**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 X**

GVHD: Nguyễn Ngọc Quí

Sinh viên thực hiện: Nguyễn Chí Nguyên

MSSV: 24521186

🙡🙢 Tp. Hồ Chí Minh, 10/2024 🙠🙣

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*……., ngày……...tháng……năm 2024*

**Người nhận xét**

*(Ký tên và ghi rõ họ tên****)***

**MỤC LỤC**

[NỘI DUNG BÀI LÀM 6](#_Toc211630079)

[1. Viết chương trình nhập vào tháng và năm, in ra lịch của tháng đó. 6](#_Toc211630080)

[2. Viết chương trình cho phép tạo mảng một chiều gồm n số nguyên ngẫu nhiên. 8](#_Toc211630081)

[3. Viết chương trình cho phép tạo mảng một chiều gồm n số nguyên ngẫu nhiên. 10](#_Toc211630082)

[4. Viết chương trình cho phép tạo mảng một chiều gồm n số nguyên ngẫu nhiên. 15](#_Toc211630083)

[5. Viết chương trình cho phép tạo mảng một chiều gồm n số nguyên ngẫu nhiên. 24](#_Toc211630084)

**DANH MỤC BẢNG**

[Bảng 0‑1 6](#_Toc211711655)

[Bảng 0‑2 7](#_Toc211711656)

[Bảng 0‑1 8](#_Toc211711657)

[Bảng 0‑2 9](#_Toc211711658)

[Bảng 0‑1 10](#_Toc211711659)

[Bảng 0‑2 13](#_Toc211711660)

[Bảng 0‑1 16](#_Toc211711661)

[Bảng 0‑2 17](#_Toc211711662)

[Bảng 0‑3 18](#_Toc211711663)

[Bảng 0‑4 19](#_Toc211711664)

[Bảng 0‑5 23](#_Toc211711665)

**DANH MỤC HÌNH ẢNH**

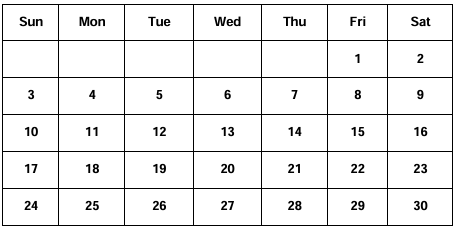
**No table of figures entries found.**

# NỘI DUNG BÀI LÀM

## 1. Viết chương trình nhập vào tháng và năm, in ra lịch của tháng đó.

Mô tả đầu vào và đầu ra của bài toán:

* Input: Hai số nguyên lần lượt là tháng và năm. Nếu tháng và năm không hợp lệ thì yêu cầu nhập lại đến khi hợp lệ.
* Output: Lịch của tháng/năm vừa nhập theo mẫu như hình



Bảng ‑

|  |
| --- |
| using System;  namespace LAB2.Bai01  {  class Program  {  static void Main(string[] args)  {  // Nhap du lieu  int month, year;  while(true)  {  Console.Write("Nhap thang (1->12): ");  string m = Console.ReadLine();  Console.Write("Nhap nam (> 0): ");  string y = Console.ReadLine();  if((!int.TryParse(m, out month) || !int.TryParse(y, out year))  || (month < 1 || month > 12 || year < 1))  {  Console.WriteLine("Invalid data. Please try again!");  }  else  {  break;  }  }  // Khai bao cac bien su dung  DateTime d = new DateTime(year, month, 1);  int startDay = (int)d.DayOfWeek;  int dayInMonth = DateTime.DaysInMonth(year, month);  // Xu ly va in lich ra man hinh  Console.WriteLine("Sun\tMon\tTue\tWed\tThu\tFri\tSat");  for (int i = 0; i < startDay; i++)  Console.Write("\t");  for(int day = 1; day <= dayInMonth; day++)  {  Console.Write(day + "\t");  if((day + startDay)% 7 == 0)  {  Console.Write("\n");  }  }    }  }  } |

Bảng ‑

|  |  |
| --- | --- |
| Input:  10 2025 |  |
| Input:  0 2025  5 -1  5 2025 |  |
| Input:  abc 2025  11 xyz  11 2025 |  |
| Input:  3.14 2025  3 20.25  3 2025 |  |
| Input:  12 2100 |  |
| Input:  2 2000 |  |

## 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 input và output của bài toán.

* Input: Đường dẫn thư mục bất kì. Nếu đường dẫn không hợp lệ thì nhập lại đến khi hợp lệ.
* Output: Tên tất cả các thư mục con và tập tin trong đường dẫn thư mục đã nhập.

