I. Tài liệu kiểm thử:

1. Chương trình:

1.1 Chương trình giao diện:

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

{

string name = txbFoodName.Text;

int categoryID = (cbFoodCategory.SelectedItem as Category).ID;

float price = (float)nmFoodPrice.Value;

if (FoodDAO.Instance.InsertFood(name, categoryID, price))

{

MessageBox.Show("Thêm món thành công");

LoadListFood();

if (insertFood != null)

insertFood(this, new EventArgs());

}

else

{

MessageBox.Show("Có lỗi khi thêm thức ăn");

}

}

1.2 Chương trình chính:

public bool InsertFood(string name, int id, float price)

{

string query = string.Format("INSERT dbo.Food ( name, idCategory, price )VALUES ( N'{0}', {1}, {2})", name, id, price);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

1.3 Chương trình phụ:

public int ExecuteNonQuery(string query, object[] parameter = null)

{

int data = 0;

using (SqlConnection connection = new SqlConnection(connectionSTR))

{

connection.Open();

SqlCommand command = new SqlCommand(query, connection);

if (parameter != null)

{

string[] listPara = query.Split(' ');

int i = 0;

foreach (string item in listPara)

{

if (item.Contains('@'))

{

command.Parameters.AddWithValue(item, parameter[i]);

i++;

}

}

}

data = command.ExecuteNonQuery();

connection.Close();

}

return data;

}

1.4 Chương trình lấy danh sách thức ăn:

public List<Food> GetListFood()

{

List<Food> list = new List<Food>();

string query = "select \* from Food";

DataTable data = DataProvider.Instance.ExecuteQuery(query);

foreach (DataRow item in data.Rows)

{

Food food = new Food(item);

list.Add(food);

}

return list;

}

1.5 Chương trình cập nhập danh sách hiển thị món:

void LoadListFood()

{

foodList.DataSource = FoodDAO.Instance.GetListFood();

}

II. Kiểm thử:

1. Xác định các nút:

1.1 Chương trình giao diện:

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

{ (1)

string name = txbFoodName.Text; (2)

int categoryID = (cbFoodCategory.SelectedItem as Category).ID; (3)

float price = (float)nmFoodPrice.Value; (4)

if (FoodDAO.Instance.InsertFood(name, categoryID, price)) (5)

{

MessageBox.Show("Thêm món thành công"); (6)

LoadListFood(); (7)

if (insertFood != null) (8)

insertFood(this, new EventArgs()); (9)

}

else

{

MessageBox.Show("Có lỗi khi thêm thức ăn"); (10)

}

} (11)

1.2 Chương trình chính:

public bool InsertFood(string name, int id, float price)

{ (12)

string query = string.Format("INSERT dbo.Food ( name, idCategory, price )VALUES ( N'{0}', {1}, {2})", name, id, price); (13)

int result = DataProvider.Instance.ExecuteNonQuery(query); (14)

return result > 0; (15)

} (16)

1.3 Chương trình phụ:

public int ExecuteNonQuery(string query, object[] parameter = null)

{ (17)

int data = 0; (18)

using (SqlConnection connection = new SqlConnection(connectionSTR)) (19)

{

connection.Open(); (20)

SqlCommand command = new SqlCommand(query, connection); (21)

if (parameter != null) (22)

{

string[] listPara = query.Split(' '); (23)

int i = 0; (24)

foreach (string item in listPara) (25)

{

if (item.Contains('@')) (26)

{

command.Parameters.AddWithValue(item, parameter[i]); (27)

i++; (28)

}

} (29)

}

data = command.ExecuteNonQuery(); (30)

connection.Close(); (31)

}

return data; (32)

} (33)

1.4 Chương trình lấy danh sách thức ăn:

public List<Food> GetListFood()

{ (34)

List<Food> list = new List<Food>(); (35)

string query = "select \* from Food"; (36)

DataTable data = DataProvider.Instance.ExecuteQuery(query); (37)

foreach (DataRow item in data.Rows) (38)

{

Food food = new Food(item); (39)

list.Add(food); (40)

} (41)

return list; (42)

} (43)

