

}

}

Main:

DaThuc daThuc3 = new DaThuc();

daThuc3.Them(-3, 3);

daThuc3.Them(-2, 5);

daThuc3.Them(8, 9);

Console.Write("Truoc khi doi dau: ");

daThuc3.InDaThuc();

daThuc3.DoiDau();

Console.Write("Sau khi doi dau: ");

daThuc3.InDaThuc();

Console.WriteLine();

Output:

A black background with white text

Description automatically generated with low confidence

5. phương thức daThuc1.Tich(daThuc2):

public DaThuc Tich(DaThuc dathuc2)

{

DaThuc result = new DaThuc();

SoHang current1 = Head;

while (current1 != null)

{

SoHang current2 = dathuc2.Head;

while (current2 != null)

{

double heSo = current1.HeSo \* current2.HeSo;

int soMu = current1.SoMu + current2.SoMu;

result.Them(heSo, soMu);

current2 = current2.KeTiep;

}

current1 = current1.KeTiep;

}

result.RutGon();

return result;

}

Main:

DaThuc dathucTich = dathuc1.Tich(dathuc2);

dathuc1.InDaThuc();

Console.WriteLine("\*");

dathuc2.InDaThuc();

Console.WriteLine("=");

dathucTich.RutGon();

dathucTich.InDaThuc();

Console.WriteLine();

Output:

A picture containing text, font, screenshot

Description automatically generated

6. phương thức daThuc.Chep( )

public DaThuc Chep()

{

DaThuc newDathuc = new DaThuc();

SoHang currentNode = Head;

while (currentNode != null)

{

SoHang newNode = new SoHang();

newNode.HeSo = currentNode.HeSo;

newNode.SoMu = currentNode.SoMu;

if (newDathuc.Head == null)

{

newDathuc.Head = newNode;

}

else

{

SoHang lastNode = newDathuc.Head;

while (lastNode.KeTiep != null)

{

lastNode = lastNode.KeTiep;

}

lastNode.KeTiep = newNode;

}

currentNode = currentNode.KeTiep;

}

return newDathuc;

}

Main:

Console.Write("Da thuc can duoc sao chep: ");

dathuc1.InDaThuc();

DaThuc daThucChep = dathuc1.Chep();

Console.Write("Da thuc chep: ");

daThucChep.InDaThuc();

Output:

A black background with white text

Description automatically generated with low confidence