Bảng ‑

|  |
| --- |
| using System;  namespace LAB2.Bai02  {  class Program  {  static void Main(string[] args)  {  string dir;  // Nhap duong dan thu muc, neu duong dan khong ton tai thi yeu cau nhap lai  while (true)  {  Console.Write("Nhap duong dan thu muc: ");  dir = Console.ReadLine();  if(!Directory.Exists(dir))  {  Console.WriteLine("Khong tim thay thu muc! Vui long thu lai duong dan khac!");  }  else  {  break;  }  }  /// Thuc hien xu ly va in ra tat ca thu muc va tap tin  string[] childFolders = Directory.GetDirectories(dir);  string[] childFiles = Directory.GetFiles(dir);  // In ra cac thu muc va tap tin  if(childFolders.Length != 0 || childFiles.Length != 0)  {  // In cac thu muc  for(int i = 0; i < childFolders.Length; i++)  {  Console.WriteLine("[DIR]\t" + Path.GetFileName(childFolders[i]));  }  // In cac tep tin  for(int i = 0; i < childFiles.Length; i++)  {  Console.WriteLine("[FILE]\t" + Path.GetFileName(childFiles[i]));  }  }  else  {  Console.WriteLine("Khong co thu muc hay tap tin nao trong duong dan thu muc da nhap!");  }  }  }  } |

Bảng ‑

|  |  |
| --- | --- |
| Input:  C:\Users\luugi\Desktop |  |
| Input:  123  C:\Users\luugi\Desktop  \LTTQ\BaiTapLythuyet |  |
| Input:  C:\Users\luugi\Desktop  \LTTQ\ThuMucRong |  |
| Input:  abc  xyz  C:\Users\luugi\Desktop  \UNITY |  |

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

1. ***Nhập / xuất ma trận hai chiều các số nguyên***
2. ***Tìm kiếm một phần tử trong ma trận***
3. ***Xuất các phần tử là số nguyên tố***
4. ***Cho biết dòng nào có nhiều số nguyên tố nhất***

Mô tả input/output của bài toán

* Input:

+ Hai số nguyên n và m lớn hơn 0, lần lượt là số dòng và số cột của ma trận.

+ Trên n dòng tiếp theo, mỗi dòng có m số nguyên là các phần tử của ma trận.

+ Số nguyên k là phần tử cần tìm kiếm trong ma trận.

* Output:

+ Ma trận n dòng và m cột đã nhập ra màn hình.

+ Kết quả tìm kiếm phần từ k trong ma trận.

+ Tất cả các phần tử là số nguyên tố trong ma trận.

+ Dòng có số lượng số nguyên tố nhiều nhất trong ma trận. In ra -1 nếu ma trận không có số nguyên tố. Nếu có nhiều kết quả thì in ra dòng đầu tiên.