1.5 Chương trình cập nhập danh sách hiển thị món:

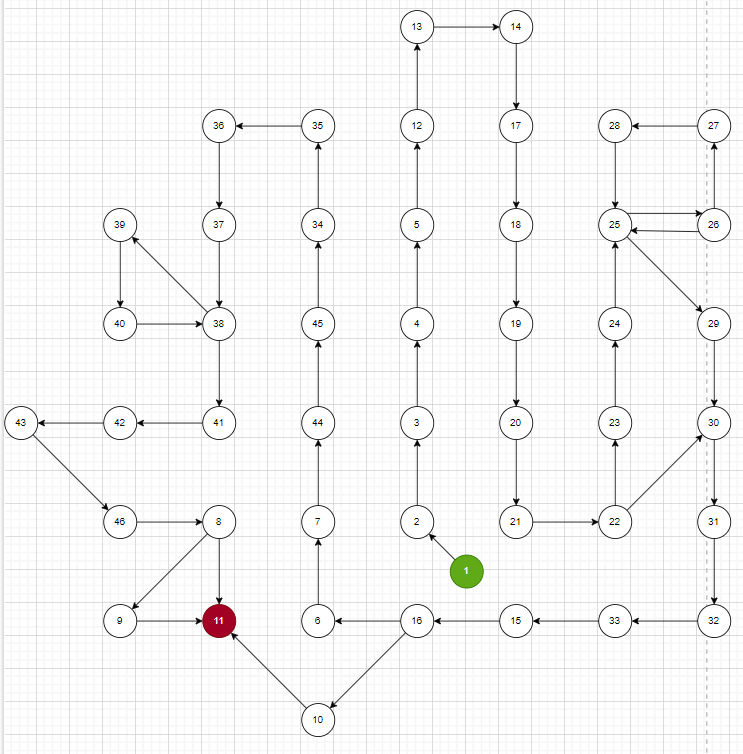
void LoadListFood()

{ (44)

foodList.DataSource = FoodDAO.Instance.GetListFood(); (45)

} (46)

2. Đồ thị thể hiện đường diễn tiến của chương trình:



Xác định số trường hợp kiểm thử V(G):

V(G) = 51 – 46 + 2 = 7

3. Các đường kiểm thử là:

|  |  |
| --- | --- |
| 1. | 1.2.3.4.5.12.13.14.17.18.19.20.21.22.23.24.25.26.27.28.25…29.30.31.32.33.15.16.6.7.44.45.34.35.36.37.38.39.40.38…41.42.43.46.8.9.11 |
| 2. | 1.2.3.4.5.12.13.14.17.18.19.20.21.22.30.31.32.33.15.16.6.7.44.45.34.35.36.37.38.39.40.38…41.42.43.46.8.9.11 |
| 3. | 1.2.3.4.5.12.13.14.17.18.19.20.21.22.23.24.25.29.30.31.32.33.15.16.6.7.44.45.34.35.36.37.38.39.40.38…41.42.43.46.8.9.11 |
| 4. | 1.2.3.4.5.12.13.14.17.18.19.20.21.22.23.24.25.26.25…29.30.31.32.33.15.16.6.7.44.45.34.35.36.37.38.39.40.38…41.42.43.46.8.9.11 |
| 5. | 1.2.3.4.5.12.13.14.17.18.19.20.21.22.23.24.25.26.27.28.25…29.30.31.32.33.15.16.6.7.44.45.34.35.36.37.38.41.42.43.46.8.9.11 |
| 6. | 1.2.3.4.5.12.13.14.17.18.19.20.21.22.23.24.25.26.27.28.25…29.30.31.32.33.15.16.6.7.44.45.34.35.36.37.38.39.40.38…41.42.43.46.8.11 |
| 7. | 1.2.3.4.5.12.13.14.17.18.19.20.21.22.23.24.25.26.27.28.25…29.30.31.32.33.15.16.10.11 |

4. Các trường hợp kiểm thử:

|  |  |  |
| --- | --- | --- |
| Đường kiểm thử | Giá trị đầu vào | Kết quả mong đợi |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |