**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**

**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**Giảng viên hướng dẫn: TS. Lê Văn Vinh**

**Sinh viên thực hiện: Nguyễn Phúc An 19110312**

**Bành Đăng Khoa 19110378**

TP. Hồ Chí Minh, tháng 12 năm 2020

Bảng Phân công công việc:

|  |  |
| --- | --- |
| Tên Thành Viên | Công Việc |
| Nguyễn Phúc An - 19110321 | Bài 1:Class Hinh và kế thừa của class Hinh, class Point, class Menu, class GRAPHIC.  Bài 2: Viết bằng ngôn ngữ C#.Viết bằng ngôn ngữ C++. Viết class NhanVien cho Python. |
| Bành Đăng Khoa - 19110378 | Bài 1: Class Complexobject, viết hàm tạo bảng cho MENU.  Bài 2: Viết bằng ngôn ngữ Java, viết hàm tạo bảng cho bên C#. Viết bằng ngôn ngữ Python. |

Ngôn ngữ: C#, Java,C++, Python

Tính chất hướng đối tượng:

* Tính đóng gói: các class Hinh, Xe,NhanVien ….
* Tính kế thừa: HinhChuNhat,HinhVuong,…, kế thừa từ Hinh.
  + - Xe2Banh,Xe4Banh,Xe4BanhNhapKhau,Xe2BanhNhapKhau, kế thừa từ Xe.
    - Sale, ThoSuaXe, BaoVe kế thừa từ NhanVien.
* Tính đa hình: Các class Xe2Banh,Xe4Banh,.., đa hình từ Xe.
  + - Các class Sale, ThoSuaXe, BaoVe đa hình từ NhanVien.
    - Các class HinhChuNhat,HinhVuong,…, đa hình từ Hinh.
* Static class: Menu, CuaHang.
* Linq: Sắp xếp xe theo giá tiền và sắp xếp nhân viên theo lương.
* Operator overloading: Trong C++.

**Bài 1**:

Danh sách các class, các biến, phương thức, hàm của từng class:

* Class POINT

- Hàm Input(): Nhập tọa độ cho điểm.

- Hàm Output(): Xuất tọa độ cho điểm.

* Class Hinh

- Hàm virtual Nhap(): Nhập tọa độ cho 2 điểm X và Y, nhập vào màu.

- Hàm virtual Xuat(): Xuất tọa độ cho 2 điểm X và Y, màu của hình.

- Hàm virtual Ve(): Hàm cơ bản để override.

- Hàm virtual TinhDienTich(): Gán diện tích = 0 để override.

- Hàm virtual DoiMau(): Để đổi màu.

- Hàm virtual DoiMau(int a):

- Hàm virtual Move(Point NP1): Tịnh tiến hình theo vecto tọa độ NP1.

* Class DoanThang : Hinh

- Hàm override Nhap(): Chỉnh lại cho phù hợp tên hình.

- Hàm override Xuat(): Chỉnh lại cho phù hợp tên hình.

- Hàm override Ve(): Chỉnh lại cho phù hợp tên hình.

* Class HinhChuNhat : Hinh

- Hàm override Nhap(): Chỉnh lại cho phù hợp tên hình, gọi các hàm TinhChieuDaiChieuRong(), hàm TinhDienTich().

- Hàm override Xuat(): Chỉnh lại cho phù hợp theo hình, xuất thêm thông tin về chiều dài, chiều rộng và diện tích.

- Hàm TinhChieuDaiChieuRong(): Tính chiều dài và chiều rộng theo công thức:

ChieuDai = Math.Abs(P1.X - P2.X);

ChieuRong = Math.Abs(P1.Y - P2.Y);

- Hàm override TinhDienTich(): DienTich = ChieuDai \* ChieuRong.

- Hàm override Ve(): Chỉnh lại cho phù hợp với hình.

* Class HinhTron : Hinh

- Hàm override Nhap(): Chỉnh lại cho phù hợp tên hình, gọi các hàm TinhBanKinh(), hàm TinhDienTich().

- Hàm override Ve(): Chỉnh lại cho phù hợp với hình.

- Hàm TinhBanKinh(): Tính bán kính theo công thức:

BanKinh = Math.Abs(P1.X - P2.X) / 2;

- Hàm override Xuat(): Chỉnh lại cho phù hợp theo hình, xuất thêm thông tin về bán kính và diện tích.

- Hàm override TinhDienTich(): DienTich = Math.PI \* Math.Pow(BanKinh,2);

* Class HinhVuong : Hinh

- Hàm override Nhap(): Chỉnh lại cho phù hợp tên hình, gọi các hàm TinhChieuDaiCanh(), hàm TinhDienTich().

- Hàm override Ve(): Chỉnh lại cho phù hợp với hình.

- Hàm TinhChieuDaiCanh():

ChieuDaiCanh = Math.Abs(P1.X - P2.X);

- Hàm override Xuat(): Chỉnh lại cho phù hợp theo hình, xuất thêm thông tin về chiều dài cạnh và diện tích.

- Hàm override TinhDienTich(): DienTich = ChieuDaiCanh \* ChieuDaiCanh.

* Class HinhTamGiac() : Hinh

- Hàm override Nhap(): Chỉnh lại cho phù hợp tên hình, gọi các hàm TinhChieuCaoDoDaiDay(), hàm TinhDienTich().

- Hàm override Ve(): Chỉnh lại cho phù hợp với hình.

- Hàm TinhChieuCaoDoDaiDay():

ChieuCao = Math.Abs(P1.Y - P2.Y);

DoDaiDay = Math.Abs(P1.X - P2.Y);

- Hàm override Xuat(): Chỉnh lại cho phù hợp theo hình, xuất thêm thông tin về chiều cao, độ dài đáy và diện tích.

- Hàm override TinhDienTich(): DienTich = (DoDaiDay \* ChieuCao) / 2;

* Class HinhLucGiacDeu : Hinh

- Hàm override Nhap(): Chỉnh lại cho phù hợp tên hình, gọi các hàm TinhDoDaiCanh(), hàm TinhDienTich().

- Hàm TinhDoDaiCanh():

DoDaiCanh = Math.Abs(P1.X - P2.X) / 2;

- Hàm override Xuat(): Chỉnh lại cho phù hợp theo hình, xuất thêm thông tin về độ dài cạnh và diện tích.

- Hàm override Ve(): Chỉnh lại cho phù hợp với hình.

- Hàm override TinhDienTich(): DienTich = 3\*Math.Sqrt(3) \* DoDaiCanh \* DoDaiCanh \* 0.5;

* Class Graphic

- Hàm Add(Hinh newHinh): Thêm newHinh vào LHinh.

- Hàm Merge(List<Hinh> newLHinh): Gọp cái hình trong newLHinh vào một Complexobject vừa khởi tạo và xóa các hình trong newLHinh ra khỏi Lhinh.

- Hàm Divide(Hinh A): Tách các hình trong complexobject A ra khỏi complexobject và thêm vào lại Lhinh.

- Hàm Xuat(): Xuất thông tin các hình đang có trong LHinh.

* Class COMPLEXOBJECT : Hinh

- Hàm override Nhap(): Rỗng.

- Hàm SetCordiante(): Chỉnh lại tọa độ mỗi khi thêm một hình vô Complexobject.

- Hàm Push(Hinh newHinh): Thêm newHinh vào Complexobject.

- Hàm Pop(): Lấy hình ra khỏi Complexobject.

- Hàm override Move(Point NP1): Tịnh tiến tất cả các hình đang có trong Complexobject theo vector tọa độ NP1.

- Hàm override Xuat(): Xuất thông tin của Complexobject, tổng diện tích các hình trong Complexobject và thông tin từng hình trong Complexobject.

* Class static Menu:

- Hàm static Run(): Hàm tạo menu.

- Hàm static Them(): Tạo các hình và thêm nó vào LHinh của Graphic.

- Hàm static Xuat(): Xuất thông tin của hình người dùng chọn đang có trong LHinh của Graphic.

- Hàm static Gop(): Cho người dùng gọp các hình thành 1 Complexobject và thêm Complexobject vào LHinh.

- Hàm static Tach(): Tách các hình trong 1 Complexobject ra và thêm ngược lại vào LHinh của Graphic.

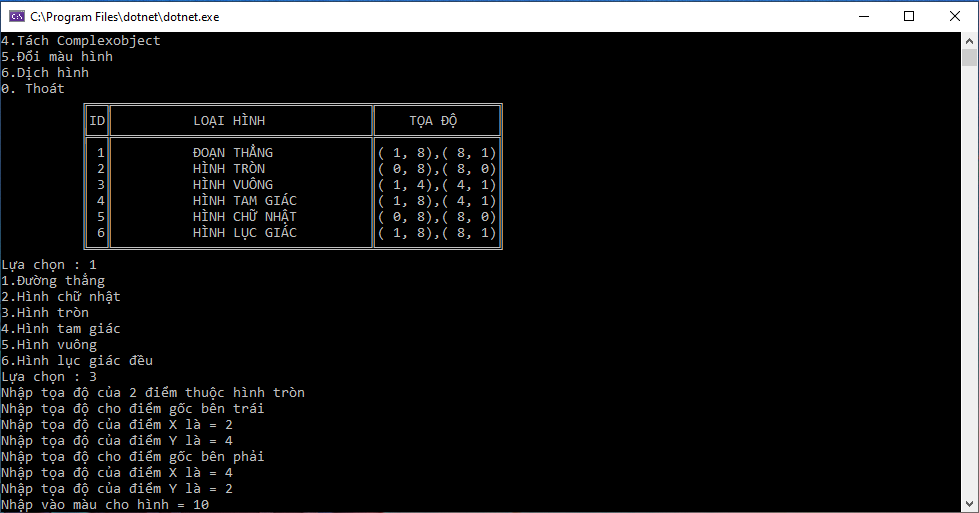
- Hàm static DoiMau(): Đổi màu cho 1 hình người dùng chọn.

- Hàm static Table(): Dùng để vẽ ra bảng chứa thông tin các hình.

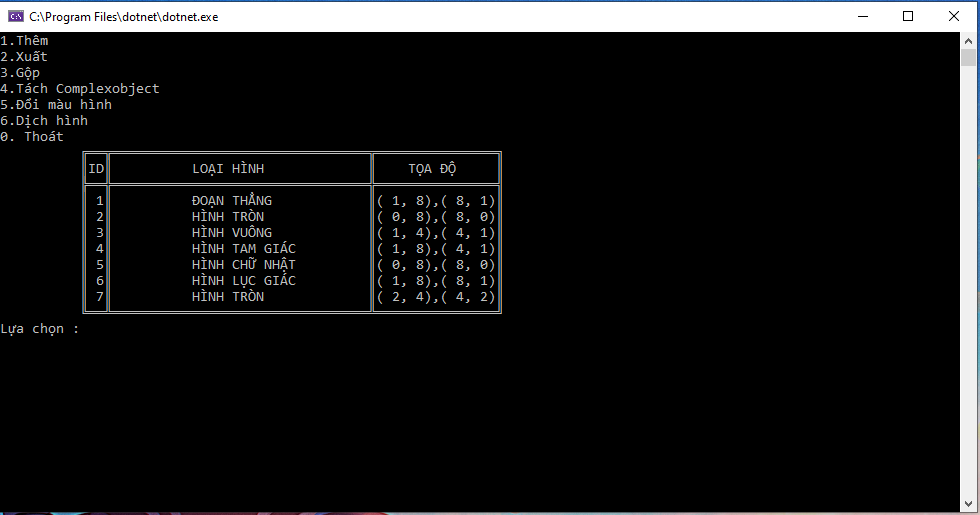
- Hàm static DefaultData(): Tạo bộ dữ liệu sẵn.

