**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**



**MÔN LẬP TRÌNH TRỰC QUAN**

**BÀI TẬP THỰC HÀNH 2**

GVHD: Nguyễn Ngọc Quí

Sinh viên thực hiện: Nguyễn Gia Hưng – 24520604

🙡🙢 Tp. Hồ Chí Minh, 10/2025 🙠🙣

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*……., ngày……...tháng……năm 2025*

**Người nhận xét**

*(Ký tên và ghi rõ họ tên****)***

MỤC LỤC

[Bài 1. Viết chương trình nhập vào tháng và năm, bạn in ra lịch của tháng đó như hình 6](#_Toc212075547)

[Bài 2. Viết chương trình yêu cầu người dùng nhập đường dẫn thư mục, xuất tất cả tên tập tin và thư mục con (tương tự lệnh DIR trong MS DOS) 9](#_Toc212075548)

[Bài 3. Xây dựng chương trình có chức năng 12](#_Toc212075549)

[Bài 4. Xây dựng lớp phân số với toán tử +, -, \*, /. Viết chương trình nhập vào hai phân số, xuất ra các giá trị là tổng, hiệu, tích và thương của hai phân số này. 22](#_Toc212075550)

[Bài 5. Công ty địa ốc Đại Phú muốn mở rộng kinh doanh thêm về thị trường Nhà phố và Chung Cư. Với tất cả các thông tin cần quản lý như sau: 29](#_Toc212075551)

DANH MỤC BẢNG

[Bảng 1.1. Nội dung code bài 1 6](#_Toc212075552)

[Bảng 1.2. Các testcase bài 1 7](#_Toc212075553)

[Bảng 2.1. Nội dung code bài 2 9](#_Toc212075554)

[Bảng 2.2. Các testcase bài 2 10](#_Toc212075555)

[Bảng 3.1. Nội dung code bài 3 12](#_Toc212075556)

[Bảng 3.2. Các testcase bài 3 16](#_Toc212075557)

[Bảng 4.1. Nội dung code bài 4 22](#_Toc212075558)

[Bảng 4.2. Các testcase bài 4 25](#_Toc212075559)

[Bảng 5.1. Nội dung code bài 5 29](#_Toc212075560)

[Bảng 5.2. Các testcase bài 5 34](#_Toc212075561)

DANH MỤC HÌNH ẢNH

[Hình 1.1. Testcase 1 bài 1 7](#_Toc212075562)

[Hình 1.2. Testcase 2 bài 1 7](#_Toc212075563)

[Hình 1.3. Testcase 3 bài 1 8](#_Toc212075564)

[Hình 1.4. Testcase 4 bài 1 8](#_Toc212075565)

[Hình 1.5. Testcase 5 bài 1 8](#_Toc212075566)

[Hình 2.1. Testcase 1 bài 2 10](#_Toc212075567)

[Hình 2.2. Testcase 2 bài 2 10](#_Toc212075568)

[Hình 2.3. Testcase 3 bài 2 11](#_Toc212075569)

[Hình 2.4. Testcase 4 bài 2 11](#_Toc212075570)

[Hình 3.1. Testcase 1 bài 3 16](#_Toc212075571)

[Hình 3.2. Testcase 2 bài 3 17](#_Toc212075572)

[Hình 3.3. Testcase 3 bài 3 17](#_Toc212075573)

[Hình 3.4. Testcase 4 bài 3 18](#_Toc212075574)

[Hình 3.5. Testcase 5 bài 3 18](#_Toc212075575)

[Hình 3.6. Testcase 6 bài 3 19](#_Toc212075576)

[Hình 3.7. Testcase 7 bài 3 20](#_Toc212075577)

[Hình 3.8. Testcase 8 bài 3 21](#_Toc212075578)

[Hình 4.1. Testcase 1 bài 4 25](#_Toc212075579)

[Hình 4.2. Testcase 2 bài 4 26](#_Toc212075580)

[Hình 4.3. Testcase 3 bài 4 27](#_Toc212075581)

[Hình 4.4. Testcase 4 bài 4 28](#_Toc212075582)

[Hình 5.1. Testcase 1 bài 5 (1) 34](#_Toc212075583)

