

## BÀI TẬP ÔN CHUYÊN ĐỀ C#

---o0o---

## PHẦN 1. NGÔN NGỮ LẬP TRÌNH

## Bài 1.

a. Chương trình có lỗi không? Nếu có, hãy giải thích? Nếu không, cho biết kết quả xuất?

```

class A {
    public int index = 5;
    public virtual void printVal() { System.Console.WriteLine("Super"); }
}
class B : A{
    int index = 2;
    public override void printVal() { System.Console.WriteLine("Sub"); }
}
public class Program {
    public static void Main()
    { A t = new B(); System.Console.WriteLine(t.index + ","); sup.printVal(); }
}

```

b. Hãy cho biết kết quả xuất của đoạn mã sau:

```

class Test {
    static void Main() { A.F(); B.F(); }
}
class A {
    static A() { Console.WriteLine("Init A"); }
    public static void F() { Console.WriteLine("A.F"); }
}
class B {
    static B() { Console.WriteLine("Init B"); }
    public static void F() { Console.WriteLine("B.F"); }
}

```

**Bài 2.** Cho lớp phân số (**PhanSo**) gồm hai thuộc tính: tử số (**m\_nTuSo**) và mẫu số (**m\_nMauSo**) với khai báo:

```

class PhanSo
{
    private int m_nTuSo;
    private int m_nMauSo;
    ...
}

```

Giả sử yêu cầu mảng phân số thực hiện các yêu cầu sau:

```

static void Main(string[] args)
{
    PhanSo a(5);    // a = [5 / 1]
    PhanSo b(1,2);  // b = [1 / 2]

    Console.WriteLine("a = {0}, b = {1}", a, b);

    PhanSo c = a + b;
    Console.WriteLine("kq = {0} + {1} = {2}", a, b, a+b);
    Console.WriteLine("Gia tri phan so a = {0}, b = {1}", (float)a, (float)b);
}

```

Hãy xác định và cài đặt các phương thức cần thiết.

**Bài 3.** Một lớp học bao gồm danh sách học sinh (**dsHocSinh**), mã lớp (**maLop**), tên lớp học (**tenLop**). Thông tin mỗi học sinh gồm mã số (**maHS**), họ lót (**hoLot**), tên học sinh (**tenHS**), giới tính (**gioiTinh**) – giá trị true cho học sinh nam và false cho học sinh nữ, năm sinh (**namSinh**). Hãy thực hiện:

- Khai báo lớp **LopHoc** và **HocSinh** có các thuộc tính theo mô tả trên.
- Xây dựng các phương thức và tính chất thỏa mãn hàm Main:

```

static void Main(string[] args)
{
    LopHoc lop1 = new LopHoc("10A1", "Lop 10 A1");
    LopHoc lop2 = new LopHoc("10A2", "Lop 10 A2");
}

```

```
LopHoc lop = new LopHoc("10A", "Lop 10 A");

lop1.ThemHocSinh(new HocSinh("01", "Nguyen Van", "A", true, 1988));
lop2.ThemHocSinh(new HocSinh("02", "Tran Thi", "B", false, 1989));

lop = lop1 + lop2;

Console.WriteLine("Hien thi thong tin lop {0}: ", lop.TenLop);
for (int i = 0; i <= lop.SiSo - 1; i++)
{
    Console.WriteLine("{0}", lop[i]);
}
}
```

**Bài 4.** Cho chương trình tính tiền thuê phòng của khách sạn, biết:

- Một phiếu thuê phòng khách sạn bao gồm họ tên khách hàng, ngày bắt đầu, ngày kết thúc, loại phòng thuê.
- Khách sạn có 3 loại phòng:
  - Loại A có đơn giá 220000 Đ / ngày
  - Loại B có đơn giá 200000 Đ / ngày
  - Loại C có đơn giá 170000 Đ / ngày
- Nếu thuê trên 6 ngày được giảm 10% tiền.

Hãy thiết kế và cài đặt các lớp cho bài toán trên.

## PHẦN 2. WINDOWS FORM

**Bài 1.** Cho chương trình Bán hàng (*frmBanHang*) với giao diện bên dưới:

Bảng mô tả các điều khiển:

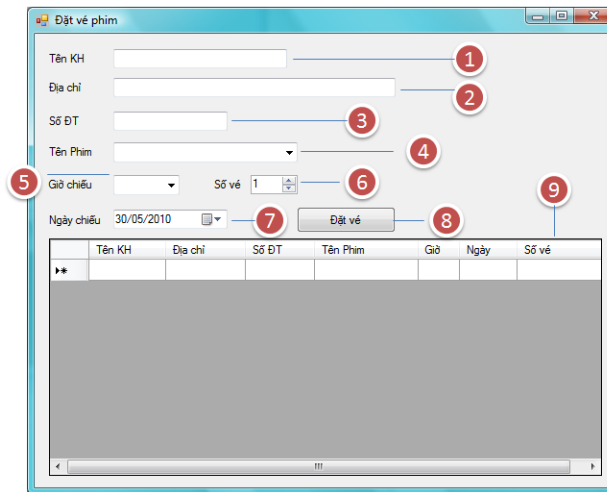
| STT | Tên điều khiển | Kiểu điều khiển | Mô tả   |
|-----|----------------|-----------------|---|
| 1   | txtMaHoaDon    | TextBox         | Mã hóa đơn, có dạng HD-{số}: HD-01, HD-2, ... |
| 2   | datNgayLap     | DateTimePicker  | Ngày lập hóa đơn                              |
| 3   | txtKhachHang   | TextBox         | Tên khách hàng                                |
| 4   | txtLienLac     | TextBox         | Địa chỉ liên lạc                              |
| 5   | cboMatHang     | ComboBox        | Danh sách mặt hàng bán                        |
| 6   | txtDonGia      | TextBox         | Đơn giá mặt hàng                              |
| 7   | txtSoLuong     | TextBox         | Số lượng mua                                  |
| 8   | lvDanhSach     | ListView        | Danh sách mặt hàng chọn mua                   |
| 9   | btnChonMua     | Button          | Chức năng chọn mua                            |
| 10  | btnBoChon      | Button          | Chức năng bỏ chọn mua                         |
| 11  | txtTongTien    | TextBox         | Tổng tiền mặt hàng chọn mua                   |

Hãy chọn lựa sự kiện và thực hiện viết chương trình theo yêu cầu sau:

- Khi *frmBanHang* vừa mới hiển thị:
  - Khởi tạo giá trị cho *cboMatHang* chứa danh sách mặt hàng với thông tin cho bên dưới:
 

|                |           |
|----------------|-----------|
| + Samsung N148 | 5500000 Đ |
| + Máy bộ       | 4500000 Đ |
| + Tivi         | 2300000 Đ |
  - Ngày lập là ngày hiện tại của hệ thống, tổng tiền là 0.
- Khi chọn mặt hàng, hiển thị thông tin đơn giá tương ứng và đặt số lượng là 1.
- Thực hiện kiểm tra dữ liệu cho *txtMaHoaDon*, *txtKhachHang*, *datNgayLap*, *txtSoLuong* theo mô tả sau:
  - Mã hóa đơn phải có dạng HD-{số}: HD-01, HD-1, ...
  - Tên khách hàng không được để trống.
  - Ngày lập phải là ngày hiện hành hoặc trước ngày hiện hành.
  - Số lượng là số nguyên dương.
- Khi nhấn vào nút **Chọn mua**:
  - Lấy thông tin mặt hàng thêm vào *lvDanhSach* (Thành tiền = Đơn giá \* Số kỳ).
  - Cập nhật lại tổng tiền.
- Khi nhấn vào nút **Bỏ chọn**:
  - Hiển thị xác nhận người dùng muốn xóa. Nếu đồng ý, xóa thông tin mặt hàng đang chọn trong *lvDanhSach*.
  - Cập nhật lại tổng tiền.

