**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM**

**THÀNH PHỐ HỒ CHÍ MINH**

**BÀI TẬP LỚN**

**XÂY DỰNG CHƯƠNG TRÌNH QUẢN LÝ NHÂN SỰ Ở CÔNG TY ELCA**

***GVHD:* ThS.Nguyễn Hồng Vũ**

**Danh sách sinh viên:**

Dương Thị Huỳnh Như – 2001190718 – 10DHTH2

Nguyễn Chí Cường – 2001180148 – 09DHTH5

Tăng Hoàn Nhựt – 2001181249 – 09DHTH6

Lê Thị Thuý Kiều – 2001180206 – 09DHTH5

Trần Hoà Thuận – 2001190848 – 10DHTH8

*TP. Hồ Chí Minh, tháng 09 năm 2021*

MỤC LỤC

[CHƯƠNG 1 GIỚI THIỆU ĐỀ TÀI 1](#_Toc82949009)

[1.1 Phạm vi và đối tượng nghiên cứu 1](#_Toc82949010)

[CHƯƠNG 2 CƠ SỞ LÝ THUYẾT 3](#_Toc82949011)

[2.1 Overload là gì? 3](#_Toc82949012)

[2.2 Override là gì? 3](#_Toc82949013)

[2.3 Trong C# 3](#_Toc82949014)

[2.4 Trong Java 6](#_Toc82949015)

[CHƯƠNG 3 THỰC HIỆN CHƯƠNG TRÌNH 11](#_Toc82949016)

[3.1 Sơ đồ phân cấp cấp trúc các lớp : 11](#_Toc82949017)

[3.1.1 Sơ đồ phân cấp: 11](#_Toc82949018)

[3.1.2 Sơ đồ các lớp: 11](#_Toc82949019)

[3.2 Chương trình cài đặt 16](#_Toc82949020)

[3.2.1 Lớp CongTyELCA 16](#_Toc82949021)

[3.2.2 Lớp NhanVien 21](#_Toc82949022)

[3.2.3 Lớp NhanVienELCA1 25](#_Toc82949023)

[3.2.4 Lớp NhanVienSanXuat 26](#_Toc82949024)

[3.2.5 Lớp NhanVienKinhDoanh 28](#_Toc82949025)

[3.2.6 Lớp CanBoQuanLy 31](#_Toc82949026)

[3.2.7 Lớp NhanVienELCA2 33](#_Toc82949027)

[3.2.8 Lớp NhanVienELCA3 35](#_Toc82949028)

[3.2.9 Lớp QuanLyPhongBan 37](#_Toc82949029)

[3.2.10 Lớp PhongBan 39](#_Toc82949030)

[3.2.11 Lớp NVELCA2\_XL 40](#_Toc82949031)

[3.2.12 Lớp NVELCA3\_XL 42](#_Toc82949032)

[3.2.13 Lớp NVKD\_XL 44](#_Toc82949033)

[3.2.14 Lớp NVSX\_XL 45](#_Toc82949034)

[3.2.15 Lớp Program 46](#_Toc82949035)

[CHƯƠNG 4 TỔNG KẾT VÀ ĐỊNH HƯỚNG PHÁT TRIỂN 51](#_Toc82949036)

[TÀI LIỆU THAM KHẢO 52](#_Toc82949037)

DANH MỤC BẢNG

[Bảng 2.3.1 So sánh Tính chất Override và Overload 4](#_Toc82948889)

[Bảng 2.4.1 Cú pháp Override và Overload trong Java 7](#_Toc82948890)

[Bảng 3.2.1 Sơ đồ lớp CongTyELCA 17](#_Toc82948891)

[Bảng 3.2.2 Sơ đồ lớp NhanVien 22](#_Toc82948892)

[Bảng 3.2.3 Sơ đồ lớp NhanVien 25](#_Toc82948893)

[Bảng 3.2.4 Sơ đồ lớp NhanVienSanXuat 26](#_Toc82948894)