[Hình 5.2. Testcase 1 bài 5 (2) 35](#_Toc212075584)

[Hình 5.3. Testcase 1 bài 5 (3) 35](#_Toc212075585)

NỘI DUNG BÀI LÀM

# Viết chương trình nhập vào tháng và năm, bạn in ra lịch của tháng đó như hình

## Mô tả đầu vào và đầu ra của bài toán:

* Input: Hai số nguyên là tháng và năm. Nhập lại nếu không hợp lệ
* Output: Các ngày trong tháng đó sắp xếp theo lịch, từ chủ nhật đến thứ bảy

## Nội dung code của Bài 1

Bảng .. Nội dung code bài 1

|  |
| --- |
| using System;  namespace Bai01  {  internal class Program  {  static void Main(string[] args)  {  //1. Nhập tháng và năm  int month;  do  {  month = NhapSoNguyen("Nhap thang: ");  if (month < 1 || month > 12)  {  Console.Write("Thang khong hop le, nhap lai thang: ");  }  else break;  }  while (true);  int year;  do  {  year = NhapSoNguyen("Nhap nam: ");  if (year < 1 || year > 9999)  {  Console.Write("Nam khong hop le, vui long nhap lai: ");  }  else break;  } while (true);  //2. In ra lịch  Console.WriteLine("====================================================");  Console.WriteLine("Sun\tMon\tTue\tWed\tThu\tFri\tSat");  Console.WriteLine("====================================================");  //Lấy số ngày trong tháng  int daysInMonth = DateTime.DaysInMonth(year, month);    //Xử lý ngày 1 của tháng  DateTime firstDate = new DateTime(year, month, 1);  int firstDayOfWeek = (int)firstDate.DayOfWeek;  for(int i = 0; i < firstDayOfWeek; i++)  {  Console.Write("\t");  }  Console.Write(1);  int currentDayOfWeek = firstDayOfWeek;  for(int day = 2; day <= daysInMonth; day++)  {  if (currentDayOfWeek == 6)  {  Console.Write("\n" + day);  currentDayOfWeek = 0;  }  else  {  Console.Write("\t" + day);  currentDayOfWeek++;  }  }  }  //Hàm nhập số nguyên  public static int NhapSoNguyen(string thongBao)  {  int value;  bool ok;  do  {  Console.Write(thongBao);  ok = int.TryParse(Console.ReadLine(), out value);  if (!ok)  {  Console.WriteLine("Gia tri khong hop le, vui long nhap lai!");  }  } while (!ok);  return value;  }  }  } |

## Các Testcase của chương trình:

Bảng .. Các testcase bài 1

|  |  |
| --- | --- |
| 1  2025 | Hình .. Testcase 1 bài 1 |
| 2  2020 | Hình .. Testcase 2 bài 1 |
| 2  2025 | Hình .. Testcase 3 bài 1 |
| 0  1  0  2000 | Hình .. Testcase 4 bài 1 |
| abc  12  abc  9999 | Hình .. Testcase 5 bài 1 |

# Viết chương trình yêu cầu người dùng nhập đường dẫn thư mục, xuất tất cả tên tập tin và thư mục con (tương tự lệnh DIR trong MS DOS)

## Mô tả đầu vào và đầu ra của bài toán:

* Input: Nhập vào một chuỗi là đường dẫn thư mục. Chọn 1 để tiếp tục nhập đường dẫn tư vị trí trước đó hoặc 0 để thoát chương trình.
* Output: In ra tất cả thư mục con, file trong thư mục, nếu không tìm thấy hoặc đường dẫn không hợp lệ thì in ra “Khong thay thu muc”

## Nội dung code của Bài 2:

Bảng .. Nội dung code bài 2

|  |
| --- |
| using System;  using System.IO;  namespace Bai02  {  internal class Program  {  static void Main(string[] args)  {  int isContinue;  string oldDir = "";  string mainDir = "";  do  {  Console.Write("Nhap duong dan thu muc: " + mainDir);  //Đường dẫn tương đối từ vị trí trước đó  string dir = Console.ReadLine();  //Cập nhật đường dẫn chính  if (!string.IsNullOrWhiteSpace(dir))  {  oldDir = mainDir;  if (dir[dir.Length - 1] == '/')  {  mainDir += dir;  }  else  {  mainDir += dir + "/";  }  }    //Nếu tìm thấy thư mục  if (Directory.Exists(mainDir))  {  DirectoryInfo di = new DirectoryInfo(mainDir);  DirectoryInfo[] subDirs = di.GetDirectories();  FileInfo[] files = di.GetFiles();  foreach (DirectoryInfo dirInfo in subDirs)  {  Console.WriteLine("{0}\t<DIR>\t{1}",  dirInfo.LastWriteTime.ToString("dd/MM/yyyy HH:mm"),  dirInfo.Name);  }  foreach (FileInfo file in files)  {  Console.WriteLine("{0}\t{1}\t{2}",  file.LastWriteTime.ToString("dd/MM/yyyy HH:mm"),  file.Length,  file.Name);  }  }  else  {  Console.WriteLine("Khong thay thu muc");  mainDir = oldDir;  }  //Tiếp tục tìm hoặc thoát chương trình  Console.WriteLine("Tiep tuc?\n" +  "[1]: Yes\t[0]: No");  isContinue = int.Parse(Console.ReadLine());  }  while (isContinue > 0);  }  }  } |

## Các Testcase của chương trình:

Bảng .. Các testcase bài 2

|  |  |
| --- | --- |
| D  1  D:  0 | Hình .. Testcase 1 bài 2 |
| D:/Hung/IT008  1  BTH2\_NguyenGiaHung\_24520604  0 | Hình .. Testcase 2 bài 2 |
| D:/Hung  1  IT001  1  IT002  1  IT004  0 | Hình .. Testcase 3 bài 2 |
| D:/Hung/IT008  1  ..  1  IT001  1  IT008  0 | Hình .. Testcase 4 bài 2 |

# Xây dựng chương trình có chức năng

**a. Nhập / xuất ma trận hai chiều các số nguyên**

**b. Tìm kiếm một phần tử trong ma trận**

**c. Xuất các phần tử là số nguyên tố**

**d. Cho biết dòng nào có nhiều số nguyên tố nhất**

## Mô tả đầu vào và đầu ra của bài toán:

* Input: Nhập vào hai số nguyên là số dòng m và số cột n của ma trận. Chọn 1 để tạo tự động ma trận ngẫu nhiên hoặc chọn 0 và nhập mn số nguyên để tạo thành ma trận. Nhập số k cần tìm trong ma trận.
* Output: In ra ma trận đã tạo, in ra vị trí phần tử k cần tìm, in ra các phân tử là số nguyên tố, in ra dòng chứa nhiều số nguyên tố nhất và in ra dòng đó

## Nội dung code của Bài 3:

Bảng .. Nội dung code bài 3

|  |
| --- |
| using System;  using System.Collections.Generic;  namespace Bai03  {  internal class Program  {  static void Main(string[] args)  {  //1. Tạo, nhập và xuất ma trận  MaTran mat = new MaTran();  mat.Nhap();  mat.Print();  //2. Tìm số nguyên trong ma trận  int iToFind = NhapSoNguyen("\n2. Nhap so can tim: ");  mat.FindFirstIf((x) => (x == iToFind)).Print();  //3. Tìm các phần tử là số nguyên tố  Console.WriteLine("\n3. Cac phan tu la so nguyen to: ");  mat.Print((x) => IsPrime(x));  //4. Tìm các dòng có nhiều số nguyên tố nhất  List<int> rows = mat.MaxRowCountIf((x) => IsPrime(x));  Console.WriteLine("\n4. Dong co so nguyen to nhieu nhat:");  foreach (int r in rows)  {  Console.Write("- Dong " + (r + 1) + ":\t");  mat.PrintRow(r);  }  }    public delegate bool DieuKien(int i);  public class MaTran  {  private List<List<int>> mat;  public int Row  {  get  {  return mat.Count;  }  }  public int Col  {  get  {  if (Row != 0)  {  return mat[0].Count;  }  else return 0;  }  }  public MaTran(int r = 1, int c = 1, bool random = true, int maxVal = 100, int minVal = -100)  {  CreateMatrix(r, c, random, maxVal, minVal);  }  //Tạo ma trận  private void CreateMatrix(int r = 1, int c = 1, bool random = true, int maxVal = 100, int minVal = -100)  {  mat = new List<List<int>>();  if (r <= 0 || c < 1) return;  Random rand = new Random();  for (int i = 0; i < r; i++)  {  List<int> row = new List<int>();  for (int j = 0; j < c; j++)  {  row.Add(random ? rand.Next(minVal, maxVal + 1) : 0);  }  mat.Add(row);  }  }  public void Nhap()  {  Console.WriteLine("Tao ma tran:");  int dong = NhapSoNguyen("Dong: ");  int cot = NhapSoNguyen("Cot: ");  int choice = NhapSoNguyen("[1]: Tao tu dong\t"  + "[0]: Tao thu cong\n");  if (choice > 0)  {  CreateMatrix(dong, cot);  }  else  {  CreateMatrix(dong, cot, false);  for (int r = 0; r < dong; r++)  {  for (int c = 0; c < cot; c++)  {  Console.Write("Nhap [{0}][{1}]: ", r, c);  mat[r][c] = NhapSoNguyen("");  }  }  }  }  //In ma trận  public void Print()  {  for (int r = 0; r < Row; r++)  {  PrintRow(r);  }  }  //In dòng r trong ma trận  public void PrintRow(int r)  {  if (r < 0 || r >= Row) return;  foreach (int i in mat[r])  {  Console.Write(i + "\t");  }  Console.WriteLine();  }  //In ra các số thỏa điều kiện dk  public void Print(DieuKien dk)  {  foreach (var row in mat)  {  foreach (int i in row)  {  if (dk(i))  {  Console.Write(i + " ");  }  }  }  Console.WriteLine();  }  //Vị trị trong ma trận  public struct Position  {  public int row { get; private set; }  public int col { get; private set; }  public Position(int r, int c)  {  row = r; col = c;  }  public void Print()  {  Console.WriteLine("Dong {0} - cot {1}", row + 1, col + 1);  }  }  //Giá giá trị một số trong vị trí  public int GetValue(Position pos)  {  if (pos.row < 0 || pos.col < 0  || pos.row >= Row || pos.col >= Col)  {  return -1;  }  return mat[pos.row][pos.col];  }  //Vị trí đầu tiên nếu tìm thấy số thỏa điều kiện dk  public Position FindFirstIf(DieuKien dk)  {  for (int r = 0; r < Row; r++)  {  for (int c = 0; c < Col; c++)  {  if (dk(mat[r][c]))  {  return new Position(r, c);  }  }  }  Console.Write("Khong tim thay: ");  return new Position(-1, -1);  }  //Những dòng có nhiều số nhất thỏa điều kiện dk  public List<int> MaxRowCountIf(DieuKien dk)  {  List<int> result = new List<int>();  int maxCount = 0;  for (int r = 0; r < Row; r++)  {  int countRow = 0;  for (int c = 0; c < Col; c++)  {  if (dk(mat[r][c])) countRow++;  }  if (countRow > maxCount)  {  maxCount = countRow;  result.Clear();  result.Add(r);  }  else if (countRow == maxCount && countRow > 0)  {  result.Add(r);  }  }  return result;  }  }  //Nhập số nguyên  public static int NhapSoNguyen(string thongBao)  {  int value;  bool ok;  do  {  Console.Write(thongBao);  ok = int.TryParse(Console.ReadLine(), out value);  if (!ok)  {  Console.WriteLine("Gia tri khong hop le, vui long nhap lai!");  }  } while (!ok);  return value;  }  //Kiểm tra số nguyên tố  public static bool IsPrime(int x)  {  if (x < 2) return false;  int root = (int)Math.Sqrt(x);  for(int i = 2; i <= root; i++)  {  if (x % i == 0) return false;  }  return true;  }  }  } |

## Các Testcase của chương trình:

Bảng .. Các testcase bài 3

|  |  |
| --- | --- |
| 2  2  1  abc  1 | Hình .. Testcase 1 bài 3 |
| 2  3  0  2  4  5  7  8  9  5 | Hình .. Testcase 2 bài 3 |
| 2  2  0  4  6  8  9  8 | Hình .. Testcase 3 bài 3 |
| 2  4  0  -3  1  2  3  5  10  11  12  11 | Hình .. Testcase 4 bài 3 |
| 1  1  0  7  7 | Hình .. Testcase 5 bài 3 |
| 3  3  1  5 | Hình .. Testcase 6 bài 3 |
| 3  4  0  2  3  4  5  6  7  8  9  10  11  13  17  13 | Hình .. Testcase 7 bài 3 |
| 2  3  0  2  3  5  7  11  13  9 | Hình .. Testcase 8 bài 3 |

# Xây dựng lớp phân số với toán tử +, -, \*, /. Viết chương trình nhập vào hai phân số, xuất ra các giá trị là tổng, hiệu, tích và thương của hai phân số này.

**Viết chương trình cho phép nhập vào một dãy các phân số:**

**- Tìm phần số lớn nhất**

**- Sắp xếp các phân số trong dãy tăng dần**

## Mô tả đầu vào và đầu ra của bài toán:

* Input: Nhập tử và mẫu hai phân số đầu tiên. Nhập số n và lần lượt nhập tử và mẫu của n phân số tiếp theo
* Output: In ra tổng, hiệu, tích, thương của hai phân số đầu tiên

## Nội dung code của Bài 4:

Bảng .. Nội dung code bài 4

|  |
| --- |
| using System;  using System.Collections.Generic;  namespace Bai04  {  internal class Program  {  static void Main(string[] args)  {  //1. Nhập hai phân số  Console.WriteLine("1. Nhap hai phan so: ");  PhanSo a = new PhanSo();  PhanSo b = new PhanSo();  a.Nhap();  b.Nhap();  Console.WriteLine();  Console.WriteLine($"Tong: {a} + {b} = {a + b}");  Console.WriteLine($"Hieu: {a} - {b} = {a - b}");  Console.WriteLine($"Tich: {a} \* {b} = {a \* b}");  try  {  Console.WriteLine($"Thuong: {a} / {b} = {a / b}");  }  catch (DivideByZeroException ex)  {  Console.WriteLine("Khong chia duoc cho 0 !");  }  int n = NhapSoNguyen("\n2. Nhap so luong phan so: ");  List<PhanSo> list = new List<PhanSo>();  for (int i = 0; i < n; i++)  {  Console.WriteLine($"Phan so {i + 1}:");  PhanSo newPhanSo = new PhanSo();  newPhanSo.Nhap();  list.Add(newPhanSo);  }  list.Sort();  Console.WriteLine($"\nPhan so lon nhat: {list[n - 1]}");  Console.Write("Day phan so da sap xep: ");  foreach (PhanSo ps in list)  {  Console.Write(ps + " ");  }  }  public class PhanSo : IComparable  {  public int Tu { get; set; }  private int mau;  public int Mau  {  get => mau;  set  {  if (value == 0)  throw new DivideByZeroException("Mau khac 0 !");  mau = value;  }  }  public PhanSo(int t = 0, int m = 1)  {  Tu = t;  try  {  Mau = m;  }  catch (DivideByZeroException ex)  {  Console.WriteLine(ex.Message);  Mau = 1;  }  RutGon();  }  public void Nhap()  {  Tu = NhapSoNguyen("Tu: ");  bool hopLe = false;  while (!hopLe)  {  try  {  Mau = NhapSoNguyen("Mau: ");  hopLe = true;  }  catch (DivideByZeroException ex)  {  Console.WriteLine(ex.Message);  }  }  RutGon();  }  private static int GCD(int a, int b)  {  a = Math.Abs(a);  b = Math.Abs(b);  while (b != 0)  {  int temp = b;  b = a % b;  a = temp;  }  return a;  }  private void RutGon()  {  if (Tu == 0)  {  Mau = 1;  return;  }  int gcd = GCD(Tu, Mau);  Tu /= gcd;  Mau /= gcd;  if (Mau < 0)  {  Tu = -Tu;  Mau = -Mau;  }  }  public override string ToString()  {  if (Mau == 1)  return $"{Tu}";  return $"{Tu}/{Mau}";  }  // Toán tử  public static PhanSo operator +(PhanSo a, PhanSo b)  {  return new PhanSo(a.Tu \* b.Mau + b.Tu \* a.Mau, a.Mau \* b.Mau);  }  public static PhanSo operator -(PhanSo a, PhanSo b)  {  return new PhanSo(a.Tu \* b.Mau - b.Tu \* a.Mau, a.Mau \* b.Mau);  }  public static PhanSo operator \*(PhanSo a, PhanSo b)  {  return new PhanSo(a.Tu \* b.Tu, a.Mau \* b.Mau);  }  public static PhanSo operator /(PhanSo a, PhanSo b)  {  if (b.Tu == 0)  throw new DivideByZeroException("Khong the chia cho phan so bang 0!");  return new PhanSo(a.Tu \* b.Mau, a.Mau \* b.Tu);  }  public int CompareTo(object obj)  {  if (obj is PhanSo p)  {  double x = (double)Tu / Mau;  double y = (double)p.Tu / p.Mau;  return x.CompareTo(y);  }  throw new ArgumentException("Doi tuong khong phai PhanSo!");  }  }  public static int NhapSoNguyen(string thongBao)  {  int value;  bool ok;  do  {  Console.Write(thongBao);  ok = int.TryParse(Console.ReadLine(), out value);  if (!ok)  Console.WriteLine("Gia tri khong hop le, vui long nhap lai!");  } while (!ok);  return value;  }  }  } |

## Các Testcase của chương trình:

Bảng .. Các testcase bài 4

|  |  |
| --- | --- |
| 1  2  -3  4  7  0  5  2  4  -3  -6  3  2  -5  7  5  -7  4  8 | Hình .. Testcase 1 bài 4 |
| 0  5  1  1  5  0  1  1  1  -1  1  2  1  -3  1 | Hình .. Testcase 2 bài 4 |
| 1  0  3  2  3  4  1  0  2  3  4  5  0  6  7  8 | Hình .. Testcase 3 bài 4 |
| 1  2  0  5  3  1  2  3  4  5  6 | Hình .. Testcase 4 bài 4 |

# Công ty địa ốc Đại Phú muốn mở rộng kinh doanh thêm về thị trường Nhà phố và Chung Cư. Với tất cả các thông tin cần quản lý như sau:

**- Khu Đất: Địa Điểm, Giá Bán (Đơn vị tính: VND) và Diện Tích (m2)**

**- Nhà Phố: Địa Điểm, Giá Bán (Đơn vị tính: VND), Diện tích (m2), Năm Xây dựng, Số tầng**

**- Chung Cư: Địa Điểm, Giá Bán (Đơn Vị Tính: VND), Diện Tích (m2), Tầng**

**Xây dựng chương trình thực hiện các yêu cầu sau:**

**- Xây dựng các lớp với chức năng thừa kế**

**- Nhập xuất danh sách các thông tin (Khu đất, Nhà phố, Chung Cư) cần quản lý.**

**- Xuất tổng giá bán cho 3 loại (Khu đất, Nhà phố, Chung Cư) của công ty Đại Phú.**

**- Xuất danh sách các khu đất có diện tích > 100m2 hoặc là nhà phố mà có diện tích >60m2 và năm xây dựng >= 2019 (nếu có).**

**- Nhập vào các thông tin cần tìm kiếm (địa điểm, giá, diện tích). Xuất thông tin danh sách tất cả các nhà phố hoặc chung cư phù hợp yêu cầu. (có địa điểm chứa chuỗi tìm kiếm không phân biệt hoa thường, có giá <= giá tìm kiếm, và diện tích >= diện tích cần tìm kiếm)**

## Mô tả đầu vào và đầu ra của bài toán:

* Input: Chọn 1, 2 hoặc 3 để nhập khu đất, nhà phố hoặc chung cư, chọn 0 để thoát không nhập danh sách. Nhập thông tin cho bất động sản để tạo danh sách. Nhập thông tin cần tìm kiếm gồm địa điểm, giá, diện tích
* Output: In ra thông tin các bất động sản đã nhập,