Bảng ‑

|  |
| --- |
| using System;  namespace LAB2.Bai03  {  class Program  {  static void Main(string[] args)  {  int numRows, numCols; // n: number of rows, m: number of column  // Nhap so dong va so cot  while(true)  {  Console.Write("Nhap so dong cua ma tran (so dong > 0): ");  string r = Console.ReadLine();  Console.Write("Nhap so cot cua ma tran (so cot > 0): ");  string c = Console.ReadLine();  if((!int.TryParse(r, out numRows) || !int.TryParse(c, out numCols))  || (numRows <= 0 || numCols <= 0))  Console.WriteLine("So dong hoac so sot khong hop le. Vui long nhap lai! ");  else  break;  }  Matrix matrix = new Matrix(numRows, numCols);  // a. Nhap/Xuat ma tran  Console.WriteLine("Nhap ma tran: ");  matrix.Nhap();  Console.WriteLine("Ma tran vua nhap: ");  matrix.Xuat();  // b. Tim kiem phan tu trong ma tran  Console.Write("Nhap phan tu can tim (so nguyen): ");  int ele;  while(true)  {  string input = Console.ReadLine();  if(!int.TryParse(input, out ele))  Console.WriteLine("Du lieu vua nhap khong hop le. Vui long nhap lai!");  else  break;  }  if (matrix.FindElement(ele))  Console.WriteLine("Phan tu {0} co ton tai trong ma tran.", ele);  else  Console.WriteLine("Phan tu {0} khong ton tai trong ma tran.", ele);  // c. Xuat cac phan tu la so nguyen to trong ma tran  Console.WriteLine("Cac phan tu la so nguyen to trong ma tran: ");  if(!matrix.PrintPrime())  {  Console.WriteLine("Khong co phan tu nao la so nguyen to trong ma tran.");  }  // d. Tim dong co so luong so nguyen to nhieu nhat  int maxPrimeCount = matrix.MaxPrimeCountRow();  Console.WriteLine("\nSo luong so nguyen to nhieu nhat trong ma tran la: {0}", maxPrimeCount);  }  private static bool IsPrime(int n)  {  if (n < 2) return false;  for (int i = 2; i \* i <= n; i++)  if (n % i == 0)  return false;  return true;  }  // lop Matrix de thuc hien cac yeu cau cua de bai  public class Matrix  {  int numRows, numCols;  int[][] matrix;  public Matrix(int numRows = 0, int numCols = 0)  {  this.numRows = numRows;  this.numCols = numCols;  matrix = new int[this.numRows][];  for (int i = 0; i < this.numRows; i++)  {  matrix[i] = new int[numCols];  }  }  // Ham nhap ma tran  public void Nhap()  {  for (int i = 0; i < numRows; i++)  {  string[] parts = Console.ReadLine().Trim().Split(' ');  for (int j = 0; j < numCols; j++)  {  if(j < parts.Length && int.TryParse(parts[j], out matrix[i][j]))  {  continue;  }  else  {  Console.WriteLine("Invalid data. Please enter the row again:");  j--;  }  }  }  }  // Ham xuat ma tran  public void Xuat()  {  for (int i = 0; i < numRows; i++)  {  for (int j = 0; j < numCols; j++)  {  Console.Write(matrix[i][j] + " ");  }  Console.Write("\n");  }  }  // Tim kiem  public bool FindElement(int ele)  {  for (int i = 0; i < numRows; i++)  {  for (int j = 0; j < numCols; j++)  {  if(matrix[i][j] == ele)  return true;  }  }  return false;  }  // Xuat cac phan tu la so nguyen to  public bool PrintPrime()  {  bool found = false;  for (int i = 0; i < numRows; i++)  {  for (int j = 0; j < numCols; j++)  {  if (IsPrime(matrix[i][j]))  {  Console.Write(matrix[i][j] + " ");  found = true;  }  }  Console.WriteLine();  }  return found;  }  // Tim dong co so luong so nguyen to nhieu nhat  public int MaxPrimeCountRow()  {  int res = -1; // Tra ve -1 neu ma tran khong co so nguyen to  int primeCount = 0;  for(int i = 0; i < numRows; i++)  {  int temp = 0;  for(int j = 0; j < numCols; j++)  {  if (IsPrime(matrix[i][j]))  temp++;  }  if(temp > primeCount)  {  primeCount = temp;  res = i;  }  }  return res;  }  }  }  } |

Bảng ‑

|  |  |
| --- | --- |
| Input:  3 3  2 4 6  1 3 5  5 2 7  6 |  |
| Input:  3 2  1 4  6 8  12 9  999 |  |
| Input:  0 3  3 0  3 3  1 2 3  3 2 1  2 3 1  1 |  |
| Input:  A 3  2 B  2 3  2 4 6  8 9 10  15 |  |
| Input:  2 3  8 9 12  5 11 25  a  @  25 |  |
| Input:  -99 3  3 3.14  3 3  1 1 1  2 2 2  4 3 3  10 |  |

## 4. Xây dựng lớp phân số với toán tử +, -, \*, /.

1. **Viết chương trình nhập vào hai phân số, xuất ra các giá trị là tổng, hiệu, tích và thương của hai phân số này.**
2. **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***

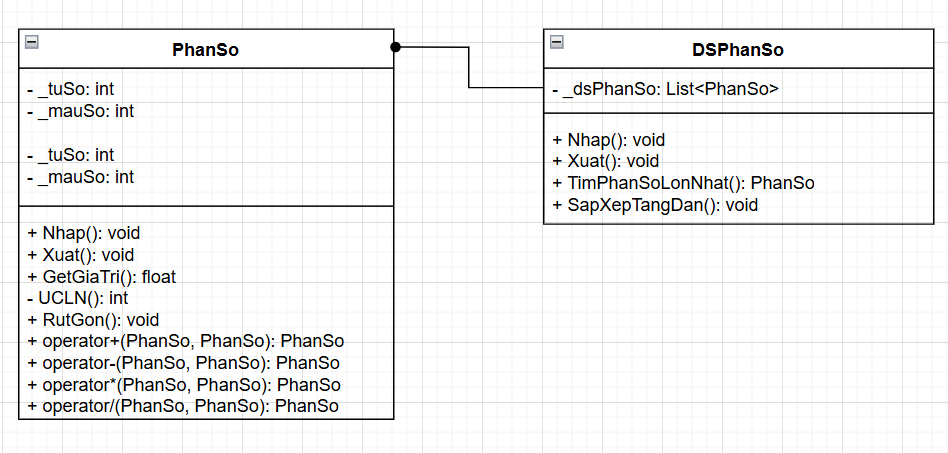
Mô tả input/output của bài toán chương trình a.

* Input: Bốn số nguyên lần lượt là tử và mẫu của phân số thứ nhất, tử và mẫu của phân số thứ hai.
* Output: Kết quả của bốn phép tính tổng, hiệu, tích, thương của hai phân số.

Mô tả input/output của bài toán chương trình b.

* Input: Số nguyên n là số lượng các phân số cần nhập. 2n số nguyên lần lượt là tử và mẫu của n phân số.
* Output: In phân số lớn nhất ra màn hình. In danh sách phân số sau khi đã sắp xếp tăng dần.

Hình



Bảng ‑

|  |
| --- |
| using System;  namespace LAB2.Bai04  {  public class PhanSo  {  private int \_tuSo;  private int \_mauSo;  public PhanSo(int tuSo = 0, int mauSo = 1)  {  \_tuSo = tuSo;  \_mauSo = mauSo;  }  public int gettuSo() => \_tuSo;  public int getmauSo() => \_mauSo;  public void settuSo(int tuSo) => \_tuSo = tuSo;  public void setmauSo(int mauSo) => \_mauSo = mauSo == 0 ? 1 : mauSo;  public void Nhap()  {  while (true)  {  Console.Write("Nhap tu so: ");  string t = Console.ReadLine();  Console.Write("Nhap mau so: ");  string m = Console.ReadLine();  if (!int.TryParse(t, out \_tuSo) || !int.TryParse(m, out \_mauSo))  Console.WriteLine("Tu hoac mau khong hop le. Vui long nhap lai!");  else  break;  }  RutGon();  }  public void Xuat()  {  if(\_tuSo == 0)  {  Console.WriteLine("0");  return;  }  else if(\_tuSo % \_mauSo == 0)  {  Console.WriteLine("{0}", \_tuSo / \_mauSo);  return;  }  Console.Write("{0}/{1}", \_tuSo, \_mauSo);  }  public float GetGiaTri() => (float)\_tuSo / \_mauSo;  private int UCLN(int a, int b)  {  if (a == 0)  return b;  return UCLN(b % a, a);  }  public void RutGon()  {  int ucln = UCLN(Math.Abs(\_tuSo), Math.Abs(\_mauSo));  \_tuSo /= ucln;  \_mauSo /= ucln;  if (\_mauSo < 0)  {  \_tuSo = -\_tuSo;  \_mauSo = -\_mauSo;  }  }  public static PhanSo operator+(PhanSo a, PhanSo b)  {  int tuSo = a.\_tuSo \* b.\_mauSo + b.\_tuSo \* a.\_mauSo;  int mauSo = a.\_mauSo \* b.\_mauSo;  PhanSo result = new PhanSo(tuSo, mauSo);  result.RutGon();  return result;  }  public static PhanSo operator-(PhanSo a, PhanSo b)  {  int tuSo = a.\_tuSo \* b.\_mauSo - b.\_tuSo \* a.\_mauSo;  int mauSo = a.\_mauSo \* b.\_mauSo;  PhanSo result = new PhanSo(tuSo, mauSo);  result.RutGon();  return result;  }  public static PhanSo operator\*(PhanSo a, PhanSo b)  {  int tuSo = a.\_tuSo \* b.\_tuSo;  int mauSo = a.\_mauSo \* b.\_mauSo;  PhanSo result = new PhanSo(tuSo, mauSo);  result.RutGon();  return result;  }  public static PhanSo operator/(PhanSo a, PhanSo b)  {  int tuSo = a.\_tuSo \* b.\_mauSo;  int mauSo = a.\_mauSo \* b.\_tuSo;  PhanSo result = new PhanSo(tuSo, mauSo);  result.RutGon();  return result;  }  }  } |

Bảng ‑

|  |
| --- |
| using System;  namespace LAB2.Bai04  {  public class DSPhanSo  {  private List<PhanSo> \_dSPhanSo;  public DSPhanSo()  {  \_dSPhanSo = new List<PhanSo>();  }  public void Nhap()  {  int n;  while(true)  {  Console.Write("Nhap so luong phan so: ");  string input = Console.ReadLine();  if (!int.TryParse(input, out n) || n <= 0)  Console.WriteLine("So luong khong hop le. Vui long nhap lai!");  else  break;  }  for (int i = 0; i < n; i++)  {  Console.WriteLine("Nhap phan so thu {0}:", i + 1);  PhanSo ps = new PhanSo();  ps.Nhap();  \_dSPhanSo.Add(ps);  }  }  public void Xuat()  {  foreach (var ps in \_dSPhanSo)  {  ps.Xuat();  Console.Write(' ');  }  Console.WriteLine();  }  public PhanSo TimPhanSoLonNhat()  {  if (\_dSPhanSo.Count == 0)  return null;  PhanSo maxPhanSo = \_dSPhanSo[0];  foreach (var ps in \_dSPhanSo)  {  if (ps.GetGiaTri() > maxPhanSo.GetGiaTri())  maxPhanSo = ps;  }  return maxPhanSo;  }  public void SapXepTangDan()  {  \_dSPhanSo.Sort((ps1, ps2) => ps1.GetGiaTri().CompareTo(ps2.GetGiaTri()));  }  }  } |

Bảng ‑

|  |
| --- |
| using System;  namespace LAB2.Bai04  {  class Program  {  static void Main(string[] args)  {  while (true)  {  Console.WriteLine("==========BAI 04 - LAB 2==========");  Console.WriteLine("1. Chuong trinh hai phan so.");  Console.WriteLine("2. Chuong trinh danh sach phan so.");  Console.WriteLine("0. Thoat.");  Console.Write("Nhap lua chon cua ban: ");  string choice = Console.ReadLine();  if (choice == "1")  ChuongTrinhHaiPhanSo();  else if (choice == "2")  ChuongTrinhDanhSachPhanSo();  else  return;  }  }  static void ChuongTrinhHaiPhanSo()  {  // Nhap vao hai phan so, xuat tong - hieu - tich - thuong cua hai phan so do  PhanSo \_phanSo1 = new PhanSo();  PhanSo \_phanSo2 = new PhanSo();  // Nhap phan so 1  Console.WriteLine("Nhap phan so thu 1:");  \_phanSo1.Nhap();  // Nhap phan so 2  Console.WriteLine("Nhap phan so thu 2:");  \_phanSo2.Nhap();  // Xuat hai phan so vua nhap  Console.Write("Phan so thu nhat la: ");  \_phanSo1.Xuat();  Console.Write("\nPhan so thu hai la: ");  \_phanSo2.Xuat();  // Tinh tong  Console.Write("\nTong hai phan so la: ");  (\_phanSo1 + \_phanSo2).Xuat();  // Tinh hieu  Console.Write("\nHieu hai phan so la: ");  (\_phanSo1 - \_phanSo2).Xuat();  // Tinh tich  Console.Write("\nTich hai phan so la: ");  (\_phanSo1 \* \_phanSo2).Xuat();  // Tinh thuong  Console.Write("\nThuong hai phan so la: ");  (\_phanSo1 / \_phanSo2).Xuat();  Console.WriteLine();  }  static void ChuongTrinhDanhSachPhanSo()  {  DSPhanSo \_dsPhanSo = new DSPhanSo();  // Nhap danh sach phan so  \_dsPhanSo.Nhap();  // Xuat danh sach phan so  Console.WriteLine("Danh sach phan so vua nhap:");  \_dsPhanSo.Xuat();  // Tim phan so lon nhat  Console.Write("Phan so lon nhat trong danh sach la: ");  PhanSo maxPhanSo = \_dsPhanSo.TimPhanSoLonNhat();  if (maxPhanSo != null)  maxPhanSo.Xuat();  else  Console.WriteLine("Danh sach rong.");  Console.WriteLine();  // Sap xep danh sach tang dan  \_dsPhanSo.SapXepTangDan();  Console.WriteLine("Danh sach phan so sau khi sap xep tang dan:");  \_dsPhanSo.Xuat();  }  }  } |

Bảng ‑

|  |  |
| --- | --- |
| Input:  1 4  3 4 |  |
| Input:  3 0  1 3  3 2 |  |
| Input:  -2 4  2 -4 |  |
| Input:  a b  a 5  3 b  3 4  3 5 |  |
| Input:  1 0  1 3  3 0  3 2 |  |
| Input:  100 100  900 900 |  |
| Input:  5 1  10 1 |  |

Bảng ‑

|  |  |
| --- | --- |
| Input:  3  1 2  -1 2  2 3 |  |
| Input:  4  1 3  -2 3  3 2  -1 3 |  |
| Input:  a  0  -5  3  1 1  2 1  -3 2 |  |

## 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:

***- 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 input/output của bài toán:

* Input:

+ Số nguyên n là số các thị trường kinh doanh cần quản lý. Trong n dòng tiếp theo, mỗi dòng nhập loại thị trường kinh doanh và các thông tin cần quản lý liên quan đến thị trường đó.

+ Các thông tin cần tìm kiếm (địa điểm, giá, diện tích).

* Output:

+ Danh sách các thông tin thị trường(Khu đất, nhà phố, chung cư).

+ Tổng giá bán theo từng loại (Khu đất, nhà phố, chung cư).

+ Danh sách các khu đất thỏa mãn diện tích > 100 m2 hoặc các nhà phố thỏa mãn diện tích > 60 m2 và năm xây dựng từ 2019 trở về sau.

+ Danh sách các nhà phố hoặc chung cư thỏa mãn điều kiện tìm kiếm (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).

Bảng ‑

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace LAB2.Bai05  {  public abstract class ThiTruongKD  {  protected string \_diaDiem;  protected long \_giaBan;  protected float \_dienTich;  // Getters/Setters  public string DiaDiem  {  get { return \_diaDiem; }  set { \_diaDiem = value; }  }  public long GiaBan  {  get { return \_giaBan; }  set { \_giaBan = value; }  }  public float DienTich  {  get { return \_dienTich; }  set { \_dienTich = value; }  }  public virtual void Nhap()  {  // Nhap dia diem  Console.Write("Nhap dia diem: ");  \_diaDiem = Console.ReadLine();  // Nhap gia ban  while (true)  {  Console.Write("Nhap gia ban (> 0, VND): ");  long giaBanInput;  if (long.TryParse(Console.ReadLine(), out giaBanInput) && giaBanInput >= 0)  {  \_giaBan = giaBanInput;  break;  }  else  {  Console.WriteLine("Gia ban khong hop le. Vui long nhap lai.");  }  }  // Nhap dien tich  while (true)  {  Console.Write("Nhap dien tich (> 0, m2): ");  float dienTichInput;  if (float.TryParse(Console.ReadLine(), out dienTichInput) && dienTichInput >= 0)  {  \_dienTich = dienTichInput;  break;  }  else  {  Console.WriteLine("Dien tich khong hop le. Vui long nhap lai.");  }  }  }  public virtual void Xuat()  {  Console.Write("Dia diem: " + \_diaDiem +  " | Gia ban: " + \_giaBan + " VND" +  " | Dien tich: " + \_dienTich + " m2");  }  }  public class KhuDat : ThiTruongKD { }  public class NhaPho : ThiTruongKD  {  private int \_namXayDung;  private int \_soTang;  // Getters/Setters  public int NamXayDung  {  get { return \_namXayDung; }  set { \_namXayDung = value; }  }  public int SoTang  {  get { return \_soTang; }  set { \_soTang = value; }  }  public override void Nhap()  {  base.Nhap();  // Nhap nam xay dung  while (true)  {  Console.Write("Nhap nam xay dung (> 0): ");  int namXayDungInput;  if (int.TryParse(Console.ReadLine(), out namXayDungInput) && namXayDungInput >= 0)  {  \_namXayDung = namXayDungInput;  break;  }  else  {  Console.WriteLine("Nam xay dung khong hop le. Vui long nhap lai.");  }  }  // Nhap so tang  while (true)  {  Console.Write("Nhap so tang (> 0): ");  int soTangInput;  if (int.TryParse(Console.ReadLine(), out soTangInput) && soTangInput >= 0)  {  \_soTang = soTangInput;  break;  }  else  {  Console.WriteLine("So tang khong hop le. Vui long nhap lai.");  }  }  }  public override void Xuat()  {  base.Xuat();  Console.Write(" | Nam xay dung: " + \_namXayDung +  " | So tang: " + \_soTang);  }  }  public class ChungCu : ThiTruongKD  {  private int \_tang;  public override void Nhap()  {  base.Nhap();  // Nhap tang  while (true)  {  Console.Write("Nhap tang (> 0): ");  int tangInput;  if (int.TryParse(Console.ReadLine(), out tangInput) && tangInput >= 0)  {  \_tang = tangInput;  break;  }  else  {  Console.WriteLine("Tang khong hop le. Vui long nhap lai.");  }  }  }  public override void Xuat()  {  base.Xuat();  Console.Write(" | Tang: " + \_tang);  }  }  } |