**Bài 2.** Cho một ứng dụng đặt vé xem phim có giao diện như hình 1, form có tên *frmDatVe*



Hình 1: Giao diện của ứng dụng đặt vé xem phim

Bảng mô tả các control trên form

| STT | Tên control  | Kiểu control   | Mô tả   |
|-----|--------------|----------------|---|
| 1   | txtTenKH     | TextBox        | Nhập tên khách hàng đặt vé                                    |
| 2   | txtDiaChi    | TextBox        | Nhập thông tin địa chỉ khách hàng                             |
| 3   | txtDienThoai | TextBox        | Nhập số điện thoại khách hàng                                 |
| 4   | cboTenPhim   | ComboBox       | Chứa danh mục phim, cho phép chọn một phim trong danh mục này |
| 5   | cboGioChieu  | ComboBox       | Liệt kê các giờ chiếu phim.                                   |
| 6   | numSoVe      | NumericUpDown  | Chọn số lượng vé đặt  |
| 7   | dtpNgayChieu | DateTimePicker | Chọn ngày chiếu phim  |
| 8   | btnDatVe     | Button         | Chức năng đặt vé  |
| 9   | dgvDSDatVe   | DataGridView   | Chứa danh sách thông tin đặt vé                               |

Bảng 1: Danh mục các control trên form

Mô tả form frmDatVe:

- Form này có một biến thành viên là **dtDSDatVe** có kiểu DataTable, chứa các trường thông tin như sau: {tên khách hàng, địa chỉ, số điện thoại, tên phim, giờ chiếu, ngày chiếu và số vé}.

Thực hiện các yêu cầu sau:

- Viết sự kiện Load của form frmDatVe, với các mô tả như sau:
  - Khởi tạo các giá trị cho **cboTenPhim** gồm bốn phim sau: *Để mai tính*, *Bảo tấp sa mạc*, *Cô gái đi tìm tình yêu*, *Cuốn sách của Eli*.
  - Khởi tạo các giờ chiếu cho cboGioChieu: 14h, 16h, 18h, 20h, 22h.
  - Khởi tạo cấu trúc của bảng dtDSDatVe, tạo các cột tương ứng cho bảng này theo mô tả như hình 1.
  - Hiển thị dữ liệu của **dtDSDatVe** lên DataGridView **dgvDSDatVe**.
- Viết sự kiện **Validating** cho txtTenKH, txtDiaChi, txtDienThoai theo mô tả sau:
  - Tên khách hàng không được để trống.
  - Địa chỉ không được để trống.
  - Số điện thoại phải là con số hợp lệ, không có ký tự khác ngoài con số.
- Viết sự kiện cho button **btnDatVe**, thực hiện yêu cầu: lấy các thông tin nhập liệu đặt vé thêm vào DataTable **dtDSDatVe**.

## PHẦN 3. CƠ SỞ DỮ LIỆU VỚI ADO.NET

**Bài 1.** Xét chương trình *Quản lý nhân khẩu* có giao diện như sau:

Mô tả các điều khiển:

| STT | Tên đối tượng | Kiểu đối tượng |
|-----|---------------|----------------|
| 1   | txtMaHoKhu    | TextBox        |
| 2   | txtDiaChi     | TextBox        |
| 3   | txtTenChuHo   | TextBox        |
| 4   | txtMaNhanKhu  | TextBox        |
| 5   | txtHoTenNK    | TextBox        |
| 6   | txtNgaySinh   | TextBox        |
| 7   | rdoNam        | RadioButton    |

| STT | Tên đối tượng | Kiểu đối tượng |
|-----|---------------|----------------|
| 8   | rdoNu         | RadioButton    |
| 9   | cboCuTru      | ComboBox       |
| 10  | txtCMND       | TextBox        |
| 11  | lvHoKhu       | ListView       |
| 12  | lvNhanKhu     | ListView       |
| 13  | btnDocDuLieu  | Button         |
| 14  | btnThoat      | Button         |

Giả sử có cơ sở dữ liệu **QLNK** được lưu ở SQL Server trên máy cục bộ có lược đồ như sau:

**HoKhu** (MaHoKhu, ChuHo, DiaChi)

**NhanKhu** (MaNK, TenNK, NgaySinh, GioiTinh, LoaiCuTru, CMND, MaHK)

Giải thích cơ sở dữ liệu:

| STT | Tên cột   | Kiểu dữ liệu  | Ý nghĩa   |
|-----|-----------|---------------|---|
| 1   | MaHoKhu   | nvarchar (20) | Mã hộ khẩu                                      |
| 2   | ChuHo     | nvarchar(20)  | Lưu mã nhân khẩu làm chủ hộ                     |
| 3   | DiaChi    | nvarchar(255) | Địa chỉ của hộ khẩu                             |
| 4   | MaNK      | nvarchar(20)  | Mã nhân khẩu                                    |
| 5   | TenNK     | nvarchar(100) | Họ tên nhân khẩu                                |
| 6   | NgaySinh  | DateTime      | Ngày sinh                                       |
| 7   | GioiTinh  | bit           | Giới tính (True – Nam; False – Nữ)              |
| 8   | LoaiCuTru | bit           | Loại cư trú: thường trú (True), tạm trú (False) |
| 9   | CMND      | nvarchar(20)  | Chứng minh nhân dân                             |
| 10  | MaHK      | nvarchar(20)  | Mã hộ khẩu xác định hộ khẩu của nhân khẩu       |

**Yêu cầu:** Hãy vẽ lược đồ cơ sở dữ liệu và viết câu SQL tạo cơ sở dữ liệu và thêm mỗi bảng 2 dòng dữ liệu.

**Hãy lựa chọn sự kiện** cho các đối tượng thích hợp và viết mã lệnh theo yêu cầu sau:

- Nhấn nút “Đọc dữ liệu”, đưa dữ liệu hộ khẩu vào danh sách hộ khẩu.
- Chọn hộ khẩu trong danh sách hộ khẩu, hiển thị danh sách nhân khẩu của hộ khẩu đó và thông tin hộ khẩu.
- Chọn nhân khẩu trong danh sách nhân khẩu, hiển thị thông tin nhân khẩu.
- Xử lý nút “Thoát”, hiển thị hộp thoại thông báo thoát và chỉ thoát khi người dùng đồng ý.

**Bài 2.** Cho cơ sở dữ liệu có tên **HocSinhDB**, trong đó có chứa bảng dữ liệu **LopHoc** và **HocSinh**:

Bảng **LopHoc**

| Tên trường   | Kiểu dữ liệu | Mô tả                   |
|--------------|--------------|-------------------------|
| <b>MaLop</b> | char(10)     | Mã lớp học (khóa chính) |
| TenLop       | nvarchar(50) | Tên lớp học             |

Bảng **HocSinh**

| Tên trường   | Kiểu dữ liệu  | Mô tả  |
|--------------|---------------|--|
| <b>MaHS</b>  | char(10)      | Mã học sinh                                      |
| <b>HoLot</b> | nvarchar(100) | Họ lót   |
| TenHS        | nvarchar(50)  | Tên học sinh                                     |
| GioiTinh     | Bit           | True – Nam, False – Nữ                           |
| NamSinh      | Int           | Năm sinh   |
| <b>MaLop</b> | char(10)      | Mã lớp – khoá ngoại tham chiếu tới <b>LopHoc</b> |

Cho giao diện **FrmDanhSachHocSinh** có các yêu cầu:

- ComboBox (Lớp học): chứa danh sách các tên lớp học.
- ListView (Danh sách học sinh): chứa danh sách các học sinh của ComboBox Lớp học gồm các thông tin: Mã học sinh, họ tên, giới tính (Nam / Nữ), Tuổi, Tên lớp học.
- Label (Số): chứa số lượng học sinh tương ứng của lớp học trong ComboBox Lớp học.
- Button (Thoát): thực hiện thoát chương trình có yêu cầu xác thực người dùng.

