**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 TUẦN 2**

GVHD: Nguyễn Ngọc Quí

Sinh viên thực hiện: Phạm Đan Trường

MSSV: 24521898

🙡🙢 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**

[1. Viết chương trình nhập vào tháng và năm, bạn in ra lịch của tháng đó như hình yêu cầu. 6](#_Toc212711388)

[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). 10](#_Toc212711389)

[3. Xây dựng chương trình có các chức năng: 14](#_Toc212711390)

**[a. Nhập / xuất ma trận hai chiều các số nguyên 14](#Demand)**

**[b. Tìm kiếm một phần tử trong ma trận 14](#Demand)**

**[c. Xuất các phần tử là số nguyên tố 14](#Demand)**

**[d. Cho biết dòng nào có nhiều số nguyên tố nhất 14](#Demand)**

[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. 24](#_Toc212711391)

[5. Xây dựng phần mềm quản lý cho Công ty địa ốc Đại Phú với các yêu cầu sau: 37](#_Toc212711392)

**DANH MỤC BẢNG**

[**Bảng 1. 1: Nội dung code của chương trình bài 1 6**](#_Toc212723602)

[**Bảng 1. 2: Các Testcase của chương trình bài 1 7**](#_Toc212723603)

[**Bảng 2. 1: Nội dung code của chương trình bài 2 10**](#_Toc212724112)

[**Bảng 2. 2: Các Testcase của chương trình bài 2 11**](#_Toc212724113)

[**Bảng 3. 1: Nội dung code của chương trình bài 3 15**](#_Toc212724119)

[**Bảng 3. 2: Các Testcase của chương trình bài 3 19**](#_Toc212724120)

[**Bảng 4. 1. 1: Nội dung code của file PhanSo.cs 25**](#_Toc212725682)

[**Bảng 4. 1. 2: Nội dung code của file Program.cs 27**](#_Toc212725683)

[**Bảng 4. 2: Các Testcase của chương trình bài 4 31**](#_Toc212725684)

[**Bảng 5. 1. 1: Nội dung code của file program.cs 37**](#_Toc212725093)

[**Bảng 5. 1. 2: Nội dung code của file ThiTruong.cs 41**](#_Toc212725094)

[**Bảng 5. 1. 3: Nội dung code của file KhuDat.cs 42**](#_Toc212725095)

[**Bảng 5. 1. 4: Nội dung code của file NhaPho.cs 43**](#_Toc212725096)

[**Bảng 5. 1. 5: Nội dung code của file ChungCu.cs 45**](#_Toc212725097)

[**Bảng 5. 2: Các Testcase của chương trình bài 5**](#Bang_5) **46**

**DANH MỤC HÌNH ẢNH**

**NỘI DUNG BÀI LÀM**

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

Mô tả đầu vào và đầu ra của bài toán:

* Input: 2 số nguyên dương lần lượt là tháng và năm.
* Output: In ra lịch của tháng đó theo yêu cầu.

Nội dung code của bài 1:

Bảng 1. : Nội dung code của chương trình bài 1

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Collections.Specialized;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Bai01  {  class Program  {  static void Main(string[] args)  {  int month, year;  Console.Write("Nhap thang: ");  while(true)  {  try  {  month = Convert.ToInt32(Console.ReadLine());  break;  }  catch  {  Console.Write("Loi dinh dang. Nhap lai: ");  }  }  Console.Write("Nhap nam: ");  while (true)  {  try  {  year = Convert.ToInt32(Console.ReadLine());  break;  }  catch  {  Console.Write("Loi dinh dang. Nhap lai: ");  }  }  if (month<=0 || year<=0 || month >12)  {  Console.WriteLine("Thang khong hop le. Khong ton tai thang vua nhap.");  return;  }  DateTime current = new DateTime(year, month, 1);  int DayInMonth = DateTime.DaysInMonth(year, month); // Số ngày trong tháng  int StartDay = (int)current.DayOfWeek; // Ngày bắt đầu là thứ mấy  int day = 0;  int now = 1;  Console.WriteLine("Month: " + month.ToString("00") + "/" + year.ToString("0000"));  Console.WriteLine("Sun Mon Tue Wed Thu Fri Sat");  while (now <= DayInMonth)  {  if (day < StartDay) Console.Write(" ");  else  {  if (day % 7 == 0) Console.WriteLine();  Console.Write("{0,3}", now);  Console.Write(" ");  now++;  }  day++;  }  }    }  } |

Các Testcase của chương trình:

Bảng 1. : Các Testcase của chương trình bài 1

|  |  |
| --- | --- |
| Input  Output | 6 2028 |
| Input  Output | 20 2025 |
| Input  Output | j -10 2026 |
| Input  Output | 12 1975 |
| Input  Output | 2 2000 |
| Input  Output | 3 -120 |
| Input  Output | 5 1 |
| Input  Output | abcd 4  Pham Dan Truong  1986 |
| Input  Output | 7 2006 |

# 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).

Mô tả đầu vào và đầu ra của bài toán:

* Input: Một chuỗi duy nhất là đường dẫn cần xem.
* Output: Tất cả tên tệp tin và thư mục con trong đường dẫn đó.

Nội dung code của chương trình bài 2:

Bảng 2. : Nội dung code của chương trình bài 2

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.IO;  namespace Bai02  {  class Program  {  static void ShowFileAndDirectory(string path)  {  try  {  string[] directory = Directory.GetDirectories(path);  string[] file = Directory.GetFiles(path);  foreach (string s in file)  {  Console.WriteLine(s);  }  foreach (string s in directory)  {  Console.WriteLine(s);  ShowFileAndDirectory(s);  }  }  catch  {  Console.WriteLine("Khong the truy cap.");  }  }  static void Main(string[] args)  {  string path;  Console.Write("Nhap duong dan thu muc ban muon: ");  while(true)  {  path = Console.ReadLine();  if (Directory.Exists(path))  {  Console.WriteLine("Danh sach cac thu muc con va tap tin: ");  ShowFileAndDirectory(path);  break;  }  Console.Write("Duong dan khong ton tai. Moi nhap lai: ");  }  }  }  } |

Các Testcase của chương trình:

Bảng 2. : Các Testcase của chương trình bài 2

|  |  |
| --- | --- |
| Input  Output | E:\Truongpham-code\DoAn\_IT008\Class\Bao Cao |
| Input  Output | C:\C++ |
| Input  Output | adshfasfdsf  E:\24521898 |
| Input  Output | E:\New\_Download |
| Input  Output | C:\System Volume Information |
| Input  Output | Linux  PhamDanTruong\_24521898  E:\Riot Games |
| Input  Output | D:\TruongPham\IT008 |

# 3. Xây dựng chương trình có các chức năng:

**a. Nhập/ Xuất ma trận 2 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: Một số nguyên là lựa chọn trong các danh sách việc muốn thực hiện với:

+ 0: Nhập dữ liệu cho ma trận. Trong phần này sẽ nhập thêm 2 số nguyên dương tương ứng lần lượt là số dòng và số cột của ma trân. Tiếp theo đó n dòng, mỗi dòng sẽ chứa m số nguyên là các phần tử trong ma trận.

+ 1: Xuất ma trận.

+ 2: Tìm kiếm một phần tử trong ma trận.

+ 3: Xuất các phần tử là số nguyên tố.

+ 4: Chỉ số của dòng có nhiều số nguyên tố nhất.

+ 5: Thoát các sự lựa chọn.

* Output: Kết quả xuất ra màn hình sẽ dựa trên lựa chọn từ phần nhập.

Nội dung code của bài 3:

Bảng 3. : Nội dung code của chương trình bài 3

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Runtime.Remoting.Metadata;  using System.Text;  using System.Threading.Tasks;  namespace Bai03  {  class Program  {  static void Main(string[] args)  {  Console.WriteLine("===============Menu===============");  Console.WriteLine("0: Nhap du lieu cho ma tran.");  Console.WriteLine("1: Xuat ma tran.");  Console.WriteLine("2: Tim kiem mot phan tu trong ma tran.");  Console.WriteLine("3: Xuat cac phan tu la so nguyen to.");  Console.WriteLine("4: Chi so cua dong co nhieu so nguyen to nhat.");  Console.WriteLine("5: Thoat.");  Console.WriteLine("==================================");  int choice;  int n = 0, m = 0;  int[,] matrix = new int[n, m];  while (true)  {  Console.Write("Nhap lua chon cua ban: ");  while (true)  {  try  {  choice = Convert.ToInt32(Console.ReadLine());  break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  if (choice == 0)  {  MatrixInput(ref matrix, ref n, ref m);  }  else if (choice == 1)  {  MatrixOutput(matrix, n, m);  }  else if (choice == 2)  {  FindElementInMatrix(matrix, n, m);  }  else if (choice == 3)  {  PrimeElement(matrix, n, m);  }  else if (choice == 4)  {  RowWithMostPrime(matrix, n, m);  }  else if (choice == 5)  {  break;  }  else Console.WriteLine("Lua chon khong hop le.");  }  }  // Nhap du lieu cho ma tran  static void MatrixInput(ref int[,] matrix, ref int n, ref int m)  {  Console.Write("Nhap so dong: ");  while (true)  {  try  {  n = Convert.ToInt32(Console.ReadLine());  if (n <= 0) Console.Write("So dong khong hop le. Moi nhap lai: ");  else break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  Console.Write("Nhap so cot: ");  while (true)  {  try  {  m = Convert.ToInt32(Console.ReadLine());  if (m <= 0) Console.Write("So cot khong hop le. Moi nhap lai: ");  else break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  matrix = new int[n, m];  Console.WriteLine("Nhap du lieu cho ma tran: ");  for (int i=0;i<n;i++)  {  string s = Console.ReadLine();  int[] array = s.Split(' ').Select(x => Convert.ToInt32(x)).ToArray();  for (int j=0;j<m;j++)  {  matrix[i,j] = array[j];  }  }  }  // Xuat du lieu cho ma tran  static void MatrixOutput(int[,] matrix, int n, int m)  {  if (n==0 && m==0)  {  Console.WriteLine("Ma tran rong.");  return;  }  Console.WriteLine("Ma tran: ");  for (int i=0;i<n;i++)  {  for (int j=0;j<m;j++)  {  Console.Write(matrix[i, j] + " ");  }  Console.WriteLine();  }  }  // Tim kiem mot phan tu trong ma tran  static void FindElementInMatrix(int[,] matrix, int n, int m)  {  if (n==0 && m==0)  {  Console.WriteLine("Ma tran rong.");  return;  }  int k;  Console.Write("Nhap phan tu ban muon tim: ");  while (true)  {  try  {  k = Convert.ToInt32(Console.ReadLine());  break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  int row =-1, col=-1;  for (int i=0;i<n;i++)  {  for (int j=0;j<m;j++)  {  if (matrix[i,j] == k)  {  row = i + 1;  col = j + 1;  break;  }  }  }  if (row==-1 && col==-1)  {  Console.WriteLine("Khong ton tai phan tu trong ma tran.");  } else  {  Console.WriteLine($"Vi tri cua phan tu do la: ({row}, {col})");  }  }  static bool IsPrime(int k)  {  for (int i=2;i\*i<=k;i++)  {  if (k % i == 0) return false;  }  return k > 1;  }  // Xuat cac phan tu la so nguyen to.  static void PrimeElement(int[,] matrix, int n, int m)  {  if (n == 0 && m == 0)  {  Console.WriteLine("Ma tran rong.");  return;  }  List<int> PrimeVector = new List<int>();  for (int i=0;i<n;i++)  {  for (int j=0;j<m;j++)  {  if (IsPrime(matrix[i,j]))  {  PrimeVector.Add(matrix[i,j]);  }  }  }  if ((int)PrimeVector.Count > 0)  {  Console.WriteLine("Cac so nguyen to trong ma tran. ");  foreach (int prime in PrimeVector)  {  Console.Write(prime + " ");  }  Console.WriteLine();  } else Console.WriteLine("Khong ton tai so nguyen to trong ma tran.");  }  // Chi so dong co nhieu so nguyen to nhat.  static void RowWithMostPrime(int[,] matrix, int n, int m)  {  if (n == 0 && m == 0)  {  Console.WriteLine("Ma tran rong.");  return;  }  int MaxPrimeNumb = 0;  int rowIndex = 0;  for (int i=0;i<n;i++)  {  int cnt = 0;  for (int j=0;j<m;j++)  {  if (IsPrime(matrix[i,j]))  {  cnt++;  }  }  if (MaxPrimeNumb < cnt)  {  rowIndex = i + 1;  MaxPrimeNumb = cnt;  }  }  if (rowIndex == 0) Console.WriteLine("Khong ton tai so nguyen to trong ma tran.");  else  {  Console.WriteLine($"Dong {rowIndex} co nhieu so nguyen to nhat.");  }  }  }  } |

Các Testcase của chương trình:

Bảng 3. : Các Testcase của chương trình bài 3

|  |  |
| --- | --- |
| Input  Output | 0  2 3  1 2 3  4 5 5  1  2  6  4  3  5 |
| Input  Output | 0  4 5  10 -20 89 76 5  1 2 3 4 5  0 0 0 0 0  100 -200 34 56 12345  3  2  10  5 |
| Input  Output | 9  0  abc 3  -10 3  100 90 80  5 6 7  -4 -3 -2  3  4  5 |
| Input  Output | 1  2  3  abc 10  0  4 2  1 2  0 10000  -90 80  34 5  1  5 |
| Input  Output | 4  0  5 5  2 5 4 1 7  5 1 7 8 3  11 8 9 2 6  3 12 1 4 9  10 6 13 15 2  3  4  -3223  5 |
| Input  Output | 0  8 7  15 2 7 9 1 10 5  8 3 11 6 4 22 19  1 5 13 18 3 7 14  9 20 6 4 8 1 12  30 5 17 21 2 9 8  11 4 6 16 1 25 3  7 19 10 3 14 5 2  2 8 12 1 23 18 6  2  11  2  23  3  4  1  5 |

# 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.

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

* **Tìm phân số lớn nhất.**
* **Sắp xếp các phân số trong dãy tăng dần.**

Mô tả đầu vào và đầu ra của bài toán:

* Input: Một số nguyên dương là một sự lựa chọn tương ứng với các lựa chọn sau:

+ 0: Nhập vào 2 phân số. Mỗi phân số sẽ nhập 2 số nguyên lần lượt là tử và mẫu của phân số đó.

+ 1: Nhập một dãy các phân số. Mỗi phân số sẽ nhập 2 số nguyên lần lượt là tử và mẫu của phân số đó.

+ 2: Xuất ra 2 phân số đã nhập.

+ 3: Xuất ra dãy phân số đã nhập.

+ 4: Tính tổng 2 phân số đã nhập.

+ 5: Tính hiệu 2 phân số đã nhập.

+ 6: Tính tích 2 phân số đã nhập.

+ 7: Tính thương 2 phân số đã nhập.

+ 8: Tìm phân số lớn nhất trong dãy phân số đã nhập.

+ 9: Sắp xếp các phân số trong dãy tăng dần.

+ 10: Thoát khỏi các sự lựa chọn.

* Output: Kết quả xuất ra màn hình sẽ dựa theo sự lựa chọn ở phần nhập.

Nội dung code của chương trình bài 4:

Bảng 4. . 1: Nội dung code của file PhanSo.cs

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Bai04  {  class PhanSo  {  private int tu, mau;  public PhanSo()  {  this.tu = 0;  this.mau = 1;  }  public int Tu  {  get { return tu; }  set { tu = value; }  }  public int Mau  {  get { return mau; }  set { mau = value; }  }  public int gcd(int x, int y)  {  while (y > 0)  {  int tmp = y;  y = x % y;  x = tmp;  }  return x;  }  public static PhanSo operator +(PhanSo a, PhanSo b)  {  PhanSo sum = new PhanSo();  int x = a.Tu \* b.Mau + a.Mau \* b.Tu;  int y = a.Mau \* b.Mau;  int common = a.gcd(Math.Abs(x), Math.Abs(y));  x /= common;  y /= common;  sum.Tu = x;  sum.Mau = y;  return sum;  }  public static PhanSo operator -(PhanSo a, PhanSo b)  {  PhanSo sub = new PhanSo();  int x = a.Tu \* b.Mau - a.Mau \* b.Tu;  int y = a.Mau \* b.Mau;  int common = a.gcd(Math.Abs(x), Math.Abs(y));  x /= common;  y /= common;  sub.Tu = x;  sub.Mau = y;  return sub;  }  public static PhanSo operator \*(PhanSo a, PhanSo b)  {  PhanSo mul = new PhanSo();  int x = a.Tu \* b.Tu;  int y = a.Mau \* b.Mau;  int common = a.gcd(Math.Abs(x), Math.Abs(y));  x /= common;  y /= common;  mul.Tu = x;  mul.Mau = y;  return mul;  }  public static PhanSo operator /(PhanSo a, PhanSo b)  {  PhanSo divide = new PhanSo();  int x = a.Tu \* b.Mau;  int y = a.Mau \* b.Tu;  int common = a.gcd(Math.Abs(x), Math.Abs(y));  x /= common;  y /= common;  divide.Tu = x;  divide.Mau = y;  return divide;  }  public static bool operator >(PhanSo a, PhanSo b)  {  if (a.Tu \* a.Mau > 0)  {  a.Tu = Math.Abs(a.Tu);  a.Mau = Math.Abs(a.Mau);  }  else  {  a.Tu = Math.Abs(a.Tu);  a.Mau = Math.Abs(a.Mau);  a.Tu = -a.Tu;  }  if (b.Tu \* b.Mau > 0)  {  b.Tu = Math.Abs(b.Tu);  b.Mau = Math.Abs(b.Mau);  }  else  {  b.Tu = Math.Abs(b.Tu);  b.Mau = Math.Abs(b.Mau);  b.Tu = -b.Tu;  }  return (a.Tu \* b.Mau > b.Tu \* a.Mau);  }  public static bool operator <(PhanSo a, PhanSo b)  {  if (a.Tu \* a.Mau > 0)  {  a.Tu = Math.Abs(a.Tu);  a.Mau = Math.Abs(a.Mau);  }  else  {  a.Tu = Math.Abs(a.Tu);  a.Mau = Math.Abs(a.Mau);  a.Tu = -a.Tu;  }  if (b.Tu \* b.Mau > 0)  {  b.Tu = Math.Abs(b.Tu);  b.Mau = Math.Abs(b.Mau);  }  else  {  b.Tu = Math.Abs(b.Tu);  b.Mau = Math.Abs(b.Mau);  b.Tu = -b.Tu;  }  return (a.Tu \* b.Mau < b.Tu \* a.Mau);  }  }  } |

Bảng 4. 1. : Nội dung code của file Program.cs

|  |
| --- |
| using System;  using System.CodeDom;  using System.Collections.Generic;  using System.Data.SqlTypes;  using System.Linq;  using System.Runtime.CompilerServices;  using System.Text;  using System.Threading.Tasks;  namespace Bai04  {  // Lop PhanSo  class Program  {  static void Main(string[] args)  {  PhanSo a = new PhanSo();  PhanSo b = new PhanSo();  int n = 0;  PhanSo[] array = new PhanSo[n];  Console.WriteLine("============Menu============");  Console.WriteLine("0: Nhap 2 phan so.");  Console.WriteLine("1: Nhap mot day cac phan so.");  Console.WriteLine("2: Xuat ra 2 phan so.");  Console.WriteLine("3: Xuat ra day phan so.");  Console.WriteLine("4: Tinh tong 2 phan so.");  Console.WriteLine("5: Tinh hieu 2 phan so.");  Console.WriteLine("6: Tinh tich 2 phan so.");  Console.WriteLine("7: Tinh thuong 2 phan so.");  Console.WriteLine("8: Tim phan so lon nhat trong day phan so.");  Console.WriteLine("9: Sap xep cac phan so trong day tang dan.");  Console.WriteLine("10: Thoat.");  Console.WriteLine("============================");  int choice;  while (true)  {  Console.Write("Nhap lua chon: ");  while (true)  {  try  {  choice = Convert.ToInt32(Console.ReadLine());  break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  if (choice == 0)  {  InputPhanSo(ref a);  InputPhanSo(ref b);  }  else if (choice == 1)  {  InputPhanSoArray(ref array, ref n);  }  else if (choice == 2)  {  Console.Write("Phan so thu nhat: ");  OutputPhanSo(a);  Console.WriteLine();  Console.Write("Phan so thu hai: ");  OutputPhanSo(b);  Console.WriteLine();  }  else if (choice == 3)  {  OutputArray(array, n);  }  else if (choice == 4)  {  Console.Write("Tong 2 phan so la: ");  OutputPhanSo(a + b);  Console.WriteLine();  }  else if (choice == 5)  {  Console.Write("Hieu 2 phan so la: ");  OutputPhanSo(a - b);  Console.WriteLine();  }  else if (choice == 6)  {  Console.Write("Tich 2 phan so la: ");  OutputPhanSo(a \* b);  Console.WriteLine();  }  else if (choice == 7)  {  Console.Write("Thuong 2 phan so la: ");  OutputPhanSo(a / b);  Console.WriteLine();  }  else if (choice == 8)  {  FindMaxPhanSo(array, n);  }  else if (choice == 9)  {  SortArray(ref array, n);  } else if (choice == 10)  {  break;  } else Console.WriteLine("Lua chon khong hop le.");  }  }  // Xuat ra 1 day phan so.  static void OutputArray(PhanSo[] array, int n)  {  if (n==0)  {  Console.WriteLine("Mang rong.");  return;  }  Console.WriteLine("Cac phan so trong day: ");  for (int i=0;i<n;i++)  {  Console.Write($"Phan so thu {i+1}: ");  OutputPhanSo(array[i]);  Console.WriteLine();  }  }  // Xuat 1 phan so  static void OutputPhanSo(PhanSo a)  {  int common = a.gcd(Math.Abs(a.Tu), Math.Abs(a.Mau));  a.Tu /= common;  a.Mau /= common;  if (a.Tu \* a.Mau > 0)  {  a.Tu = Math.Abs(a.Tu);  a.Mau = Math.Abs(a.Mau);  } else  {  a.Tu = Math.Abs(a.Tu);  a.Mau = Math.Abs(a.Mau);  a.Tu = -a.Tu;  }  Console.Write(a.Tu);  if (a.Mau != 1) Console.Write("/" + a.Mau);  }  // Nhap 2 phan so  static void InputPhanSo(ref PhanSo a)  {  Console.Write("Nhap tu cho phan so: ");  while(true)  {  try  {  a.Tu = Convert.ToInt32(Console.ReadLine());  break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  Console.Write("Nhap mau so cho phan so: ");  while (true)  {  try  {  a.Mau = Convert.ToInt32(Console.ReadLine());  break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  }  // Nhap mot day phan so  static void InputPhanSoArray(ref PhanSo[] array, ref int n)  {  Console.Write("Nhap so luong phan tu trong day: ");  while(true)  {  try  {  n = Convert.ToInt32(Console.ReadLine());  if (n <= 0) Console.Write("Phai nhap so duong. Moi nhap lai: ");  else break;  }  catch  {  Console.Write("Loi.Moi nhap lai: ");  }  }  array = new PhanSo[n];  for (int i=0;i<n;i++)  {  array[i] = new PhanSo();  InputPhanSo(ref array[i]);  }  }  // Tim phan so lon nhat trong day  static void FindMaxPhanSo(PhanSo[] array, int n)  {  if (n==0)  {  Console.WriteLine("Mang rong.");  return;  }  PhanSo MaxPhanSo = new PhanSo();  MaxPhanSo.Tu = -100000;  for (int i=0;i<n;i++)  {  if (MaxPhanSo < array[i])  {  MaxPhanSo = array[i];  }  }  Console.Write("Phan so lon nhat trong day la: ");  OutputPhanSo(MaxPhanSo);  Console.WriteLine();  }  // Sap xep cac phan so trong day tang dan  static void SortArray(ref PhanSo[] array, int n)  {  if (n==0)  {  Console.WriteLine("Mang rong.");  return;  }  for (int i=0;i<n;i++)  {  for (int j=i+1;j<n;j++)  {  if (array[i] > array[j])  {  PhanSo tmp = array[i];  array[i] = array[j];  array[j] = tmp;  }  }  }  }  }  } |