## Nội dung code của Bài 5:

Bảng .. Nội dung code bài 5

|  |
| --- |
| using System;  using System.Collections.Generic;  namespace Bai05  {  internal class Program  {  public delegate bool DieuKieu(KhuDat k);  static void Main(string[] args)  {  //1. Nhập danh sách  Console.WriteLine("1. Nhap danh sach");  CongTyDaiPhu cty = new CongTyDaiPhu();  cty.Nhap();  //2. Xuất danh sách  Console.WriteLine("\n2. Xuat danh sach\n2.1. Khu dat: ");  cty.Xuat(KhuDat.LOAI.KhuDat);  Console.WriteLine("2.2. Nha pho: ");  cty.Xuat(KhuDat.LOAI.NhaPho);  Console.WriteLine("2.3. Chung cu:");  cty.Xuat(KhuDat.LOAI.ChungCu);  //3. Tổng giá bán 3 loại  Console.WriteLine("\n3. Tong gia ban:" +  "\n- Khu dat: " + cty.GiaBan(KhuDat.LOAI.KhuDat) +  "\n- Nha pho: " + cty.GiaBan(KhuDat.LOAI.NhaPho) +  "\n- Chung cu: " + cty.GiaBan(KhuDat.LOAI.ChungCu));  //4. Xuất danh sách theo điều kiện  Console.WriteLine("\n4. Xuat danh sach" +  "\n4.1. Khu dat > 100:");  cty.Xuat(khu => (khu.loai == KhuDat.LOAI.KhuDat && khu.DienTich > 100));  Console.WriteLine("\n4.2. Nha pho dien tich > 60, nam > 2019");  cty.Xuat(khu =>  {  if (khu is NhaPho nhaPho)  {  return nhaPho.DienTich > 60 && nhaPho.NamXayDung >= 2019;  }  return false;  });  //5. Tìm kiếm nhà phố hoặc chung cư  Console.WriteLine("\n5. Tim kiem nha pho hoac chung cu");  cty.Find();  }  public class CongTyDaiPhu  {  List<KhuDat> list;  public CongTyDaiPhu()  {  list = new List<KhuDat>();  }  public void Nhap()  {  Console.WriteLine("NHAP DANH SACH BAT DONG SAN: ");  do  {  KhuDat newKhuDat;  int choice = NhapSoNguyen("[0]: Thoat\t[1]: Khu dat\t[2]: Nha pho\t[3]: Chung cu\n");  if (choice == 0) break;  switch (choice)  {  case 2: newKhuDat = new NhaPho(); break;  case 3: newKhuDat = new ChungCu(); break;  default: newKhuDat = new KhuDat(); break;  }  newKhuDat.Nhap();  list.Add(newKhuDat);  }  while (true);  }  public void Xuat(KhuDat.LOAI loai)  {  bool has = false;  foreach (KhuDat khuDat in list)  {  if (loai == KhuDat.LOAI.All || khuDat.loai == loai)  {  Console.WriteLine(khuDat);  has = true;  }  }  if (!has)  {  Console.WriteLine("Khong co ket qua!");  }  }  //Tìm khu đất thỏa điều kiện dk  public void Xuat(DieuKieu dk)  {  bool has = false;  foreach(KhuDat khuDat in list)  {  if (dk(khuDat))  {  Console.WriteLine(khuDat);  has = true;  }  }  if (!has)  {  Console.WriteLine("Khong co ket qua!");  }  }  //Lấy giá bán theo loại bất động sản  public double GiaBan(KhuDat.LOAI loai)  {  double gia = 0;  foreach (KhuDat khudat in list)  {  if (loai == KhuDat.LOAI.All || khudat.loai == loai)  {  gia += khudat.GiaBan;  }  }  return gia;  }  public void Find()  {  Console.Write("+ Dia diem: ");  string diadiem = Console.ReadLine()?.Trim() ?? string.Empty;  double gia = NhapSoThuc("+ Gia ban: ");  double dienTich = NhapSoThuc("+ Dien tich: ");  List<KhuDat> result = new List<KhuDat>();  foreach (var khuDat in list)  {  if (khuDat.loai == KhuDat.LOAI.KhuDat) continue;  if (string.IsNullOrWhiteSpace(diadiem) || string.IsNullOrWhiteSpace(khuDat.DiaDiem)) continue;  if (khuDat.DiaDiem.IndexOf(diadiem, StringComparison.OrdinalIgnoreCase) < 