



using System;

using System.Collections.Generic;

using System.Text;

namespace AVL\_tree

{

public class Client :IComparable

{

private String m\_clientName;

private int m\_printingTime;

private int m\_leftTime;

private int m\_timeInQueue;

private int m\_priority;

public Client()

{

Random rtime = new Random();

int randtime = (int)(rtime.Next(5)+15);

Random rprio = new Random();

int randprio = (int)(rprio.Next(100));

Random rchar = new Random();

this.m\_clientName += (char)(rchar.Next(25) + (int)'A');

for (int i = 0; i < 7; i++)

this.m\_clientName += (char)(rchar.Next(25)+(int)'a');

this.m\_printingTime = randtime;

this.m\_leftTime = this.m\_printingTime;

this.m\_priority = randprio;

this.m\_timeInQueue = 0;

}

public String GetClientName()

{

return m\_clientName;

}

public int GetPrintingTime()

{

return m\_printingTime;

}

public int GetLeftTime()

{

return m\_leftTime;

}

public int GetTimeInQueue()

{

return m\_timeInQueue;

}

public int GetPriority()

{

return m\_priority;

}

public void DecLeftTime()

{

if (this.m\_leftTime > 0)

this.m\_leftTime--;

else

this.m\_leftTime = 0;

}

public void UpdateTimeInQueue()

{

m\_timeInQueue++;

}

public int CompareTo(object client)

{

const string s = "сравнимый объект не принадлежит классу Client";

Client p = client as Client;

if (!p.Equals(null))

return m\_priority.CompareTo(p.m\_priority);

throw new ArgumentException(s);

}

public static bool operator < (Client p1, Client p2)

{

return (p1.CompareTo(p2) < 0);

}

public static bool operator >(Client p1, Client p2)

{

return (p1.CompareTo(p2) > 0);

}

public static bool operator <= (Client p1, Client p2)

{

return (p1.CompareTo(p2) <= 0);

}

public static bool operator >=(Client p1, Client p2)

{

return (p1.CompareTo(p2) >= 0);

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AVL\_tree

{

public partial class Form1 : Form

{

private AVLTree<Client> m\_printerQueue = new AVLTree<Client>();

private Queue<Client> temp = new Queue<Client>();

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

timer1.Start();

}

private void timer1\_Tick(object sender, EventArgs e)

{

int count = m\_printerQueue.Count;

int count2 = temp.Count;

for (int i = 0; i < count2; i++)

{

Client peekedClient = temp.Dequeue();

peekedClient.UpdateTimeInQueue();

temp.Enqueue(peekedClient);

}

if (m\_printerQueue.Count > 0)

{

Client peekedClient = m\_printerQueue.MaxValue;

peekedClient.DecLeftTime();

if (peekedClient.GetLeftTime() == 0)

{

m\_printerQueue.Remove(peekedClient);

richTextBox2.Text += " " + peekedClient.GetClientName() + " (priority: " + peekedClient.GetPriority().ToString() + ";time: " + peekedClient.GetPrintingTime().ToString() + ") has finished printing\n";

}

ShowPrintQueue(m\_printerQueue.Count);

}

Random r = new Random();

int rand = (int)(r.Next(7));

if ((rand != 0 && m\_printerQueue.Count > 3) || m\_printerQueue.Count > 15) return;

Client client = new Client();

richTextBox1.Text = "> Client <" + client.GetClientName() + "> has joined to the printing queue...\n";

richTextBox1.Text += " " + client.GetClientName() + "'s priority is " + client.GetPriority().ToString() + "/100\n " + client.GetClientName() + "'s time for printing is " + client.GetPrintingTime().ToString() + "sec.\n";

m\_printerQueue.Add(client);

temp.Enqueue(client);

ShowPrintQueue(m\_printerQueue.Count);

}

private void ShowPrintQueue(int highlightedOne)

{

richTextBox3.Text = "";

AVLTree<Client> reservedQueue = new AVLTree<Client>();

Client[] clientArr = new Client[highlightedOne];

m\_printerQueue.CopyTo(clientArr);//copy queue

foreach (Client element in clientArr)

{

reservedQueue.Add(element);

}

int count = reservedQueue.Count;

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

{

Client peekedClient = reservedQueue.MaxValue;

bool d = reservedQueue.Remove(peekedClient);

richTextBox3.Text += (i + 1).ToString() + ". " + peekedClient.GetClientName() + " (" + peekedClient.GetPriority() + ")" + " [" + peekedClient.GetLeftTime() + "]";

if (i == highlightedOne || peekedClient.GetTimeInQueue() < 3)

richTextBox3.Text += " <<<";

richTextBox3.Text += "\n";

}

}

}

}