Các Testcase của chương trình bài 4:

Bảng 4. 2: Các Testcase của chương trình bài 4

|  |  |
| --- | --- |
| Input  Output | 0  2 3  10 7  2  1  4  2 5  6 7  10 9  100 20  3  10 |
| Input  Output | 1  5  1009 9  20 6  7 8  9 10  3 4  8  9  3  abcd  -100  10 |
| Input  Output | 0  20 4  2 3  2  4  5  6  7  8  9  3  10 |
| Input  Output | -abda -100 12 0  abcd 2.6 -10 4  2  10  2  1  -100  5  2 3  4 5  6 7  7 8  9 10  9  3 10 |
| Input  Output | 1  6  100 6  20 7  90 4  3 2  -9 4  -23 6  3  9  3  2.897  10 |

# 5. Xây dựng phần mềm quản lý cho Công ty địa ốc Đại Phú với các yêu cầu sau:

**- Xây dựng các lớp theo yêu cầu của đề bài với chức năng thừa kế.**

**- Nhập suất 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: Một số nguyên dương biểu thị cho sự lựa chọn với:

+ 1: Nhập thông tin cho các khu vực muốn quản lý. Dòng đầu tiên sẽ là một số nguyên dương – số lượng khu vực quản lý. Với mỗi khu vực quản lý sẽ yêu cầu phần nhập thông tin khác nhau.

+ 2: Xuất thông tin các khu vực quản lý.

+ 3: Xuất tổng giá bán cho từng loại khu vực (Khu đất, Nhà phố, Chung cư).

+ 4: Xuất danh sách các khu vực thỏa mãn điều kiện đề ra ở đề bài.

+ 5: Tìm kiếm các khu vực thỏa mãn 3 tiêu chỉ địa điểm, giá cả và diện tích như đề bài yêu cầu. Nhập vào một chuỗi, một số nguyên dương và một số thực – tương ứng lần lượt là địa điểm, giá cả và diện tích cần tìm.

* Output: Dựa theo sự lựa chọn từ phần nhập sẽ cho ra kết quả tương ứng.

Nội dung code của chương trình bài 5:

Bảng 5. . 1: Nội dung code của file program.cs

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Diagnostics.Eventing.Reader;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Bai05  {  class Program  {  static void Main(string[] args)  {  int n = 0;  ThiTruong[] PhanMem = new ThiTruong[n];  Console.WriteLine(" PHAN MEM QUAN LY");  Console.WriteLine("============Menu============");  Console.WriteLine("1: Nhap thong tin cac khu vuc.");  Console.WriteLine("2: Xuat thong tin cac khu vuc.");  Console.WriteLine("3: Xuat tong gia ban cho tung loai khu vuc.");  Console.WriteLine("4: Xuat danh sach cac khu vuc thoa man dieu kien de ra.");  Console.WriteLine("5: Tim kiem khu vuc thoa.");  Console.WriteLine("6: Thoat.");  Console.WriteLine("============================");  int choice;  while(true)  {  Console.Write("Moi nhap lua chon: ");  while(true)  {  try  {  choice = Convert.ToInt32(Console.ReadLine());  break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  if (choice == 1)  {  NhapThongTin(ref PhanMem, ref n);  }  else if (choice == 2)  {  XuatThongTin(PhanMem, n);  }  else if (choice == 3)  {  TongTien(PhanMem, n);  }  else if (choice == 4)  {  Validate(PhanMem, n);  }  else if (choice == 5)  {  FindLocation(PhanMem, n);  }  else if (choice == 6)  {  break;  }  else Console.WriteLine("Lua chon khong hop le. Moi chon lai: ");  }  }  // Nhap thong tin cho tung khu vuc  static void NhapThongTin(ref ThiTruong[] PhanMem, ref int n)  {  Console.Write("Nhap so luong khu vuc quan ly: ");  while(true)  {  try  {  n = Convert.ToInt32(Console.ReadLine());  if (n <= 0)  {  Console.Write("Moi nhap lai: ");  } else break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  PhanMem = new ThiTruong[n];  for (int i=0;i<n;i++)  {  Console.WriteLine($"Nhap thong tin cho khu vuc thu {i + 1}: ");  int type;  Console.Write("Nhap loai khu vuc (1: Khu Dat 2: Nha Pho 3: Chung Cu): ");  while (true)  {  while (true)  {  try  {  type = Convert.ToInt32(Console.ReadLine());  break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  if (type == 1)  {  PhanMem[i] = new KhuDat();  break;  }  if (type == 2)  {  PhanMem[i] = new NhaPho();  break;  }  if (type == 3)  {  PhanMem[i] = new ChungCu();  break;  }  Console.Write("Lua chon khong hop le. Moi nhap lai: ");  }  PhanMem[i].Input();  }  }  // Xuat thong tin cho tung khu vuc  static void XuatThongTin(ThiTruong[] PhanMem, int n)  {  if (n==0)  {  Console.WriteLine("Danh sach rong.");  return;  }  Console.WriteLine("Thong tin cac khu vuc phan mem quan ly: ");  for (int i=0;i<n;i++)  {  Console.WriteLine($"Thong tin khu vuc thu {i+1}: ");  PhanMem[i].Output();  }  }  static void TongTien(ThiTruong[] PhanMem, int n)  {  int TienKhuDat = 0, TienNhaPho = 0, TienChungCu = 0;  for (int i=0;i<n;i++)  {  TienKhuDat += PhanMem[i].GiaKhuDat();  TienNhaPho += PhanMem[i].GiaNhaPho();  TienChungCu += PhanMem[i].GiaChungCu();  }  Console.WriteLine("Tong tien cua cac khu vuc la khu dat: " + TienKhuDat);  Console.WriteLine("Tong tien cua cac khu vuc la nha pho: " + TienNhaPho);  Console.WriteLine("Tong tien cua cac khu vuc la chung cu: " + TienChungCu);  }  // Xuat danh sach cac khu vuc thoa ma yeu cau de bai  static void Validate(ThiTruong[] PhanMem, int n)  {  if (n==0)  {  Console.WriteLine("Danh sach rong.");  return;  }  List<ThiTruong> MyVector = new List<ThiTruong>();  for (int i=0;i<n;i++)  {  if (PhanMem[i].Valid())  {  MyVector.Add(PhanMem[i]);  }  }  if ((int)MyVector.Count > 0)  {  Console.WriteLine("Danh sach cac khu vuc thoa: ");  foreach (ThiTruong t in MyVector)  {  t.Output();  }  return;  }  Console.WriteLine("Khong co khu vuc nao thoa dieu kien.");  }  // Tim kiem danh sach cac khu vuc theo dia diem, gia va dien tich  static void FindLocation(ThiTruong[] PhanMem, int n)  {  if (n == 0)  {  Console.WriteLine("Danh sach rong.");  return;  }  string location;  int price;  double area;  Console.Write("Nhap dia diem muon tim: ");  location = Console.ReadLine();  Console.Write("Nhap gia muon tim: ");  while (true)  {  try  {  price = Convert.ToInt32(Console.ReadLine());  if (price < 0) Console.Write("So am. Moi nhap lai: ");  else break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  Console.Write("Nhap dien tich muon tim: ");  while (true)  {  try  {  area = Convert.ToDouble(Console.ReadLine());  if (area < 0f) Console.Write("So am. Moi nhap lai: ");  else break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  List<ThiTruong> PossibleLocation = new List<ThiTruong>();  for (int i=0;i<n;i++)  {  if (PhanMem[i].Possible(location, price, area))  {  PossibleLocation.Add(PhanMem[i]);  }  }  if ((int)PossibleLocation.Count>0)  {  Console.WriteLine("Danh sach cac nha pho va chung cu thoa yeu cau: ");  foreach (ThiTruong t in PossibleLocation)  {  t.Output();  }  return;  }  Console.WriteLine("Khong co khu vuc nao thoa yeu cau can tim.");  }  }  } |

Bảng 5. 1. : Nội dung code của file ThiTruong.cs

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Bai05  {  abstract class ThiTruong  {  protected string DiaDiem;  protected int GiaBan;  protected double DienTich;  protected void InfoInput()  {  Console.Write("Nhap dia diem: ");  DiaDiem = Console.ReadLine();  Console.Write("Nhap gia ban: ");  while(true)  {  try  {  GiaBan = Convert.ToInt32(Console.ReadLine());  if (GiaBan < 0) Console.Write("So am. Moi nhap lai: ");  else break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  Console.Write("Nhap dien tich: ");  while(true)  {  try  {  DienTich = Convert.ToDouble(Console.ReadLine());  if (DienTich < 0f) Console.Write("So am. Moi nhap lai: ");  else break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  }  protected void InfoOutput()  {  Console.WriteLine("Dia diem cua khu vuc nay: " + DiaDiem);  Console.WriteLine("Gia ban cua khu vuc nay: " + GiaBan);  Console.WriteLine("Dien tich cua khu vuc nay: " + DienTich);  }  public abstract void Input();  public abstract void Output();  public abstract int GiaKhuDat();  public abstract int GiaNhaPho();  public abstract int GiaChungCu();  public abstract bool Valid();  public abstract bool Possible(string location, int price, double area);  }  } |