Hàm Them():

Khi chọn lệnh thêm :

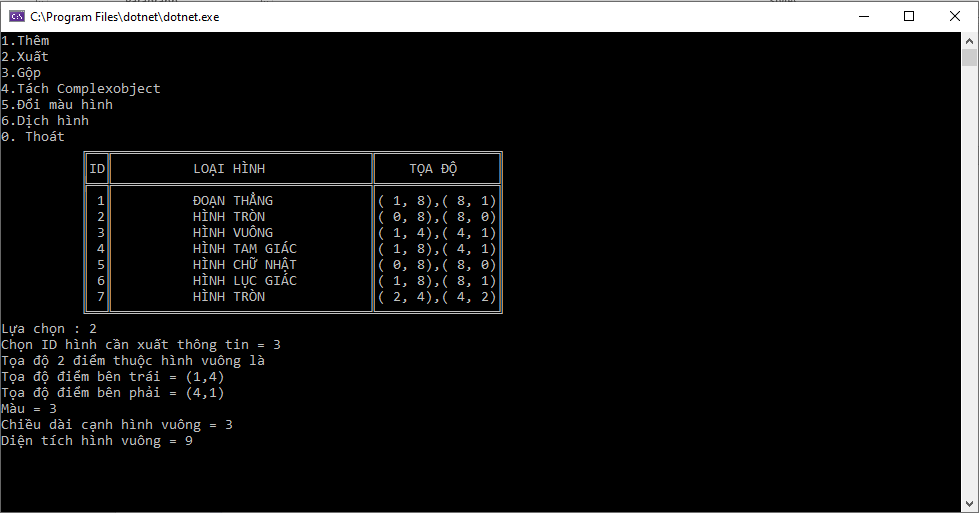


Kết quả :



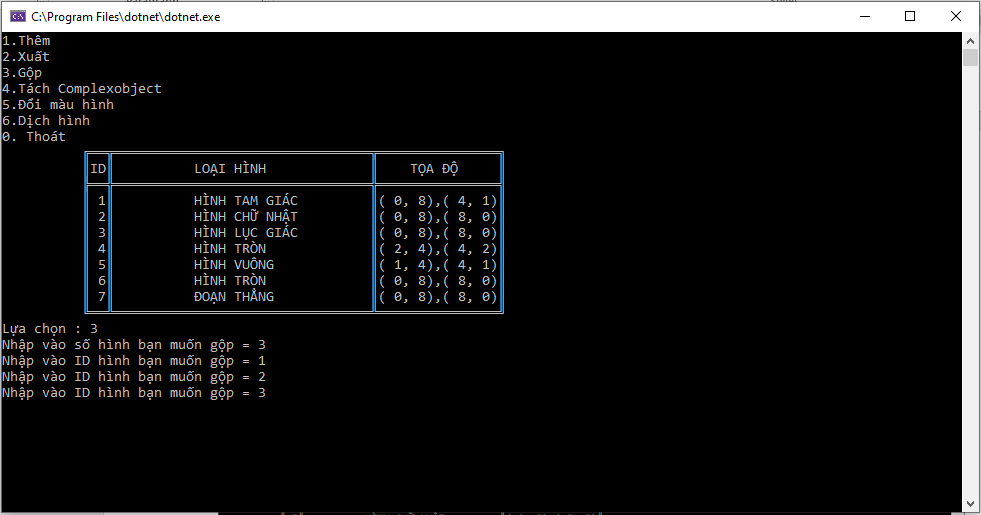
Hàm Xuat():

Khi chọn lệnh Xuất :

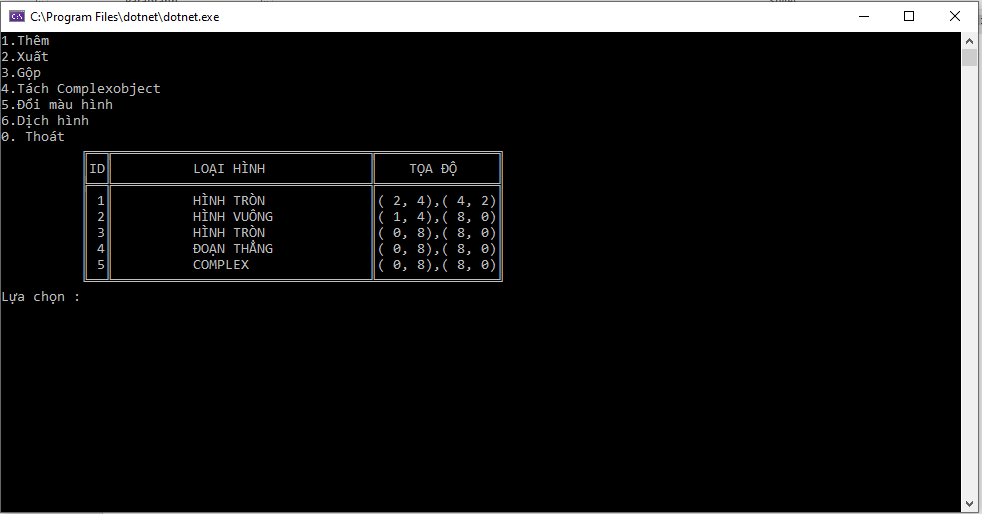


Hàm Gop():

Khi chọn lệnh Gộp :

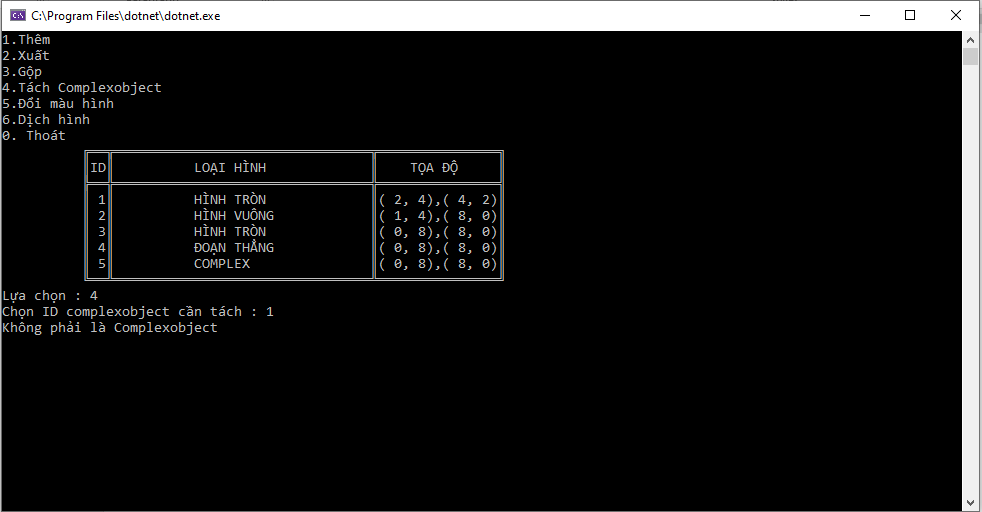


Kết quả :

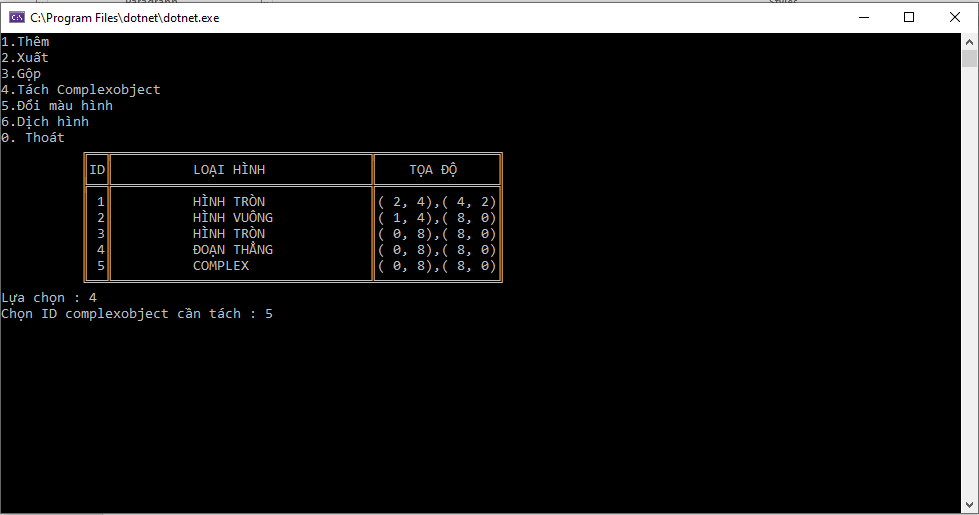


Hàm Tach():

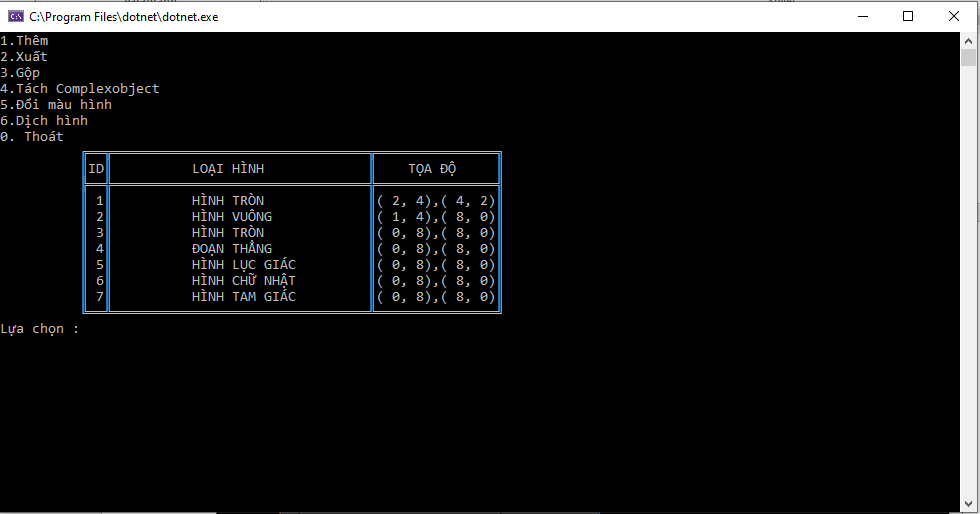
Khi chọn lệnh Tách Complexobject và chọn hình không phải là complexobject :



Khi chọn lệnh Tách Complexobject với chọn đúng ComplexObject :