[Bảng 3.2.5 Sơ đồ lớp NhanVienKinhDoanh 29](#_Toc82948895)

[Bảng 3.2.6 Sơ đồ lớp CanBoQuanLy 31](#_Toc82948896)

[Bảng 3.2.7 Sơ đồ lớp NhanVienELCA2 33](#_Toc82948897)

[Bảng 3.2.8 Sơ đồ lớp NhanVienELCA3 35](#_Toc82948898)

[Bảng 3.2.9 Sơ đồ lớp QuanLyPhongBan 37](#_Toc82948899)

[Bảng 3.2.10 Sơ đồ lớp PhongBan 39](#_Toc82948900)

[Bảng 3.2.11 Sơ đồ lớp NVELCA2\_XL 40](#_Toc82948901)

[Bảng 3.2.12 Sơ đồ lớp NVKD\_XL 44](#_Toc82948902)

[Bảng 3.2.13 Sơ đồ lớp NVSX\_XL 45](#_Toc82948903)

DANH MỤC HÌNH

[Hình 3.1.1.1 Ví dụ Overload 4](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948906)

[Hình 3.1.1.2 Ví dụ Override 5](#_Toc82948907)

[Hình 3.1.1.3 Ví dụ Override 5](#_Toc82948908)

[Hình 3.1.1.4 Ví dụ Override 5](#_Toc82948909)

[Hình 3.1.1.5 Ví dụ Override 6](#_Toc82948910)

[Hình 3.1.1.6 Kết quả Ví dụ Override 6](#_Toc82948911)

[Hình 3.1.1.1 Lớp Person 8](#_Toc82948912)

[Hình 3.1.1.2 Lớp Student 9](#_Toc82948913)

[Hình 3.1.1.3 Ví dụ Overload 9](#_Toc82948914)

[Hình 3.1.1.1 Sơ đồ phân cấp 11](#_Toc82948915)

[Hình 3.1.1.1 Class CongtyELCA 11](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948916)

[Hình 3.1.2.2 Class NhanVien 12](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948917)

[Hình 3.1.2.3 Class NhanVienELCA1 12](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948918)

[Hình 3.1.2.4 CLASS NhanVienELCA2 12](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948919)

[Hình 3.1.2.5 Class NhanVienELCA3 13](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948920)

[Hình 3.1.2.6 Class CanBoQuanLy 13](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948921)

[Hình 3.1.2.7 Class Nhanvienkinhdoanh 13](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948922)

[Hình 3.1.2.8 Class NhanVienSanXuat 13](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948923)

[Hình 3.1.2.9 Class NVELCA2\_XL 14](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948924)

[Hình 3.1.2.10 Class NVKD\_XL 14](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948925)

[Hình 3.1.2.11 Class NVELCA3\_XL 14](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948926)

[Hình 3.1.2.12 Class NVSX\_XL 15](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948927)

[Hình 3.1.2.13 Class PhongBan 15](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948928)

[Hình 3.1.2.14 Class QuanLyPhongBan 15](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948929)

[Hình 3.1.2.1 Cây thư mục chương trình 16](#_Toc82948930)

[Hình 3.2.1.1 Sơ đồ lớp CongTyELCA 18](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948931)

[Hình 3.2.1.2 Cài đặt phương thức ThemNhanVien (nv)\_2 18](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948932)

[Hình 3.2.1.3 Cài đặt phương thức TimNhanVienTheoCTy(CongTy) 19](#_Toc82948933)

[Hình 3.2.1.4 Cài đặt phương thức CapNhatThongTinNhanVien (NhanVien nv) 19](#_Toc82948934)

[Hình 3.2.1.5 Cài đặt phương thức XoaNhanVien(maNV) 20](#_Toc82948935)

[Hình 3.2.1.6 Cài đặt phương thức demsoNVmoiCTy() 20](#_Toc82948936)

[Hình 3.2.1.7 Cài đặt phương thức NhanVienCoThamNienTren10Nam() 20](#_Toc82948937)

[Hình 3.2.1.8 Cài đặt phương thức TongPCTNTheoNam(nam) 20](#_Toc82948938)