Bảng 5. 1. : Nội dung code của file KhuDat.cs

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Bai05  {  class KhuDat:ThiTruong  {  public override void Input()  {  Console.WriteLine("Nhap thong tin khu dat: ");  InfoInput();  }  public override void Output()  {  Console.WriteLine("Thong tin khu dat: ");  InfoOutput();  }  public override int GiaChungCu()  {  return 0;  }  public override int GiaKhuDat()  {  return GiaBan;  }  public override int GiaNhaPho()  {  return 0;  }  public override bool Valid()  {  if (DienTich > 100) return true;  return false;  }  public override bool Possible(string location, int price, double area)  {  return false;  }  }  } |

Bảng 5. 1. : Nội dung code của file NhaPho.cs

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Bai05  {  class NhaPho: ThiTruong  {  private int NamXayDung;  private int SoTang;  public override void Input()  {  Console.WriteLine("Nhap thong tin khu pho: ");  InfoInput();  Console.Write("Nhap nam xay dung: ");  while(true)  {  try  {  NamXayDung = Convert.ToInt32(Console.ReadLine());  if (NamXayDung <= 0)  {  Console.Write("So am. Moi nhap lai: ");  } else break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  Console.Write("Nhap so tang: ");  while (true)  {  try  {  SoTang = Convert.ToInt32(Console.ReadLine());  if (SoTang <= 0)  {  Console.Write("So am. Moi nhap lai: ");  }  else break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  }  public override void Output()  {  Console.WriteLine("Thong tin khu pho: ");  InfoOutput();  Console.WriteLine("Nam xay dung: " + NamXayDung);  Console.WriteLine("So tang: " + SoTang);  }  public override int GiaChungCu()  {  return 0;  }  public override int GiaKhuDat()  {  return 0;  }  public override int GiaNhaPho()  {  return GiaBan;  }  public override bool Valid()  {  if (DienTich > 60 && NamXayDung >= 2019) return true;  return false;  }  public override bool Possible(string location, int price, double area)  {  string current = DiaDiem.ToLower();  location = location.ToLower();  bool have = current.Contains(location);  if (have && GiaBan <= price && DienTich >= area) return true;  return false;  }  }  } |

Bảng 5. 1. : Nội dung code của file ChungCu.cs

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Bai05  {  class ChungCu: ThiTruong  {  private int SoTang;  public override void Input()  {  Console.WriteLine("Nhap thong tin cho chung cu: ");  InfoInput();  Console.Write("Nhap so tang: ");  while(true)  {  try  {  SoTang = Convert.ToInt32(Console.ReadLine());  if (SoTang <= 0)  {  Console.Write("So am. Moi nhap lai: ");  } else break;  }  catch  {  Console.Write("Loi. Moi nhap lai: ");  }  }  }  public override void Output()  {  Console.WriteLine("Thong tin khu chung cu: ");  InfoOutput();  Console.WriteLine("So tang: " + SoTang);  }  public override int GiaChungCu()  {  return GiaBan;  }  public override int GiaKhuDat()  {  return 0;  }  public override int GiaNhaPho()  {  return 0;  }  public override bool Valid()  {  return false;  }  public override bool Possible(string location, int price, double area)  {  string current = DiaDiem.ToLower();  location = location.ToLower();  bool have = current.Contains(location);  if (have && GiaBan <= price && DienTich >= area) return true;  return false;  }  }  } |

Các Testcase của chương trình bài 5:

Bảng 5. 2: Các Testcase của chương trình bài 5

|  |  |
| --- | --- |
| Input  Output | 5  4  1  4  1  Binh Duong  100000  60,7  2  Quan 1  500000  300  2020  30  3  Binh Thanh  40000  50  10  2  Lao Cai  300500  200  2006  30  2  6 |
| Input  Output | 1  5  1  300000  80  2  Binh Phuoc  200000  60,78  2000  20  3  Binh Quoi  450000  200  20  2  Phuoc Long  120000  45,6  1975  20  3  Binh Dinh  600000  400  40  3  4  5  Binh Phuoc  300000  30  6 |
| Input  Output | 100  abcd  -19  1  3  Vinh Long  300000  120  2  An Giang  40000  75,8  2020  20  3  Gia Lai  300000  20,7  20  4  6 |
| Input  Output | 3  Pham Dan Truong  1  6  1  Binh Duong  200000  120  2  Binh Duong  250000  76,8  2020  10  3  Binh Duong  400000  80,9  20  2  Binh Thuan  abccde  -100  250000  90,89  2024  30  5  Binh Duong  500000  50  6 |
| Input  Output | 2,78  5  1  -20  abcdfe  3  k  2  Binh Thuan  An Khang  24000  24,67  2020  20  1  Binh Dinh  23000  24,78  1  Binh Thuan  120000  12,6  2  3  4  5  Binh Thuan  23000  12  6 |

Link GitHub: <https://github.com/thichcodedao2006/IT008>