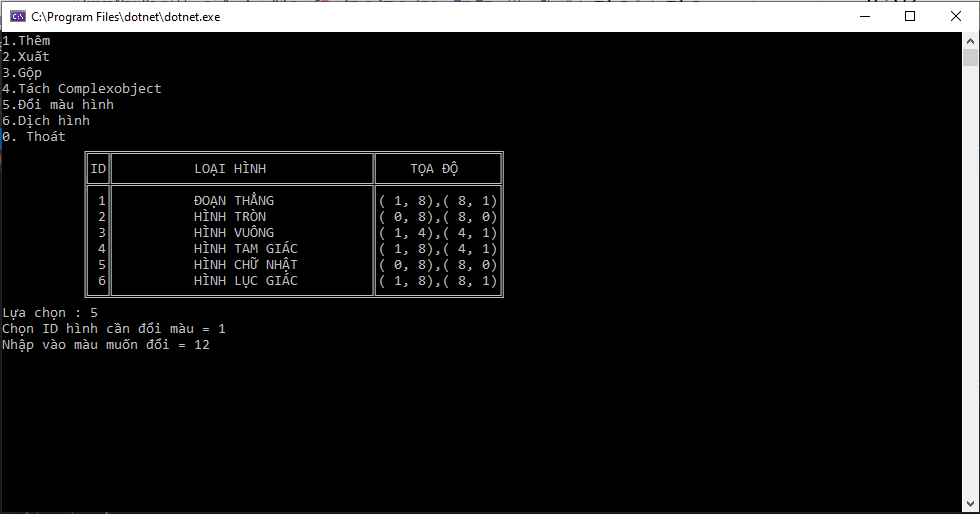


Kết quả :

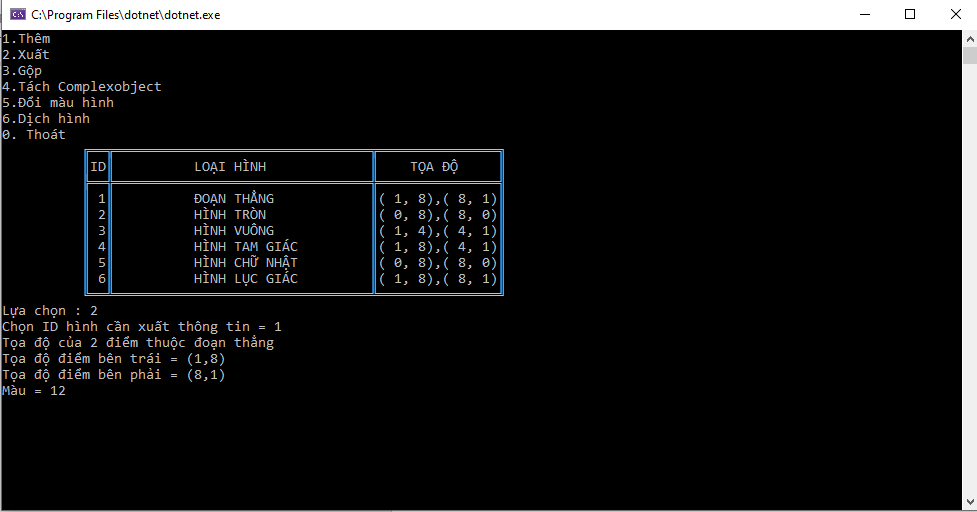


Hàm DoiMau():

Khi chọn lệnh đổi màu hình :

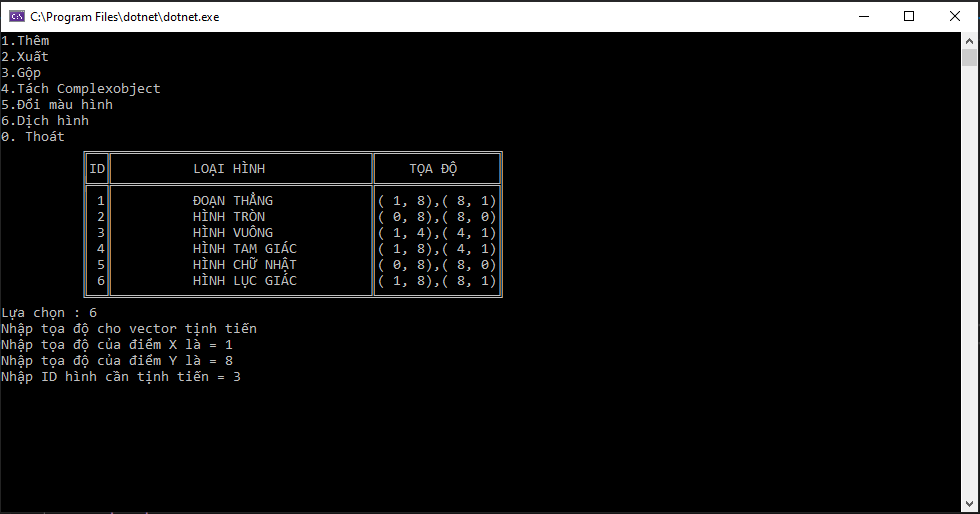


Kết quả :

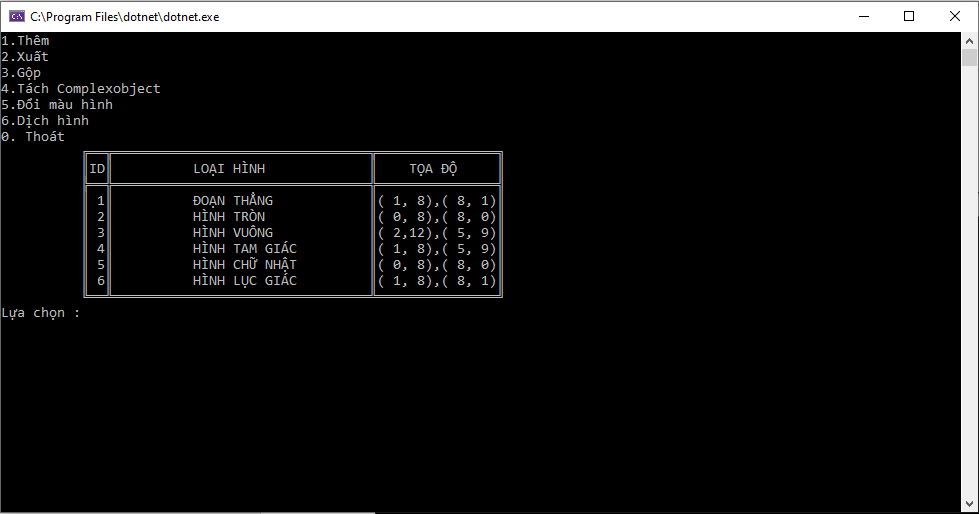


Hàm Dich():

Khi chọn lệnh Dịch hình :



Kết quả :



**Bài 2:**

* Class Xe

- Hàm virtual Nhap(): Nhập thông tin cho xe.

- Hàm virtual Xuat(): Xuất thông tin của xe.

- Hàm virtual TinhGiaTien(): Rỗng.

* Class Xe2Banh : Xe

- Hàm Nhap(): Nhập thông tin cho xe 2 bánh,

- Hàm Xuat(): Xuất thông tin của xe.

- Hàm TinhGiaTien(): Tính giá tiền của xe theo công thức:

if (Displacement > 175)

SalePrice = Price + Price \* 0.2;

else

SalePrice = Price;

* Class Xe2BanhNhapKhau : Xe2Banh

- Hàm Nhap(): Nhập thông tin cho xe 2 bánh, nhập xuất xứ.

- Hàm Xuat(): Xuất thông tin của xe.

- Hàm TinhGiaTien(): Tính giá tiền của xe theo công thức:

if (Displacement > 175)

SalePrice = Price + Price \* 0.2 + Price \* 0.35;

else

SalePrice = Price + Price \* 0.35;

* Class Xe4Banh : Xe

- Hàm Nhap(): Nhập thông tin cho xe 4 bánh, nhập số ghế ngồi trên xe.

- Hàm Xuat(): Xuất thông tin của xe.

- Hàm TinhGiaTien(): Tính giá tiền của xe theo công thức:

if (Displacement > 1500)

SalePrice = Price + Price \* 0.5;

else

SalePrice = Price;

* Class Xe4BanhNhapKhau : Xe4Banh

- Hàm Nhap(): Nhập thông tin cho xe 4 bánh nhập khẩu, nhập số ghế ngồi, nhập xuất xứ của xe.

- Hàm Xuat(): Xuất thông tin của xe.

- Hàm TinhGiaTien(): Tính giá tiền của xe theo công thức :

if (Displacement > 1500)

SalePrice = Price + Price \* 0.5 + Price \* 0.7;

else

SalePrice = Price + Price \* 0.7;

* Class Xe4BanhNhapKhau : Xe4Banh

- Hàm Nhap(): Nhập thông tin cho xe 4 bánh nhập khẩu, nhập số ghế ngồi, nhập xuất xứ của xe.

- Hàm Xuat(): Xuất thông tin của xe.

- Hàm TinhGiaTien(): Tính giá tiền của xe theo công thức :

if (Displacement > 1500)

SalePrice = Price + Price \* 0.5 + Price \* 0.7;

else

SalePrice = Price + Price \* 0.7;

* Class NhanVien

- Hàm virtual Nhap(): Nhập thông tin cho nhân viên.

- Hàm virtual Xuat(): Xuất thông tin của nhân viên.

- Hàm TinhLuong(): Rỗng.

* Class Sale : NhanVien

- Hàm Nhap(): Nhập thông tin cho nhân viên Sale.

- Hàm virtual Xuat(): Xuất thông tin của nhân viên Sale.

- Hàm virtual TinhLuong(): Tính lương của nhân viên Sale theo công thức:

Salary = Salary + SoLuongXeBanDuoc \* 500000;

* Class ThoSuaXe : NhanVien

- Hàm Nhap(): Nhập thông tin của thợ sửa xe.

- Hàm Xuat(): Xuất thông tin của thợ sửa xe.

- Hàm TinhLuong(): Tính giá tiền của thợ sửa xe theo công thức:

Salary = Salary + SoXeDaSua \* 100000;

* Class BaoVe : NhanVien

- Hàm Nhap(): Nhập thông tin của nhân viên bảo vệ.

- Hàm Xuat(): Xuất thông tin của nhân viên bảo vệ.

- Hàm TinhLuong(): Tính giá tiền của nhân viên bảo vệ theo công thức:

Salary = Salary + SoCaLamThem \* 185000;

* Class static CuaHang

- Hàm TinhTongGiaTriKhoHang(): Tính tổng giá trị của các xe đang bán ở của hàng.

- Hàm TinhTongGiaTriTheoXe(): Tính tổng giá trị theo từng loại xe.

- Hàm TimKiemXeTheoModel(): Tìm model xe cần tìm và xuất ra thông tin của xe đó.

- Hàm TimKiemNhanVien(): Tìm kiếm nhân viên theo họ hoặc họ và tên.

- Hàm TinhTongLuong(): Tính tổng lương của nhân viên.

- Hàm SapXepXe(): Sắp xếp xe theo giá tăng dần.

- Hàm SapXepNhanVien(): Sắp xếp nhân viên theo lương tăng dần.

* Class static MENU

- Hàm RUN(): Tạo MENU.

- Hàm Them(): Cho người dùng thêm nhân viên hoặc thêm xe theo ý muốn vào cửa hàng.