**Hãy thực hiện các yêu cầu sau:**

- Thiết kế giao diện **FrmDanhSachHocSinh** và đặt tên cho các điều khiển.
- Hãy vẽ lược đồ cơ sở dữ liệu và viết câu SQL tạo cơ sở dữ liệu và thêm mỗi bảng 2 dòng dữ liệu.
- Khai báo các biến cần thiết, viết phương thức kết nối với cơ sở dữ liệu, giả sử cơ sở dữ liệu là **SQLEXPRESS** đặt tại máy cục bộ.
- Khi **chương trình mở**:
  - Nạp dữ liệu vào ComboBox Lớp học.
  - Hiển thị thông tin lớp học đầu tiên trong ComboBox Lớp học và ListView danh sách học sinh.
- Khi **chọn lớp học trong ComboBox Lớp học**, hiển thị thông tin ListView danh sách học sinh và Label Số tương ứng với lớp học đang chọn.
- Khi **nhấn Button Thoát**, thực hiện thoát chương trình có xác thực của người dùng.

**Bài 3.** Thiết kế chương trình **QuanLyHocSinh** có giao diện như sau:

**Database:** QLSV (Sử dụng hệ quản trị CSDL Microsoft Office Access)

**Table:** SinhVien(MaSV, HoTen, Khoa, MaLop, Diem, DiaChi, GioiTinh, NgaySinh)

**Table:** Lop(MaLop, TenLop)

**Yêu cầu:** Hãy vẽ lược đồ cơ sở dữ liệu và viết câu SQL tạo cơ sở dữ liệu và thêm mỗi bảng 2 dòng dữ liệu.

Viết chương trình xử lý cho các nút sau:

- Button Load:** Nạp dữ liệu và Binding Data cho các control của Form.

2. Button Previous và Button Next: Di chuyển đến record trước và record kế tiếp.
3. Button Update: Lưu những thay đổi của record hiện tại xuống cơ sở dữ liệu.
4. Button Delete: Xóa dòng tại record hiện tại.
5. Button Add: Thêm một record mới (dữ liệu chỉ được nhập từ các TextBox, không thêm trực tiếp từ DataGrid).
6. Button Search: Tìm kiếm các record thỏa điều kiện được nhập vào TextBox Search và tìm kiếm theo trường (Field) được chọn trong ComboBox Search. ComboBox Search là danh sách gồm 3 trường sau: Họ tên, Lớp, Khoa.

**\* Chú ý:**

1. Khi Form vừa mới Load lên, chỉ có duy nhất nút Load được Enable.
2. Sau khi nút Load được nhấn, nút Load sẽ bị Disable, các nút khác sẽ được Enable.
3. Khi có sự thay đổi về Position của record (do nhấn nút Previous hay nút Next) hay số lượng record có sự thay đổi, label Position phải được cập nhật lại cho đúng.
4. Khi nhấn nút Add, tất cả các nút sẽ bị Disable, trừ nút Add và nút Update không bị Disable. Chương trình sẽ tự động thay đổi trường Text của nút Add để nó trở thành nút Cancel. Trong quá trình thêm dữ liệu vào, nếu nhấn nút Cancel (là nút Add sau khi đổi trường Text) thì sẽ ngưng quá trình thêm vào, nếu nhấn nút Update chương trình sẽ thêm 1 record mới vào database. Sau khi đã thêm 1 dòng mới xong, nút Cancel sẽ trở lại thành nút Add và tất cả các nút sẽ được Enable trở lại, ngoại trừ nút Load vẫn bị Disable.
5. Khi nhấn nút Search, chương trình sẽ dựa vào trường được chọn trong ComboBox Search để lọc ra những sinh viên thỏa điều kiện. (Trường Họ tên: tìm kiếm gần đúng – sử dụng từ khóa like; Trường Lớp và Khoa: Tìm kiếm chính xác).
6. Cơ sở dữ liệu sẽ được đóng khi form được đóng.

PHẦN 4. CƠ SỞ DỮ LIỆU VỚI LINQ

--- Hết ---