Bảng ‑

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace LAB2.Bai05  {  public class QuanLyKinhDoanh  {  private List<ThiTruongKD> \_danhSachThiTruong;  public QuanLyKinhDoanh()  {  \_danhSachThiTruong = new List<ThiTruongKD>();  }  public void ThemThiTruong()  {  int choice;  ThiTruongKD thiTruong = null;  while (true)  {  Console.WriteLine("Chon loai thi truong (1. Khu Dat, 2. Nha Pho, 3. Chung Cu): ");  Console.Write("Lua chon cua ban (1-3): ");  if (int.TryParse(Console.ReadLine(), out choice) && choice >= 1 && choice <= 3)  break;  else  Console.WriteLine("Lua chon khong hop le. Vui long chon lai.");  }  switch (choice)  {  case 1:  thiTruong = new KhuDat();  break;  case 2:  thiTruong = new NhaPho();  break;  case 3:  thiTruong = new ChungCu();  break;  }  thiTruong.Nhap();  \_danhSachThiTruong.Add(thiTruong);  }  public void Nhap()  {  int soLuong;  while (true)  {  Console.Write("Nhap so luong thi truong can them: ");  if (int.TryParse(Console.ReadLine(), out soLuong) && soLuong > 0)  break;  else  Console.WriteLine("So luong khong hop le. Vui long nhap lai.");  }  for (int i = 0; i < soLuong; i++)  {  Console.WriteLine($"\nNhap thong tin cho thi truong thu {i + 1}:");  ThemThiTruong();  }  }  public void Xuat()  {  int i = 1;  foreach(ThiTruongKD thiTruong in \_danhSachThiTruong)  {  Console.Write(i++ + ". ");  thiTruong.Xuat();  Console.WriteLine();  }  }  public long TinhTongGiaBan()  {  long tongGiaBan = 0;  foreach (var thiTruong in \_danhSachThiTruong)  {  tongGiaBan += thiTruong.GiaBan;  }  return tongGiaBan;  }  public long TinhTongGiaBanKhuDat()  {  long tongGiaBan = 0;  foreach (var thiTruong in \_danhSachThiTruong)  {  if(thiTruong is KhuDat)  tongGiaBan += thiTruong.GiaBan;  }  return tongGiaBan;  }  public long TinhTongGiaBanNhaPho()  {  long tongGiaBan = 0;  foreach (var thiTruong in \_danhSachThiTruong)  {  if (thiTruong is NhaPho)  tongGiaBan += thiTruong.GiaBan;  }  return tongGiaBan;  }  public long TinhTongGiaBanChungCu()  {  long tongGiaBan = 0;  foreach (var thiTruong in \_danhSachThiTruong)  {  if (thiTruong is ChungCu)  tongGiaBan += thiTruong.GiaBan;  }  return tongGiaBan;  }  public void TimKiemNhaPhoHoacChungCuTheoThongTinCungCap(string diaDiem,long gia, int dienTich)  {  bool found = false;  foreach(ThiTruongKD thiTruong in \_danhSachThiTruong)  {  if (thiTruong is ChungCu || thiTruong is NhaPho)  {  if(thiTruong.DiaDiem.ToLower().Contains(diaDiem.ToLower())  && thiTruong.GiaBan <= gia && thiTruong.DienTich >= dienTich)  {  thiTruong.Xuat();  found = true;  Console.WriteLine();  }  }  }  if(!found)  {  Console.WriteLine("Khong tim thay nha pho hay chung cu thoa man dieu kien tim kiem!");  }  }  public void TimKiemKhuDatHoacNhaPhoTheoDieuKienDatBieṭ()  {  // Dieu kien dac biet:  // // khu dat co dien tich > 100 m2 hoac  // // nha pho co dien tich > 60 m2 va nam xay dung >= 2019  bool found = false;  foreach(ThiTruongKD thiTruong in \_danhSachThiTruong)  {  if(thiTruong is KhuDat khuDat && khuDat.DienTich > 100)  {  thiTruong.Xuat();  found = true;1  Console.WriteLine();  }  else if(thiTruong is NhaPho nhaPho  && nhaPho.DienTich > 60  && nhaPho.NamXayDung >= 2019)  {  thiTruong.Xuat();  found = true;  Console.WriteLine();  }  if(!found)  {  Console.WriteLine("Khong tim thay khu dat va nha pho phu hop.");  }  }  }  }  } |

