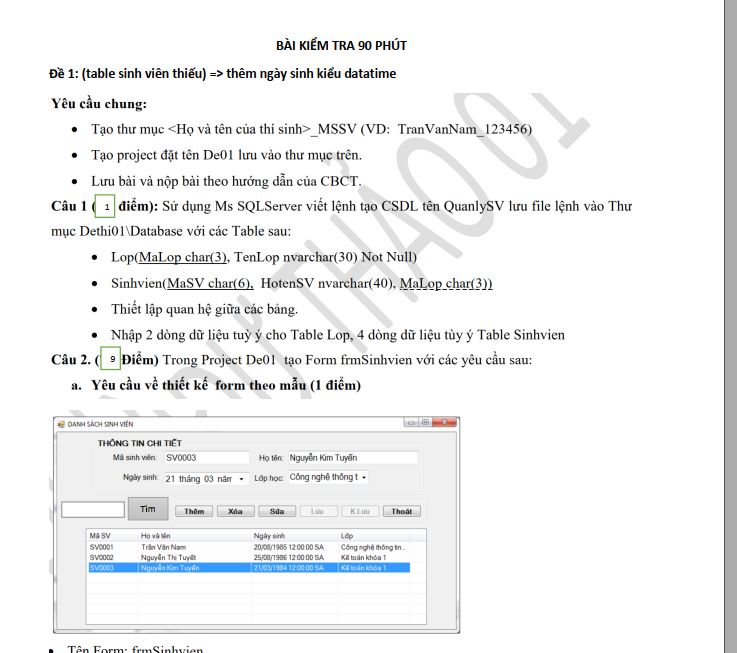
# Kiểm tra GK

Sinh viên thực hiện : Trịnh Tuấn Đan

MSSV: 2180602115 Lớp: 21DTHA4

Thời gian 120 phút.  
- Sinh viên chụp hình toàn bộ chức năng làm được và**tự chấm điểm** vào file word với cú pháp mssv\_hoten.docx  
- Có đường link github (hoặc một nền tảng lưu trữ trực tuyến khác) lưu source code trong file word trên.