[Hình 3.2.1.9 Cài đặt phương thức TongLuong2021() 21](#_Toc82948939)

[Hình 3.2.1.10 Cài đặt phương thức NhanVienLaChienSiThiDua() 21](#_Toc82948940)

[Hình 3.2.1.11 Cài đặt phương thức xuatDSKhongDatChiTieu\_01\_2021() 21](#_Toc82948941)

[Hình 3.2.2.1 Cài đặt phương thức checkSdt(sdt) 24](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948942)

[Hình 3.2.2.2 Cài đặt phương thức checkGioiTinh(gt) 24](file:///C:\Users\Huynh%20Nhu\Downloads\BTL_LTHDT_.docx#_Toc82948943)

[Hình 3.2.2.3 Cài đặt theo phương thức checkTuoi(ns) 24](#_Toc82948944)

[Hình 3.2.2.4 Cài đặt phương thức Thamnien() 25](#_Toc82948945)

[Hình 3.2.2.5 Cài đặt phương thức PhuCapThamnien() 25](#_Toc82948946)

[Hình 3.2.2.6 Cài đặt phương thức xuat() 25](#_Toc82948947)

[Hình 3.2.3.1 Cài đặt phương thức NhanLuong(xl,Luong) 26](#_Toc82948948)

[Hình 3.2.3.2 Cài đặt phương thức xuat() 26](#_Toc82948949)

[Hình 3.2.4.1 Cài đặt phương thức Luong() 27](#_Toc82948950)

[Hình 3.2.4.2 Cài đặt phương thức ChuaDatChiTieu\_1\_2021() 27](#_Toc82948951)

[Hình 3.2.4.3 Cài đặt phương thức xuat() 28](#_Toc82948952)

[Hình 3.2.4.4 Cài đặt phương thức XepLoaiThiDua() 28](#_Toc82948953)

[Hình 3.2.4.5 Cài đặt phương thức TongLuong2021 28](#_Toc82948954)

[Hình 3.2.4.6 Cài đặt phương thức tongPCTheoNam(nam) 28](#_Toc82948955)

[Hình 3.2.5.1 Cài đặt phương thức ChuaDatChiTieu\_1\_2021() 29](#_Toc82948956)

[Hình 3.2.5.2 Cài đặt phương thức Luong() 30](#_Toc82948957)

[Hình 3.2.5.3 Cài đặt phương thức TongLuong2021() 30](#_Toc82948958)

[Hình 3.2.5.4 Cài đặt phương thức tongPCTheoNam(nam) 30](#_Toc82948959)

[Hình 3.2.5.5 Cài đặt phương thức XepLoaiThiDua() 30](#_Toc82948960)

[Hình 3.2.5.6 Cài đặt phương thức xuat() 30](#_Toc82948961)

[Hình 3.2.6.1 Cài đặt phương thức ChuaDatChiTieu\_1\_2021() 32](#_Toc82948962)

[Hình 3.2.6.2 Cài đặt phương thức Luong() 32](#_Toc82948963)

[Hình 3.2.6.3 Cài đặt phương thức TongLuong2021() 32](#_Toc82948964)

[Hình 3.2.6.4 Cài đặt phương thức tongPCTheoNam(nam) 32](#_Toc82948965)

[Hình 3.2.6.5 Cài đặt phương thức XepLoaiThiDua() 32](#_Toc82948966)

[Hình 3.2.6.6 Cài đặt phương thức xuat() 32](#_Toc82948967)

[Hình 3.2.7.1 Cài đặt phương thức ChuaDatChiTieu\_1\_2021() 33](#_Toc82948968)

[Hình 3.2.7.2 Cài đặt phương thức Luong() 34](#_Toc82948969)

[Hình 3.2.7.3 Cài đặt phương thức xuat() 34](#_Toc82948970)

[Hình 3.2.7.4 Cài đặt phương thức XepLoaiThiDua() 34](#_Toc82948971)