- Hàm Xuat(): Cho người dùng xuất thông tin nhân viên hoặc xe theo ý muốn.

- Hàm TimKiemXe(): Gọi lại hàm TimKiemXeTheoModel() ở class CuaHang.

- Hàm TimKiemNhanVien(): Gọi lại hàm TimKiemNhanVien() ở class CuaHang.

- Hàm SapXepNhanVien(): Gọi lại hàm SapXepNhanVien() ở class CuaHang.

- Hàm SapXepXe(): Gọi lại hàm SapXepXe() ở class CuaHang.

- Hàm Fill(int x): thêm x lần “ ” vào vị trí con trỏ hiện tại.

- Hàm Table(): Tạo bảng tổng hợp của cả xe 2 bánh và xe 4 bánh.

- Hàm Table2(list<Xe> LXTV): tạo bảng theo list xe truyền vào.

- Hàm TableNV(): Tạo bảng chứa thông tin nhân viên.

- Hàm DanhSachXe2Banh(): Sử dụng hàm Table2 và cho người dùng xuất thông tin đang có ở trên bảng.

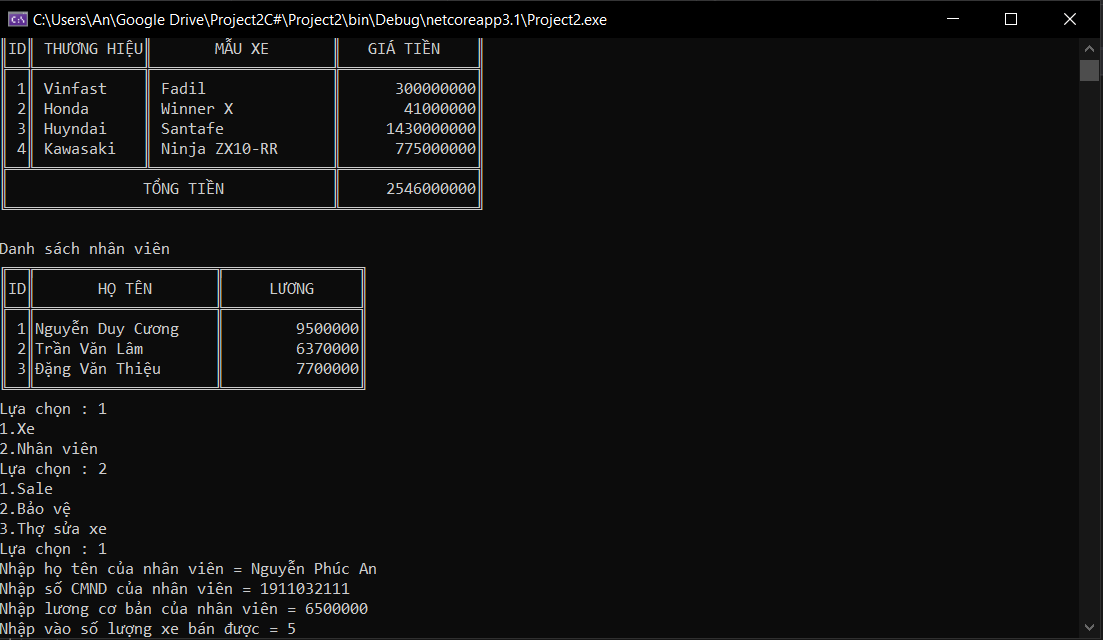
- Hàm DanhSachXe4Banh(): Sử dụng hàm Table2 và cho người dùng xuất thông tin đang có ở trên bảng.

- Hàm DefaultData(): Tạo dữ liệu sẵn cho chương trình.

Hàm Them():

Cho người dùng thêm xe hoặc nhân theo lựa chọn vào của hàng :

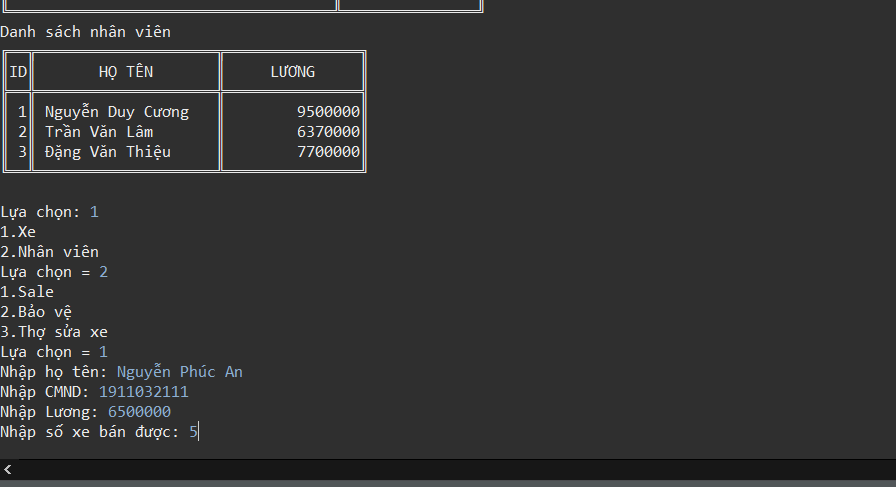
C#



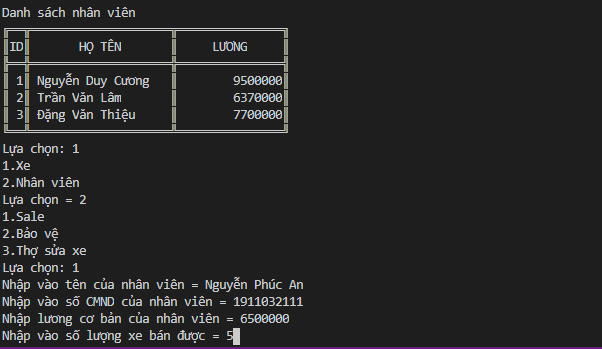
C++



Java

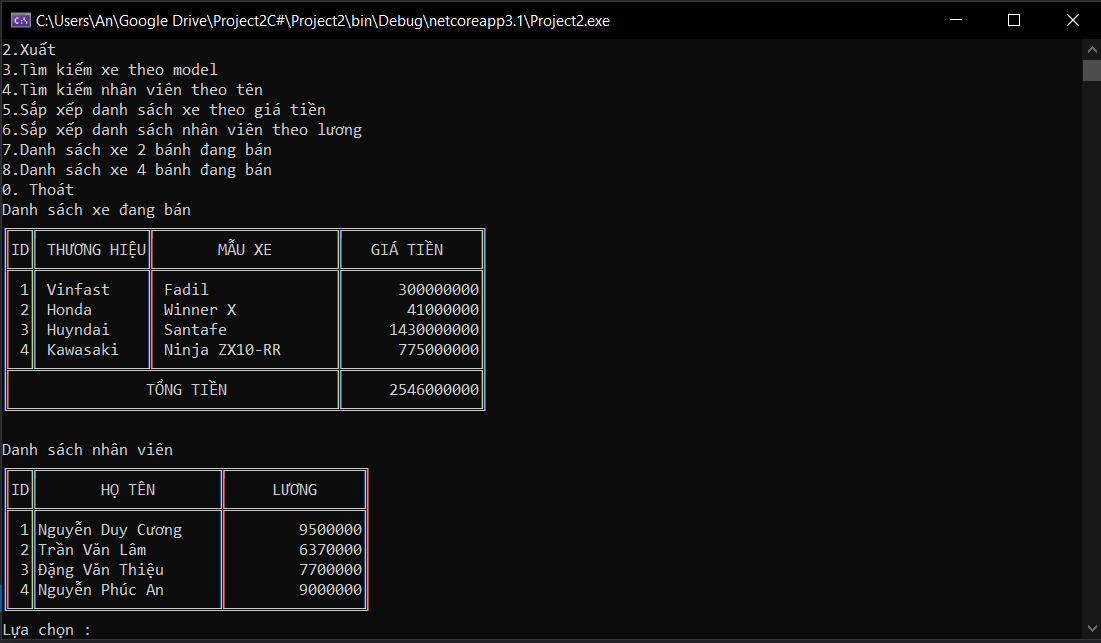


Python

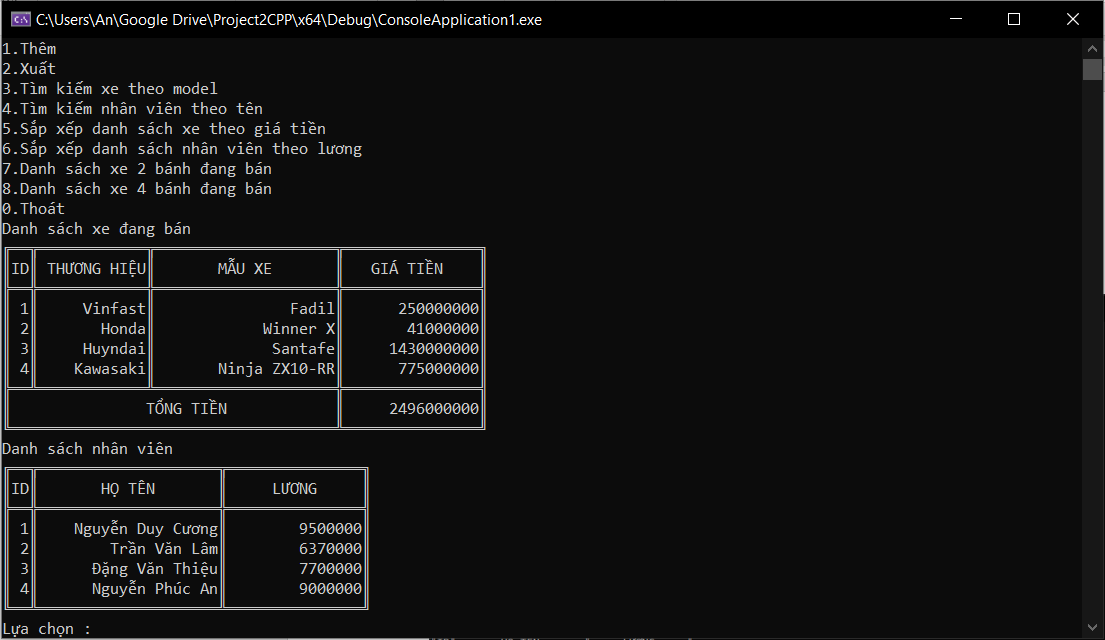


Kết quả :

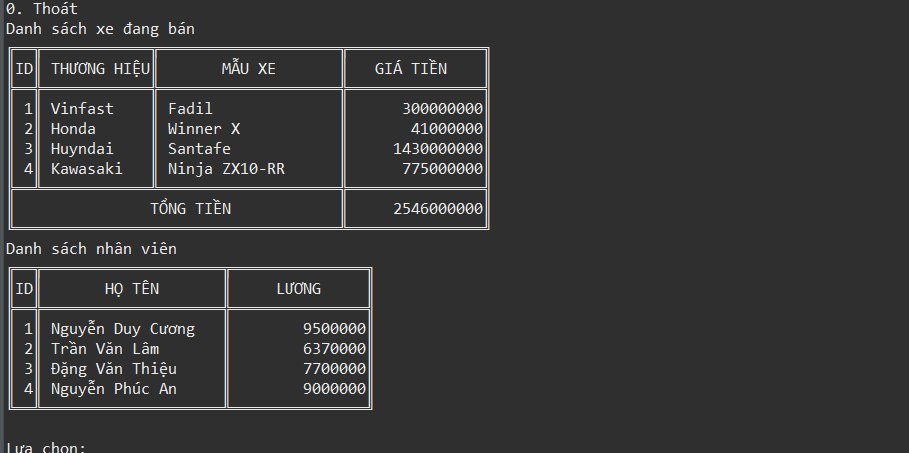
C#



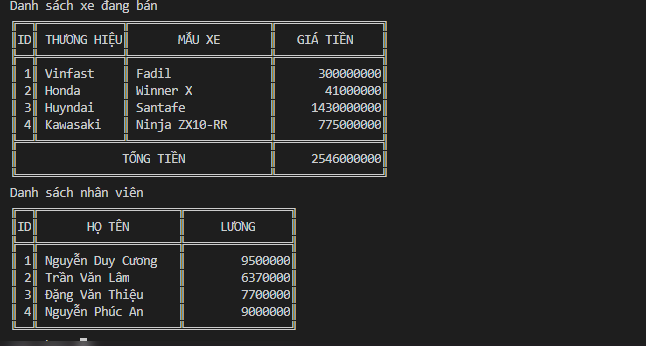
C++



Java



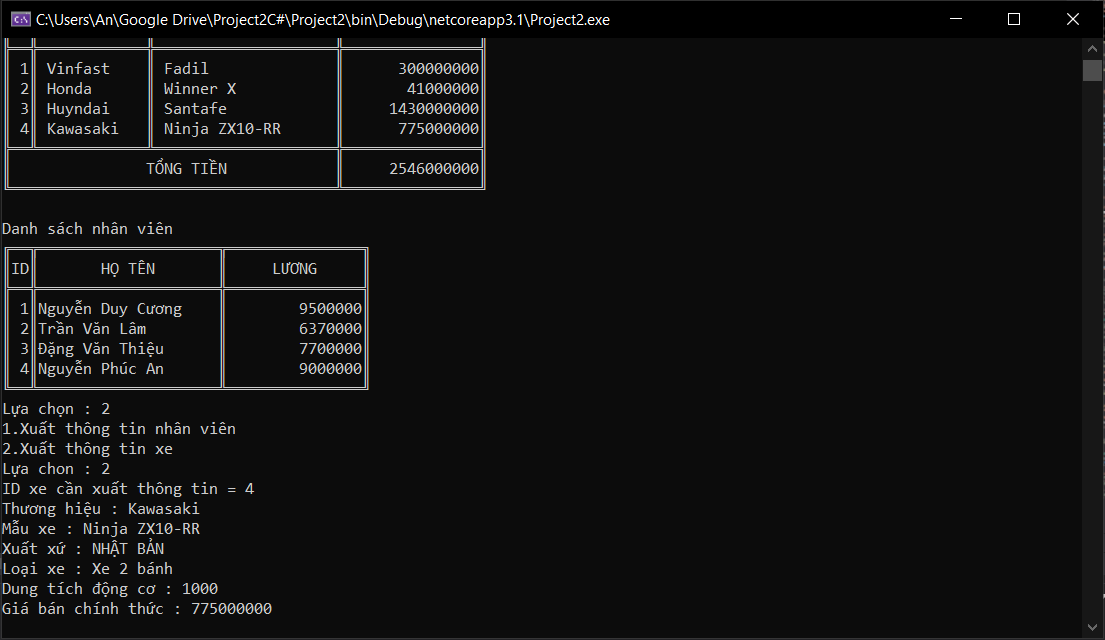
Python



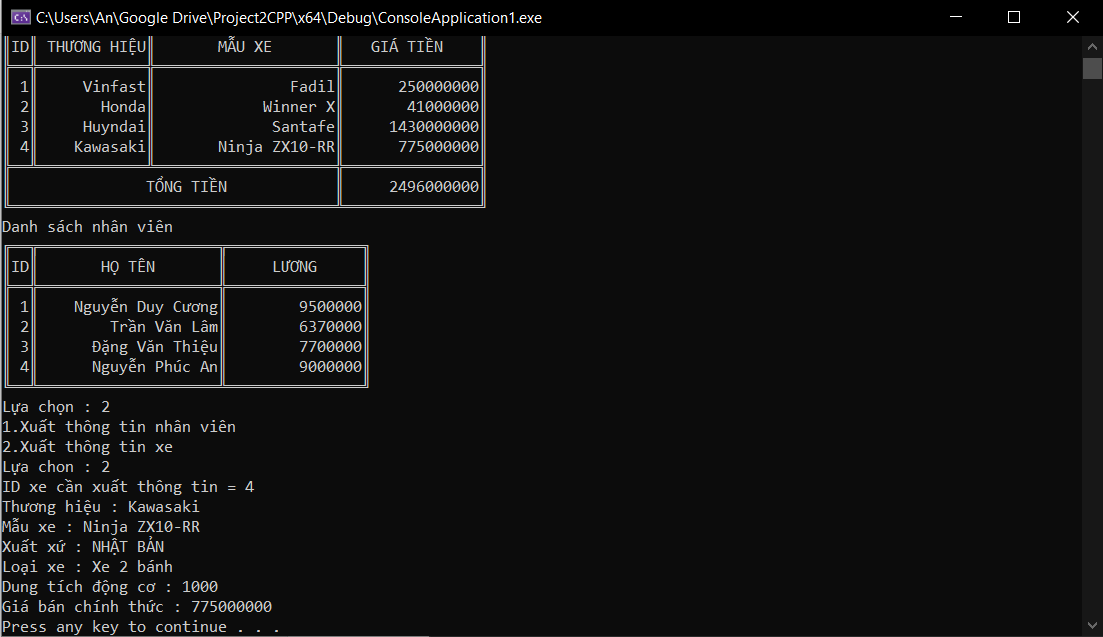
Hàm Xuat():

Cho người dùng xuất thông tin của một nhân viên hay xe tùy chọn:

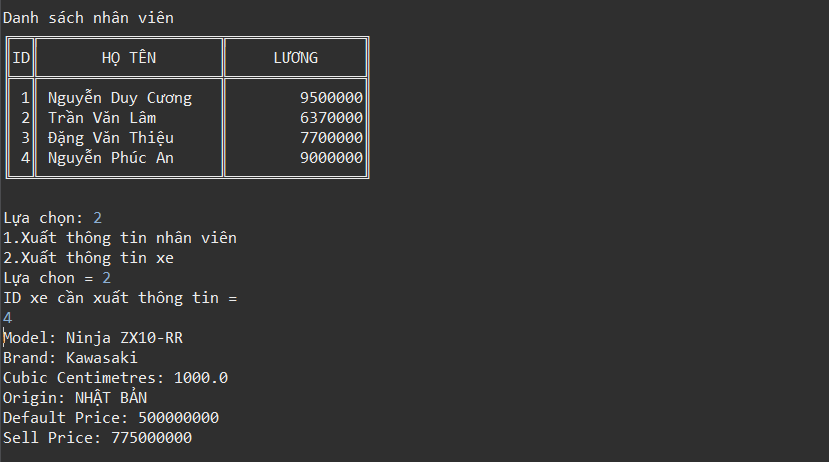
C#



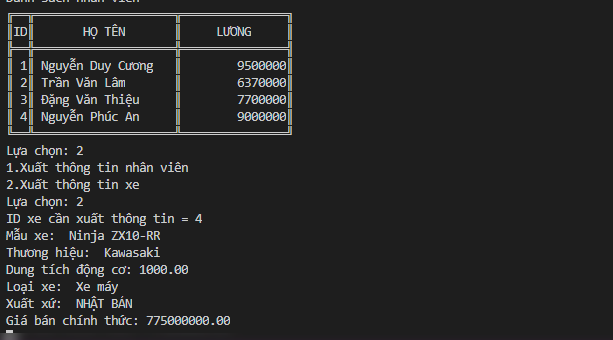
C++



Java



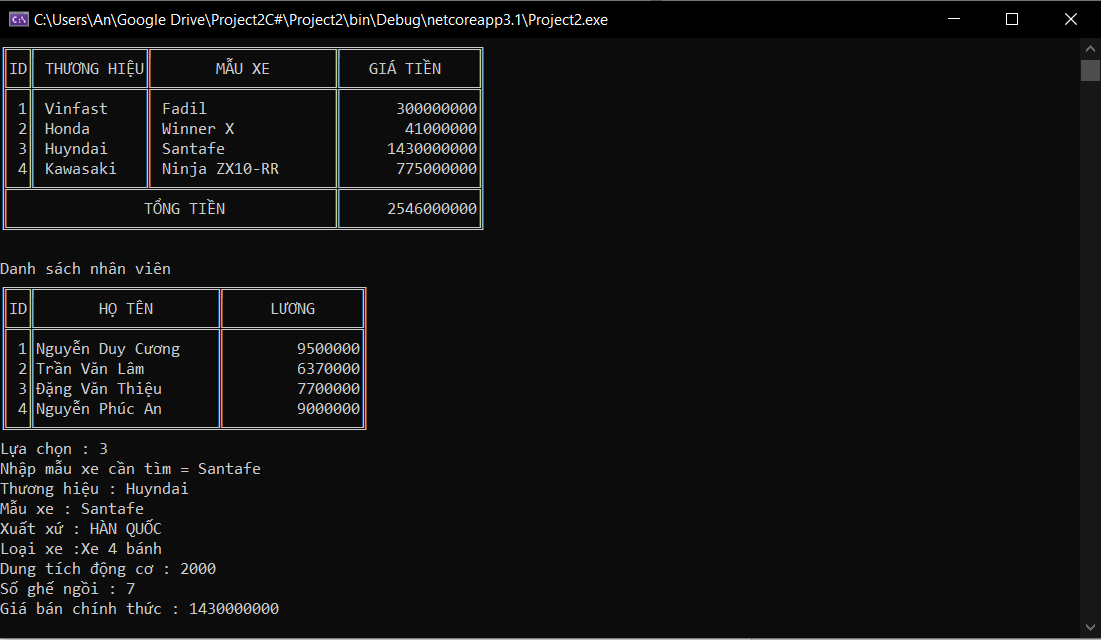
Python

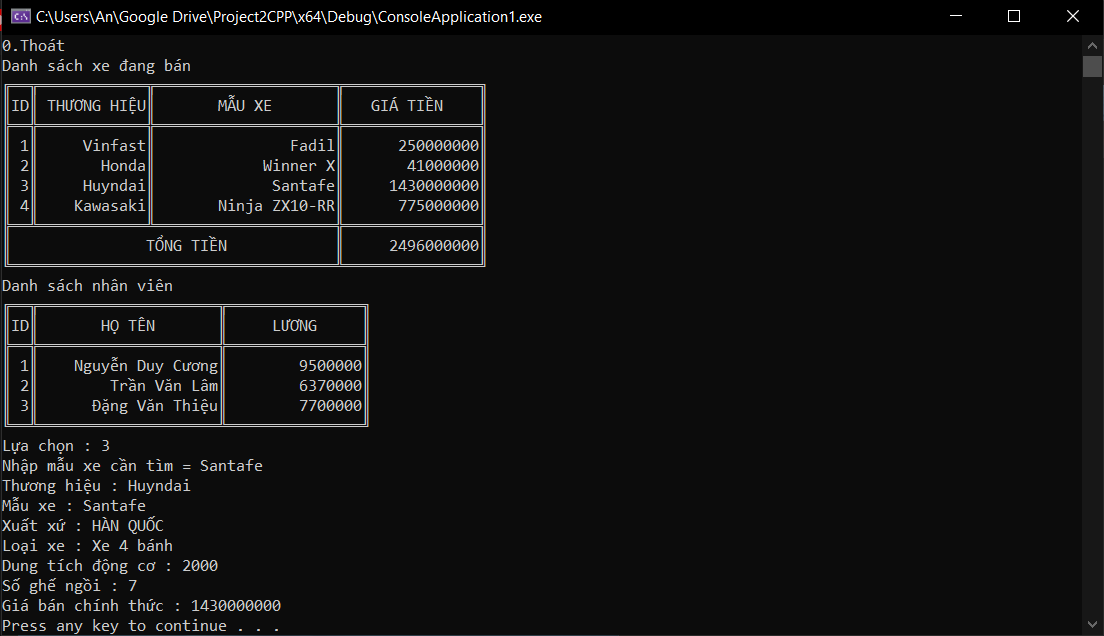


Hàm TimKiemXe():