Đề 1



|  |
| --- |
| using BUS;  using DAL.Entities;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Runtime.Remoting.Contexts;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace GUI  {      public partial class frmSinhVien : Form      {          private readonly StudentServices studentService = new StudentServices();          private readonly FacultyServices facultyService = new FacultyServices();          SinhVienContextDB context;          public frmSinhVien()          {              InitializeComponent();              context = new SinhVienContextDB();          }          // Hàm binding list có tên hiển thị là tên khoa, giá trị là Mã khoa - Phương thức FillFalcultyCombobox được sử dụng để đổ danh sách khoa vào combobox cmdKhoa.          private void FillFalcultyCombobox(List<Lop> listFalcultys)          {              listFalcultys.Insert(0, new Lop());              this.cboLop.DataSource = listFalcultys;              this.cboLop.ValueMember = "MaLop";              this.cboLop.DisplayMember = "TenLop";          }          //Phương thức BindGrid được sử dụng để hiển thị danh sách sinh viên lên DataGridView          public void BindGrid(List<SinhVien> listNhanVien)          {              dgvDSSV.Rows.Clear();              foreach (var item in listNhanVien)              {                  int index = dgvDSSV.Rows.Add();                  dgvDSSV.Rows[index].Cells[0].Value = item.MaSV;                  dgvDSSV.Rows[index].Cells[1].Value = item.HoTenSV;                  if (item.MaLop != null)                      dgvDSSV.Rows[index].Cells[2].Value = item.NgaySinh;                  dgvDSSV.Rows[index].Cells[3].Value = item.Lop.TenLop;              }          }            private void frmSinhVien\_Load(object sender, EventArgs e)          {                var listFacultys = facultyService.GetAll();              var listStudent = studentService.GetAll();              FillFalcultyCombobox(listFacultys);              BindGrid(listStudent);          }          private void dgvDSSV\_CellClick(object sender, DataGridViewCellEventArgs e)          {              try              {                  if (dgvDSSV.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)                  {                      dgvDSSV.CurrentRow.Selected = true;                      txtMaSV.Text = dgvDSSV.Rows[e.RowIndex].Cells[0].FormattedValue.ToString();                      txtHotenSV.Text = dgvDSSV.Rows[e.RowIndex].Cells[1].FormattedValue.ToString();                      dtNgaySinh.Value = Convert.ToDateTime(dgvDSSV.Rows[e.RowIndex].Cells[2].FormattedValue);                      cboLop.SelectedIndex = cboLop.FindString(dgvDSSV.Rows[e.RowIndex].Cells[3].FormattedValue.ToString());                  }              }              catch (Exception ex)              {                  MessageBox.Show(ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);              }          }          //--------------------------------------------------------------          private int GetSeclectedRow(string MSSV)          {              for (int i = 0; i < dgvDSSV.Rows.Count; i++)              {                  if (dgvDSSV.Rows[i].Cells[0].Value != null)                  {                      if (dgvDSSV.Rows[i].Cells[0].Value.ToString() == MSSV)                      {                          return i;                      }                  }              }              return -1;          }          private bool CheckValidate()          {              if (txtMaSV.Text == "" || txtHotenSV.Text == "")              {                  MessageBox.Show("Vui long Nhập Đây Đủ Thông Tin Nhân Viên!", "Thông Báo", MessageBoxButtons.OK);                  return false;              }              return true;          }          //Load lại thông tin trên dataGridview          private void reloadDGV()          {              var listStudents = studentService.GetAll();              BindGrid(listStudents);          }          // Refresh các ô nhập dữ liệu          private void clean()          {              txtMaSV.Text = "";              txtHotenSV.Text = "";              dtNgaySinh.Value = DateTime.Now;              cboLop.SelectedIndex = -1;          }          private void btThem\_Click(object sender, EventArgs e)          {              if (CheckValidate())              {                  string mssv = txtMaSV.Text.Trim();                  SinhVien existingStudent = context.SinhViens.FirstOrDefault(s => s.MaSV == mssv);                  if (existingStudent == null)                  {                      SinhVien newStudent = new SinhVien                      {                          MaSV = txtMaSV.Text,                          HoTenSV = txtHotenSV.Text,                          NgaySinh = dtNgaySinh.Value,                          MaLop = cboLop.SelectedValue.ToString()                      };                      context.SinhViens.Add(newStudent);                      context.SaveChanges();                      reloadDGV();                      clean();                      MessageBox.Show("Thêm mới dữ liệu thành công!", "Thông Báo", MessageBoxButtons.OK);                  }                  else                  {                      MessageBox.Show("Sinh viên đã tồn tại!", "Thông Báo", MessageBoxButtons.OK);                  }              }          }          private void btXoa\_Click(object sender, EventArgs e)          {              string mssv = txtMaSV.Text.Trim();              SinhVien studentToDelete = context.SinhViens.FirstOrDefault(s => s.MaSV == mssv);              if (studentToDelete != null)              {                  context.SinhViens.Remove(studentToDelete);                  context.SaveChanges();                  reloadDGV();                  clean();                  MessageBox.Show("Xóa Sinh Viên Thành Công!", "Thông Báo", MessageBoxButtons.OK);              }              else              {                  MessageBox.Show("Không tìm thấy Sinh viên cần xóa!", "Thông Báo", MessageBoxButtons.OK);              }          }          private void btSua\_Click(object sender, EventArgs e)          {              if (CheckValidate())              {                  string mssv = txtMaSV.Text.Trim();                  SinhVien studentToUpdate = context.SinhViens.FirstOrDefault(s => s.MaSV == mssv);                  if (studentToUpdate != null)                  {                      studentToUpdate.HoTenSV = txtHotenSV.Text;                      studentToUpdate.NgaySinh = dtNgaySinh.Value;                      studentToUpdate.MaLop = cboLop.SelectedValue.ToString();                      context.SaveChanges();                      reloadDGV();                      clean();                      MessageBox.Show("Cập nhật dữ liệu thành công!", "Thông Báo", MessageBoxButtons.OK);                  }                  else                  {                      MessageBox.Show("Không tìm thấy Sinh viên cần sửa!", "Thông Báo", MessageBoxButtons.OK);                  }              }          }          private void btThoat\_Click(object sender, EventArgs e)          {              if (MessageBox.Show("Bạn có muốn Thoát không", "Thông Báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)              {                  Application.Exit();              }          }      }  } |

Demo code

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

Link Source Code: https://github.com/TrinhTuandan/TrinhTuanDan\_218060211\_KiemTraGiuaKi5.git