Bảng ‑

|  |
| --- |
| using System;  using System.Runtime.CompilerServices;  namespace LAB2.Bai05  {  class Program  {  static void Main(string[] args)  {  QuanLyKinhDoanh quanLyKinhDoanh = new QuanLyKinhDoanh();  while (true)  {  Console.WriteLine("==========BAI 05 - LAB 2==========");  Console.WriteLine("1. Nhap danh sach thi truong.");  Console.WriteLine("2. Xuat danh sach thi truong.");  Console.WriteLine("3. Tinh tong gia ban tat ca thi truong.");  Console.WriteLine("4. Tinh tong gia ban khu dat.");  Console.WriteLine("5. Tinh tong gia ban nha pho.");  Console.WriteLine("6. Tinh tong gia ban chung cu.");  Console.WriteLine("7. Tim kiem nha pho hoac chung cu theo dia diem, gia ban va dien tich.");  Console.WriteLine("8. Tim kiem khu dat co dien tich < 100m2 hoac nha pho co dien tich > 60m2 va nam xay dung >= 2019.");  Console.WriteLine("0. Thoat.");  Console.Write("Nhap lua chon cua ban: ");  string choice = Console.ReadLine();  switch (choice)  {  case "1":  quanLyKinhDoanh.Nhap();  break;  case "2":  quanLyKinhDoanh.Xuat();  break;  case "3":  Console.WriteLine($"Tong gia ban tat ca thi truong: {quanLyKinhDoanh.TinhTongGiaBan()}");  break;  case "4":  Console.WriteLine($"Tong gia ban khu dat: {quanLyKinhDoanh.TinhTongGiaBanKhuDat()}");  break;  case "5":  Console.WriteLine($"Tong gia ban nha pho: {quanLyKinhDoanh.TinhTongGiaBanNhaPho()}");  break;  case "6":  Console.WriteLine($"Tong gia ban chung cu: {quanLyKinhDoanh.TinhTongGiaBanChungCu()}");  break;  case "7":  Console.Write("Nhap dia diem can tim: ");  string diaDiem = Console.ReadLine();  Console.Write("Nhap gia can tim: ");  int gia = int.Parse(Console.ReadLine());  Console.Write("Nhap dien tich can tim: ");  int dienTich = int.Parse(Console.ReadLine());  Console.Write("Thong tin nha pho hoac chung cu tim duoc:\n");  quanLyKinhDoanh.TimKiemNhaPhoHoacChungCuTheoThongTinCungCap(diaDiem, gia, dienTich);  break;  case "8":  Console.Write("Thong tin khu dat hoac nha pho tim duoc:\n");  quanLyKinhDoanh.TimKiemKhuDatHoacNhaPhoTheoDieuKienDatBieṭ();  break;  default:  return;  }  }  }  }  } |

Hình

Bảng ‑

|  |  |
| --- | --- |
| Testcase 1:  Input:  Nhập danh sách gồm 3 thị trường.  + Thị trường 1 là khu đất: địa điểm TPHCM, giá bán 1500000000 VND, diện tích 100 m2.  + Thị trường 2 là nhà phố: địa điểm Cà Mau, giá bán 1200000000 VND, diện tích 80 m2, năm xây dựng 2020, số tầng 2.  + Thị trường 3 là chung cư: địa điểm TPHCM, giá bán 3000000000 VND, diện tích 120 m2, tầng 6. | - Nhập/xuất danh sách các thị trường kinh doanh.      - Giá bán của từng loại thị trường:    - Danh sách các khu đất thỏa mãn diện tích > 100 m2 hoặc các nhà phố thỏa mãn diện tích > 60 m2 và năm xây dựng >= 2019.    - Danh sách các nhà phố hoặc chung cư thỏa mãn điều kiện tìm kiếm (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) |
|  |  |