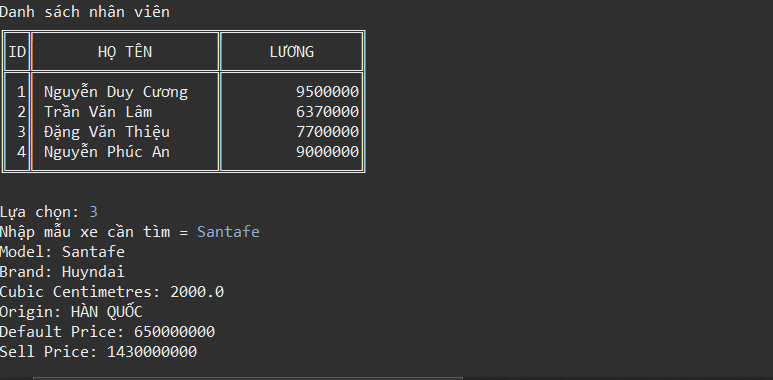
Cho người dùng tìm kiếm xe theo model người dùng nhập vào:

C#

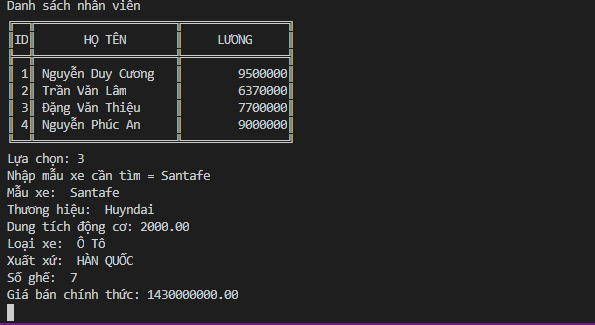
C++



Java



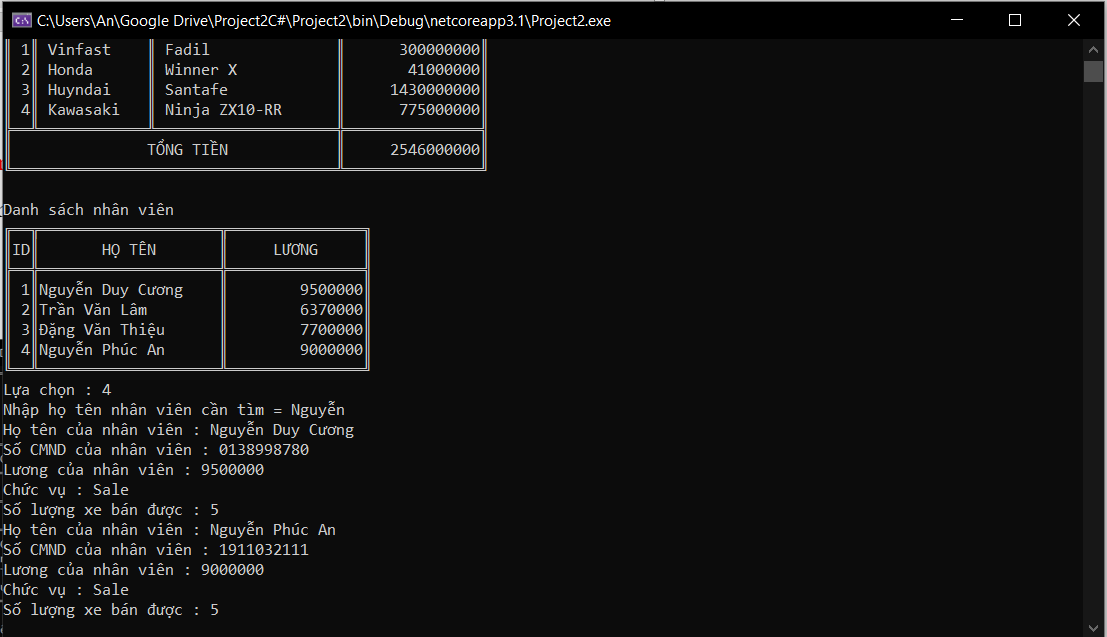
Python



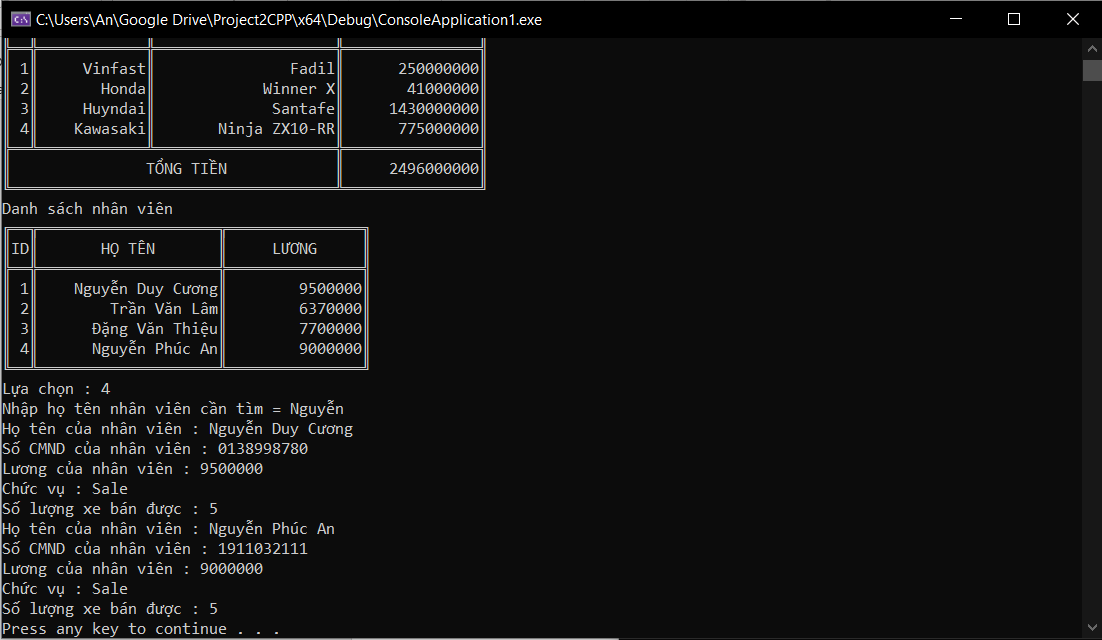
Hàm TimKiemNhanVien():

Cho người dùng tìm kiếm nhân viên với họ, tên hoặc họ và tên đầy đủ :

