Tutorial 18

Programming exercise 1 – MyForms

1. Add new class to WIUT project, call it MyForms
2. Add the following code to MyForms class

public static T GetForm<T>() where T : class, new()

{

return Application.OpenForms.OfType<T>().FirstOrDefault() ?? new T();

}

This code is using some advanced .Net concepts. You may want to do some research on the topic of “generic parameters” and “generic parameters constraint”

Programming exercise 2 – Applicants list form

1. Add new form. Name it ApplicantListForm.
2. Set Text property of the form to Applicants
3. Add DataGridView control. Name it dgv. Set the following properties:
   1. ReadOnly = true
   2. AllowUserToAddRows = false
   3. AllowUserToDeleteRows = false
   4. AllowUserToOrderColumns = false
   5. SelectionMode = FullRowSelect
4. Add 4 buttons – Add, Update, Delete, Refresh
5. Add a combobox and a button to facilitate sorting
6. Add a combobox, textbox and a button to facilitate searching
7. Use Anchor property of the controls to enable proper resizing of the form
8. Add code in form load event of ApplicantEventForm

MdiParent = MyForms.GetForm<ParentForm>();

1. Add code for “All applicants” menu item in ParentForm

MyForms.GetForm<ApplicantListForm>().Show();

1. Run the program and see the result
2. Return to ApplicantListForm and go to Data Sources ->Add new data source. Select Object. Navigate to and select WIUT.DAL.Applicant.
3. Set up DataGridView to be bound to newly created object data source.
4. Create a new function inside ApplicantListForm

private void LoadData()

{

dgv.DataMember = "";

dgv.DataSource = null;

dgv.DataSource = new ApplicantList().GetAllApplicants();

}

1. Call LoadData function from FormLoad event handler and Refresh button
2. Populate the database with sample data if needed and run the project
3. Add items to sort combobox

Name

Last name

Date of birth

Course

1. Add the code to Sort button

private void btnSort\_Click(object sender, EventArgs e)

{

if (cbxSort.SelectedIndex < 0)

MessageBox.Show("Select an attribute to sort by");

else

{

ByAttribute selectedAttribute;

if (cbxSort.SelectedIndex == 0)

selectedAttribute = ByAttribute.Name;

else if (cbxSort.SelectedIndex == 1)

selectedAttribute = ByAttribute.Surname;

else if (cbxSort.SelectedIndex == 2)

selectedAttribute = ByAttribute.DoB;

else selectedAttribute = ByAttribute.Course;

dgv.DataMember = "";

dgv.DataSource = null;

dgv.DataSource = new ApplicantList().Sort(selectedAttribute);

}

}

1. Run the project and see the result
2. Add items to search combobox

Name

Last name

1. Add code for Search button

private void btnSearch\_Click(object sender, EventArgs e)

{

if (cbxSearch.SelectedIndex < 0)

MessageBox.Show("Select an attribute to search by");

else if (string.IsNullOrWhiteSpace(tbxSearch.Text))

MessageBox.Show("Provide the search term");

else

{

var selectedAttribute = cbxSearch.SelectedIndex == 0 ? ByAttribute.Name : ByAttribute.Surname;

dgv.DataMember = "";

dgv.DataSource = null;

dgv.DataSource = new ApplicantList().Search(tbxSearch.Text, selectedAttribute);

}

}

1. Add override for ToSting() function in Course class to return Name property
2. Run the project and see the result

Programming exercise 3 – Course list form

Independent task – create course list form

Home work

Do research on Queue and Stack data structures