0) continue;  if (khuDat.GiaBan > gia) continue;  if (khuDat.DienTich < dienTich) continue;  result.Add(khuDat);  }  if (result.Count > 0)  {  Console.WriteLine("KET QUA TIM KIEM: ");  foreach (KhuDat khu in result)  {  Console.WriteLine(khu);  }  }  else  {  Console.WriteLine("KHONG TIM THAY KET QUA");  }  }  }  public class KhuDat  {  public string DiaDiem { get; protected set; }  public double GiaBan { get; protected set; }  public double DienTich { get; protected set; }  public enum LOAI  {  KhuDat,  NhaPho,  ChungCu,  All  }  public LOAI loai { get; protected set; }  public KhuDat()  {  DiaDiem = string.Empty;  GiaBan = 0;  DienTich = 0;  loai = LOAI.KhuDat;  }  public virtual void Nhap()  {  Console.WriteLine("NHAP KHU DAT:");  NhapThongTinCoBan();  }  protected void NhapThongTinCoBan()  {  Console.Write("- Dia diem: ");  DiaDiem = Console.ReadLine();  GiaBan = NhapSoThuc("- Gia ban: ");  DienTich = NhapSoThuc("- Dien tich: ");  }  protected virtual string ThongTinCoBan()  {  return "\n- Dia diem: " + DiaDiem +  "\n- Gia ban: " + GiaBan.ToString() +  "\n- Dien tich: " + DienTich.ToString();  }  public override string ToString()  {  return "KHU DAT:" +  ThongTinCoBan();  }  }  public class NhaPho : KhuDat  {  public int NamXayDung { get; protected set; }  int soTang;  public NhaPho() : base()  {  NamXayDung = 0;  soTang = 0;  loai = LOAI.NhaPho;  }  public override void Nhap()  {  Console.WriteLine("NHAP NHA PHO:");  base.NhapThongTinCoBan();  NamXayDung = NhapSoNguyen("- Nam xay dung: ");  soTang = NhapSoNguyen("- So tang: ");  }  public override string ToString()  {  return "NHA PHO: " + ThongTinCoBan() +  "\n- Nam xay dung: " + NamXayDung.ToString() +  "\n- So tang: " + soTang.ToString();  }  }  public class ChungCu : KhuDat  {  int tang;  public ChungCu() : base()  {  tang = 0;  loai = LOAI.ChungCu;  }  public override void Nhap()  {  Console.WriteLine("NHAP CHUNG CU:");  base.NhapThongTinCoBan();  tang = NhapSoNguyen("- So tang: ");  }  public override string ToString()  {  return "CHUNG CU: " + ThongTinCoBan() +  "\n- So tang: " + tang.ToString();  }  }  public static int NhapSoNguyen(string thongBao)  {  int value;  bool ok;  do  {  Console.Write(thongBao);  ok = int.TryParse(Console.ReadLine(), out value);  if (!ok)  {  Console.WriteLine("Gia tri khong hop le, vui long nhap lai!");  }  } while (!ok);  return value;  }  public static double NhapSoThuc(string prompt)  {  double value;  bool isSuccess;  do  {  Console.Write(prompt);  isSuccess = double.TryParse(Console.ReadLine(), out value);  if (!isSuccess)  {  Console.WriteLine("Gia tri khong hop le, vui long nhap so thuc!");  }  } while (!isSuccess);  return value;  }  }  } |

## Các Testcase của chương trình:

Bảng .. Các testcase bài 5

|  |  |
| --- | --- |
| 1  Thu Duc  2500000000  120  2  Binh Thanh  4800000000  80  2020  3  3  Quan 7  3200000000  70  15  2  Thu Duc  3600000000  55  2018  2  0  duc  400000000000  50 | Hình .. Testcase 1 bài 5 (1)    Hình .. Testcase 1 bài 5 (2)    Hình .. Testcase 1 bài 5 (3) |

LinkCode: <https://github.com/nguyenhun11/IT008.Q14_BTH2_NguyenGiaHung_24520604>