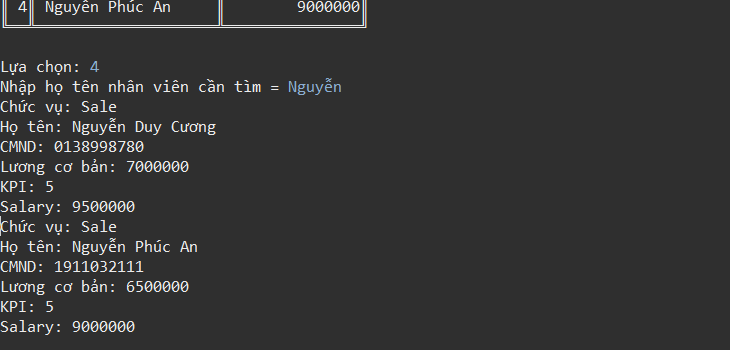
C#



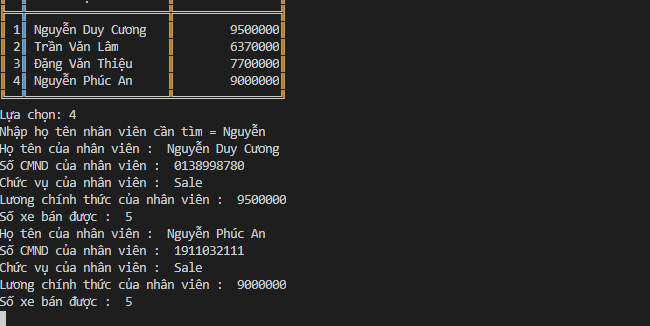
C++



Java



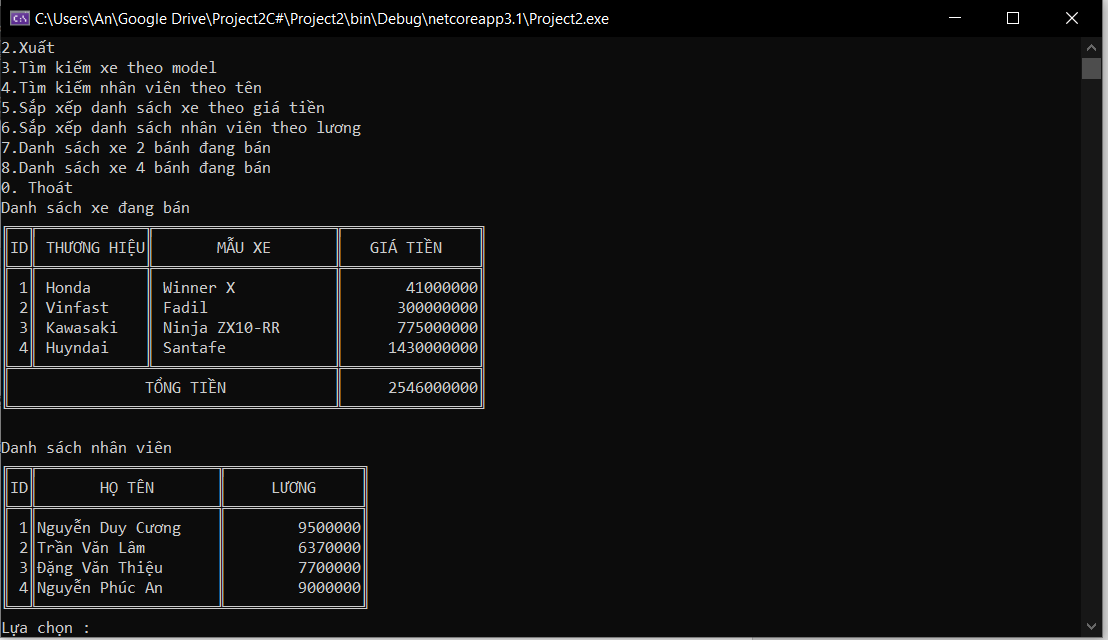
Python



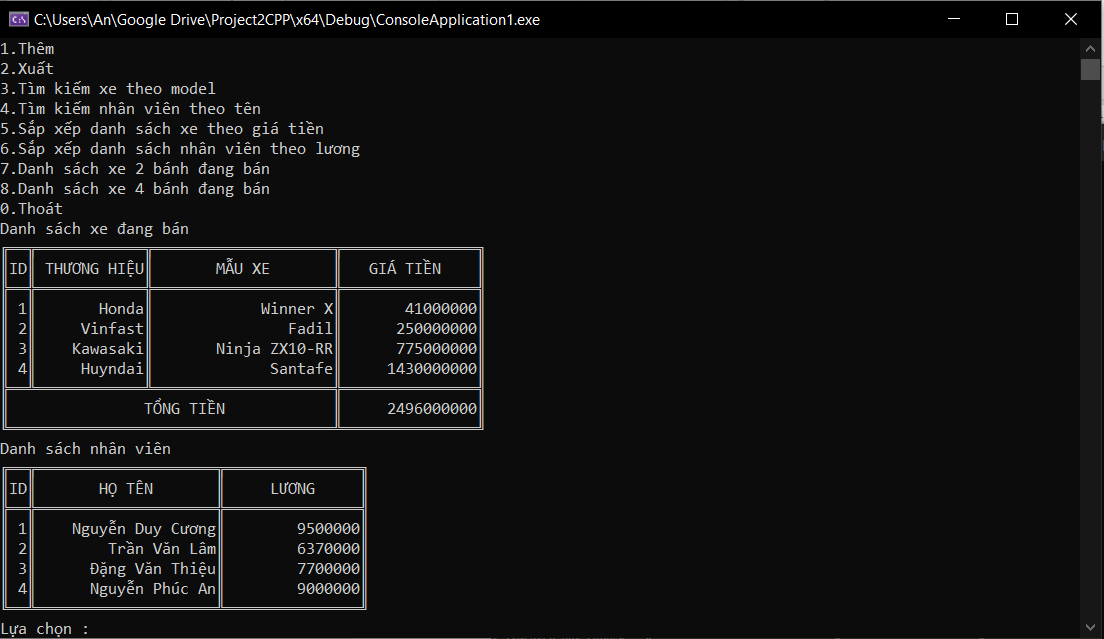
Hàm SapXepXe():

Sắp xếp xe đang bán theo thứ tự giá bán tăng dần:

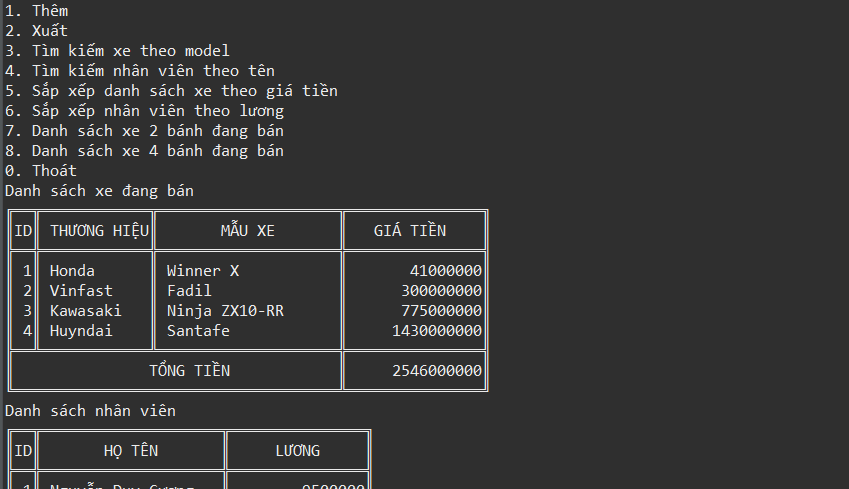
C#



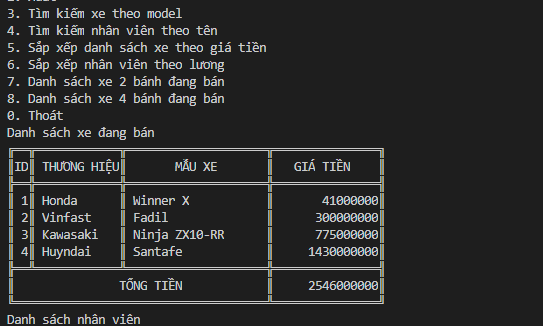
C++



Java



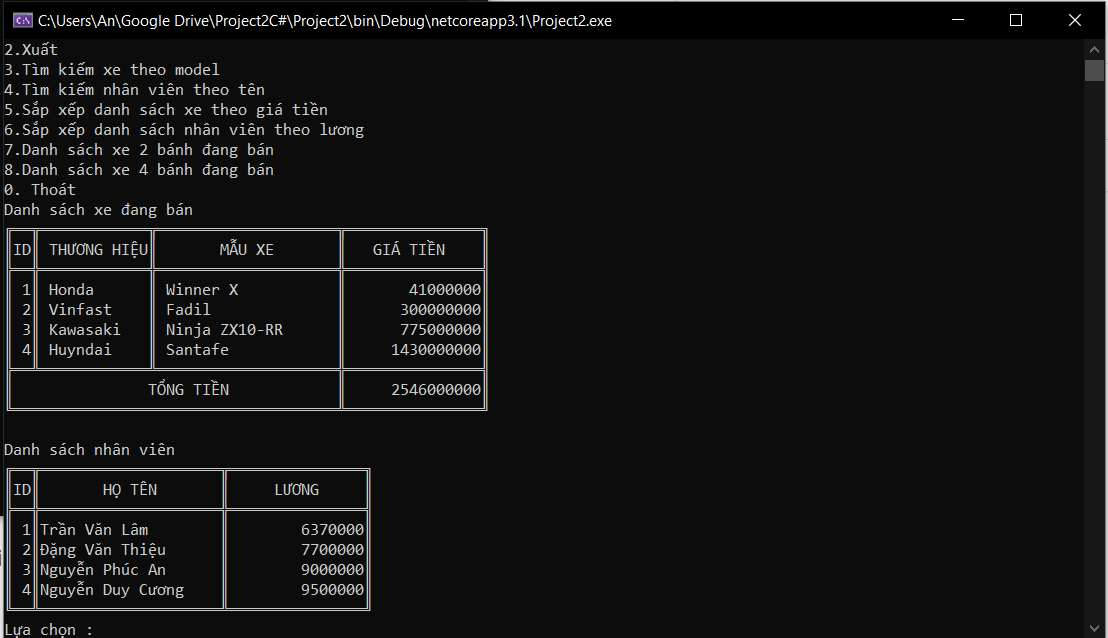
Python



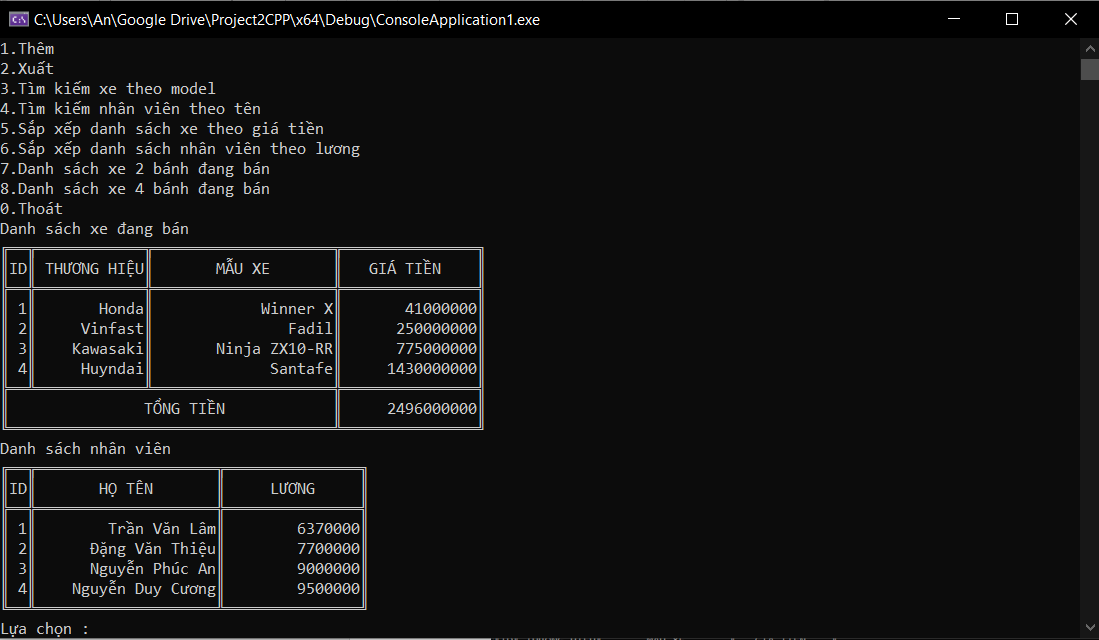
Hàm SapXepNhanVien():

Sắp xếp nhân viên theo lương tăng dần:

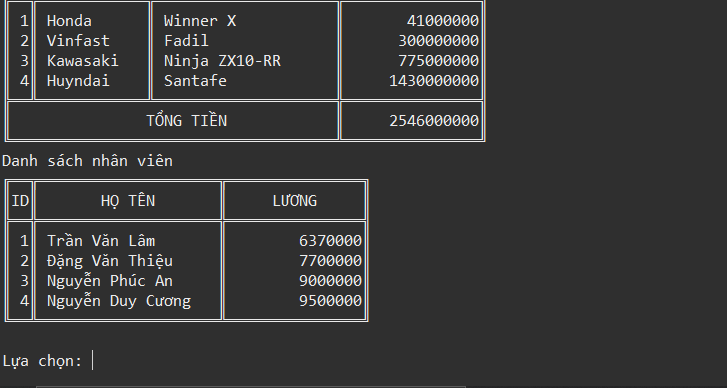
C#



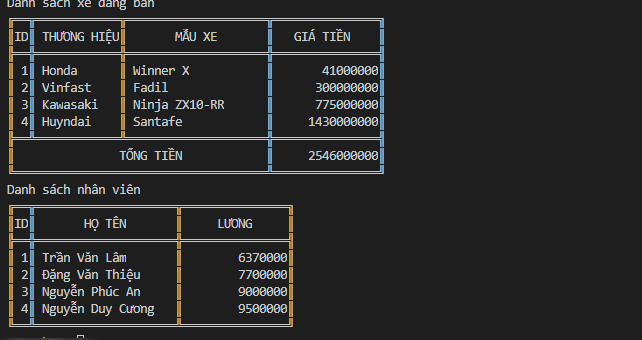
C++



Java



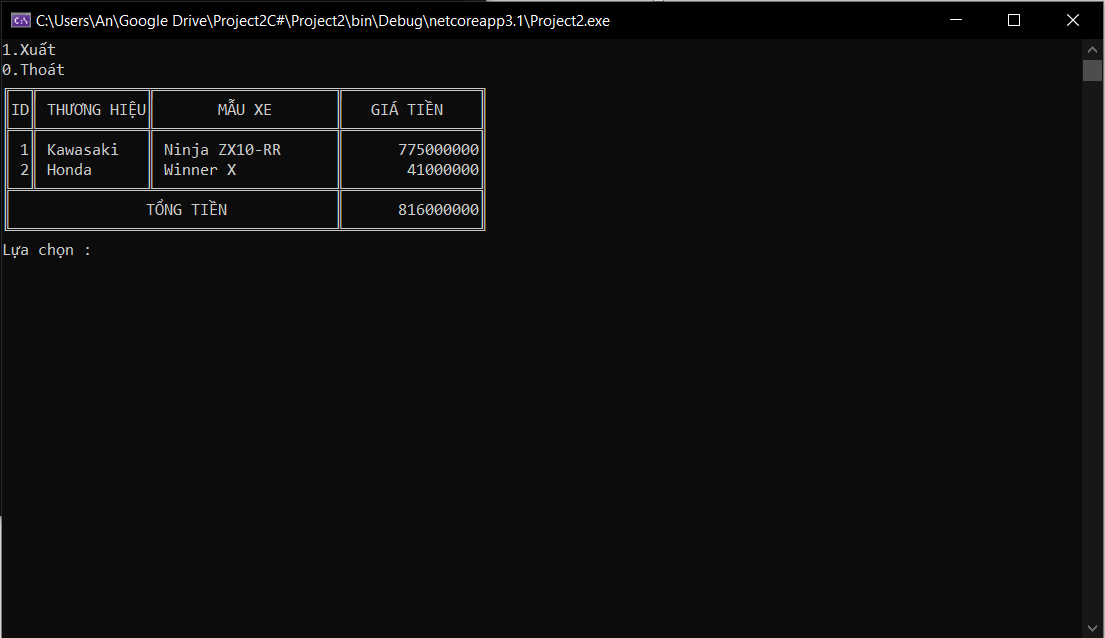
Python



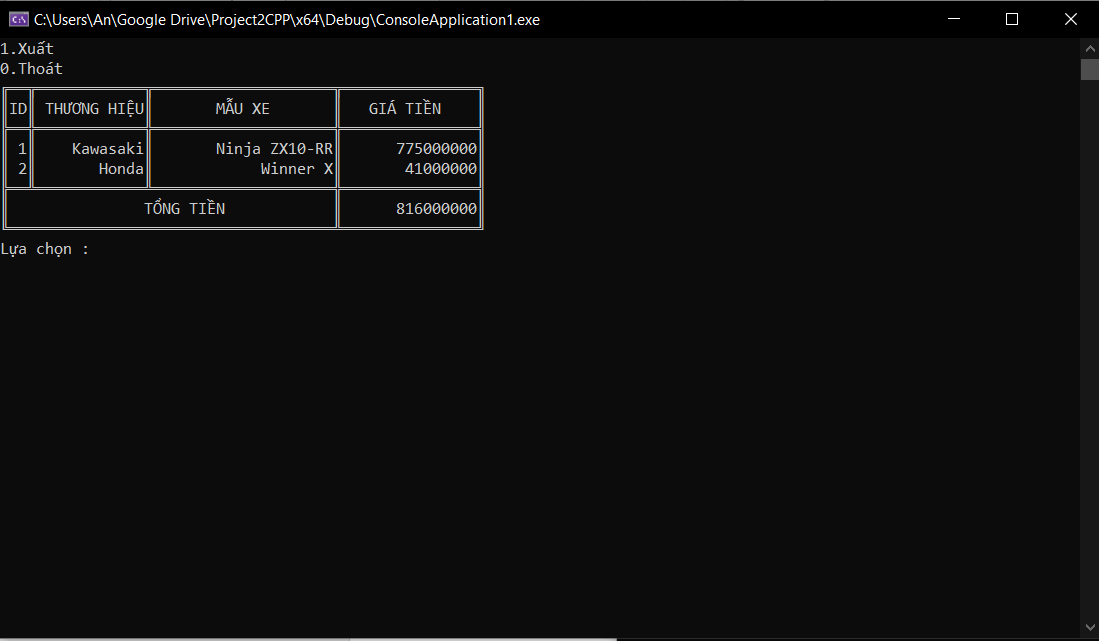
Hàm DanhSachXe2Banh():

Cho người dùng lọc ra danh sách xe 2 bánh đang bán ở cửa hàng và xuất thông tin xe tùy theo lựa chọn:

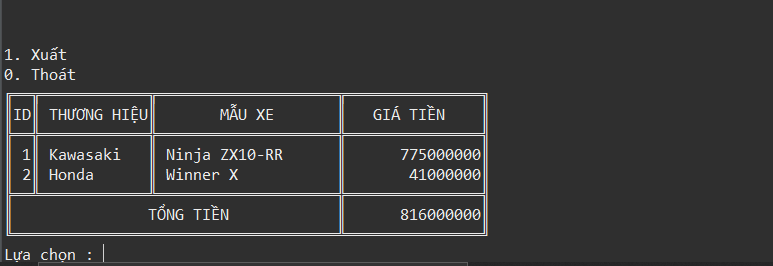
C#



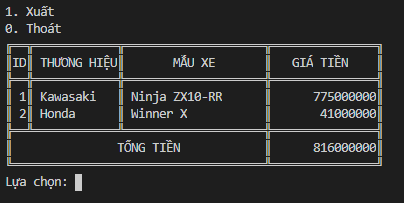
C++



Java



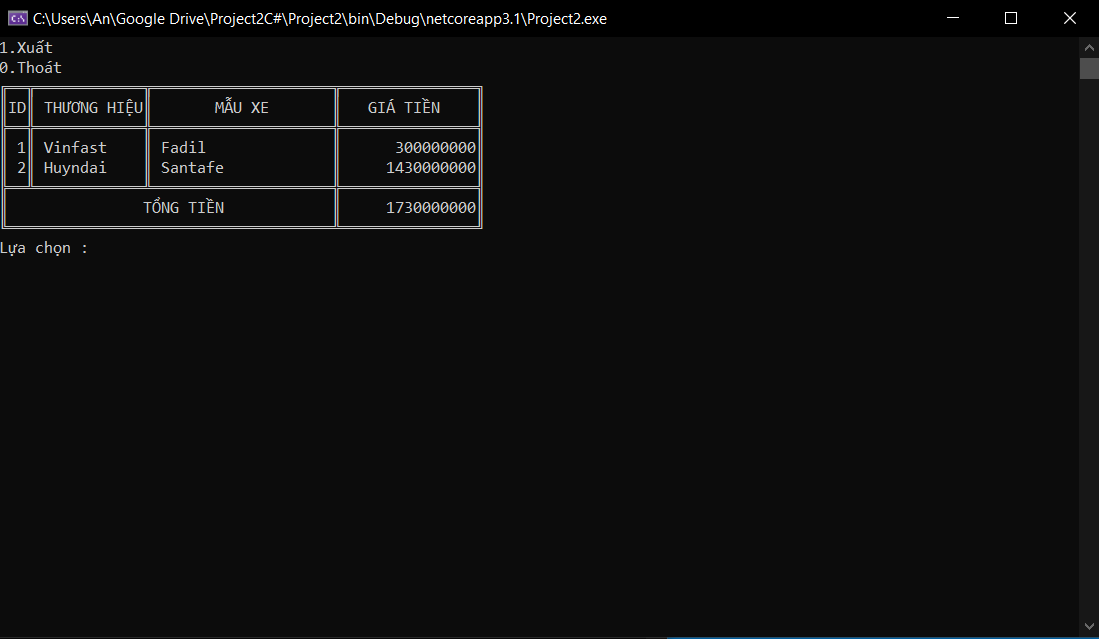
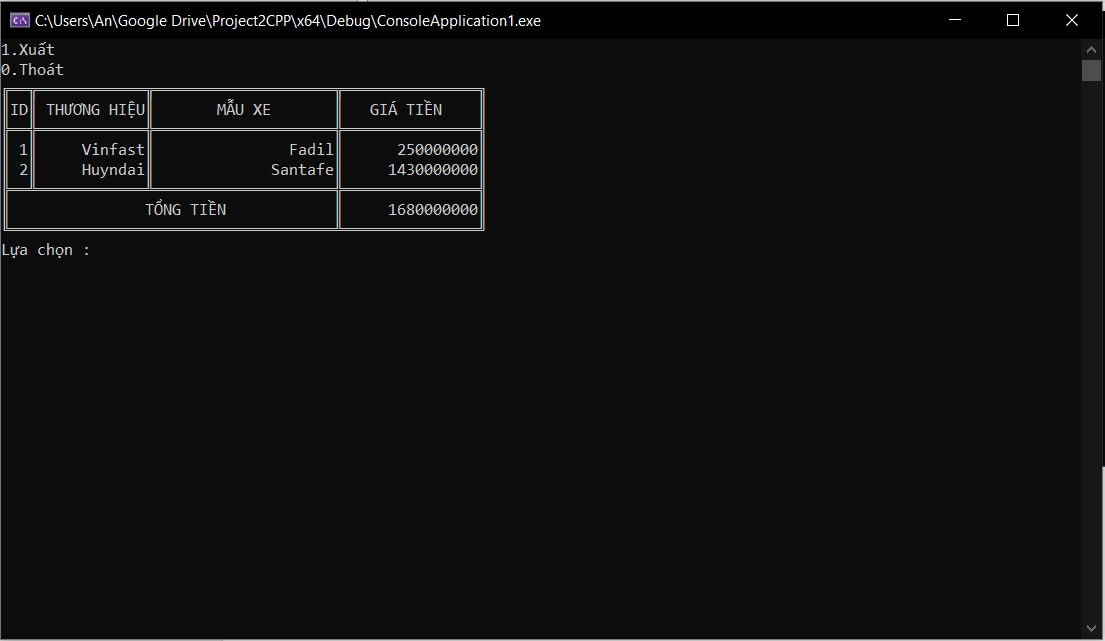
Python



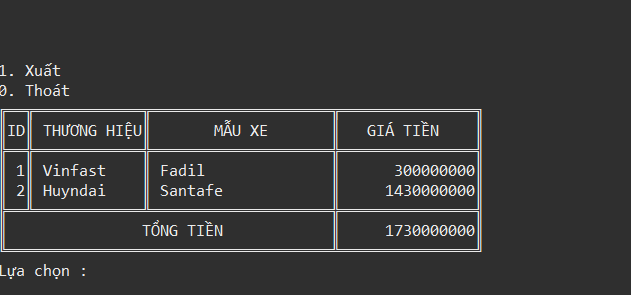
Hàm DanhSachXe4Banh():

Cho người dùng lọc ra danh sách xe 4 bánh đang bán ở cửa hàng và xuất thông tin xe tùy theo lựa chọn:

C#

C++

Java



Python