[Hình 3.2.7.5 Cài đặt phương thức TongLuong2021() 34](#_Toc82948972)

[Hình 3.2.7.6 Cài đặt phương thức tongPCTheoNam(nam) 34](#_Toc82948973)

[Hình 3.2.8.1 Cài đặt phương thức ChuaDatChiTieu\_1\_2021() 35](#_Toc82948974)

[Hình 3.2.8.2 Cài đặt phương thức Luong() 36](#_Toc82948975)

[Hình 3.2.8.3 Cài đặt phương thức xuat() 36](#_Toc82948976)

[Hình 3.2.8.4 Cài đặt phương thức XepLoaiThiDua() 36](#_Toc82948977)

[Hình 3.2.8.5 Cài đặt hương thức TongLuong2021() 36](#_Toc82948978)

[Hình 3.2.8.6 Cài đặt phương thức tongPCTheoNam(nam) 36](#_Toc82948979)

[Hình 3.2.9.1 Cài đặt phương thức Them() 37](#_Toc82948980)

[Hình 3.2.9.2 Cài đặt phương thức Xoa() 37](#_Toc82948981)

[Hình 3.2.9.3 Cài đặt phương thức Sua() 38](#_Toc82948982)

[Hình 3.2.9.4 Cài đặt phương thức xuatDSPhongBan() 38](#_Toc82948983)

[Hình 3.2.10.1 Cài đặt phương thức checkmaPB() 39](#_Toc82948984)

[Hình 3.2.10.2 Cài đặt phương thức xuat() 40](#_Toc82948985)

[Hình 3.2.11.1 Cài đặt phương thức xeploaithang() 41](#_Toc82948986)

[Hình 3.2.11.2 Cài đặt phương thức Luong() 41](#_Toc82948987)

[Hình 3.2.11.3 Cài đặt phương thức xuat() 42](#_Toc82948988)

[Hình 3.2.12.1 Sơ đồ lớp NVELCA3\_XL 42](#_Toc82948989)

[Hình 3.2.12.2 Cài đặt phương thức xeploaithang() 43](#_Toc82948990)

[Hình 3.2.12.3 Cài đặt phương thức Luong() 43](#_Toc82948991)

[Hình 3.2.12.4 Cài đặt phương thức xuat() 43](#_Toc82948992)

[Hình 3.2.13.1 Cài đặt phương thức XepLoaiThiDua() 45](#_Toc82948993)

[Hình 3.2.13.2 Cài đặt phương thức DTHoaHong() 45](#_Toc82948994)

[Hình 3.2.13.3 Cài đặt phương thức xuất 45](#_Toc82948995)

[Hình 3.2.14.1 Cài đặt phương thức XepLoaiThiDua() 46](#_Toc82948996)

[Hình 3.2.14.2 Cài đặt phương thức Xuat() 46](#_Toc82948997)

[Hình 3.2.15.1 Chạy yêu cầu\_1 47](#_Toc82948998)

[Hình 3.2.15.2 Chạy yêu cầu\_2 47](#_Toc82948999)

[Hình 3.2.15.3 Chạy yêu cầu\_3 47](#_Toc82949000)

[Hình 3.2.15.4 Chạy yêu cầu\_4 48](#_Toc82949001)

[Hình 3.2.15.5 Chạy yêu cầu\_5 48](#_Toc82949002)

[Hình 3.2.15.6 Chạy yêu cầu\_6 48](#_Toc82949003)

[Hình 3.2.15.7 Chạy yêu cầu\_7 48](#_Toc82949004)

[Hình 3.2.15.8 Chạy yêu cầu\_8 49](#_Toc82949005)

[Hình 3.2.15.9 Chạy yêu cầu\_9 49](#_Toc82949006)

[Hình 3.2.15.10 Chạy yêu cầu\_10 49](#_Toc82949007)

[Hình 3.2.15.11 Chạy yêu cầu\_11 50](#_Toc82949008)

LỜI CAM ĐOAN

Chúng em xin cam đoan đây là công trình nghiên cứu của riêng chúng em. Các số liệu, kết quả nêu trong đồ án là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Chúng em xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện đồ án này đã được cảm ơn và các thông tin trích dẫn trong đồ án đã được chỉ rõ nguồn gốc.

**Sinh viên thực hiện**

*(Đại diện ký tên)*

*Như*

*Dương Thị Huỳnh Như*

LỜI CẢM ƠN

Lời đầu tiên cho phép chúng em gửi lời cảm ơn tới toàn thể các thầy cô giáo trong Khoa Công Nghệ Thông Tin - Trường Đại học Công Nghiệp Thực Phẩm TP.HCM, những người đã hết mình truyền đạt và chỉ dẫn cho chúng em những kiến thức, những bài học quý báu và bổ ích trong suốt thời gian học vừa qua.

Để hoàn thành được bài tập lớn này, đặc biệt chúng em xin được bày tỏ sự tri ân và xin chân thành cảm ơn giảng viên thầy Nguyễn Hồng Vũ người trực tiếp hướng dẫn, chỉ bảo chúng em trong suốt quá trình học tập và thực hành để hoàn thành bài tập lớn này.

Trong quá trình thực hành và làm báo cáo do năng lực, kiến thức, trình độ bản thân còn hạn hẹp nên không tránh khỏi những thiếu sót. Chúng em mong nhận được sự thông cảm và những ý kiến đóng góp của quý thầy cô và các bạn

Chúng em xin chân thành cảm ơn !

# GIỚI THIỆU ĐỀ TÀI

## Phạm vi và đối tượng nghiên cứu

* Phạm vi nghiên cứu: Tính chất của phương thức overload và override trong lập trình hướng đối tượng. So sánh cú pháp của chúng giữa ngôn ngữ C# và Java. Cho ví dụ minh hoạ. Xây dựng chương trình quản lý nhân sự ở công ty ELCA.
* Đối tượng nghiên cứu: Phương thức override và overload. Ứng dụng kế thừa và đa hình trong việc xây dựng chương trình quản lý nhân viên.

**2.2 Mục tiêu nghiên cứu**

* Nghiên cứu , tìm hiểu về override, overload trong C# và Java.
* Giới thiệu bài toán ứng dụng xây dựng chương trình quản lý nhân viên.
* Trình bày được các kiến thức cơ bản về lý thuyết liên quan đến bài toán.
* Ứng dụng thực tế và Demo áp dụng các kiến thức đã học trong cách giải quyết vấn đề.
* Đánh giá tính hiệu quả và kết quả đạt được của đề tài

**3.3 Sự cần thiết và lý do chọn đề tài.**

* Trong những năm trở lại đây, ngoài việc code thì việc ứng dụng các phương thức cơ bản giúp kế thừa và rút ngắn source code được nhiều bạn coder áp dụng.
* Nó không chỉ giúp vấn đề được giải quyết nhanh gọn về thời gian mà còn có tính ứng dụng khá cao trong việc thực hiện các chương trình ứng dụng.
* Vì lý do trên, chúng em xin chọn đề tài “So sánh 2 tính chất của phương thức overload và override trong lập trình hướng đối tượng. So sánh cú pháp của chúng giữa ngôn ngữ C# và Java. Cho ví dụ minh họa. Xây dựng chương trình quản lý Nhân sự ở công ty ELCA.” làm đề tài nghiên cứu cho bài tập lớn của mình .

# CƠ SỞ LÝ THUYẾT

## Overload là gì?

* Overload(nạp chồng) là việc tạo ra nhiều phương thức (method) có cùng tên, trong cùng một phạm vi (scope), nhưng khác nhau về đối số đầu vào hoặc kiểu trả về.
* Overloading là nhiều phương thức trong một lớp có cùng tên nhưng :
* Khác nhau về tham số và có cùng kiểu dữ liệu của tham số.
* Có cùng tham số và khác nhau kiểu dữ liệu của tham số.
* Hàm dựng cũng có khả năng nạp chồng với quy tắc trên.

## Override là gì?

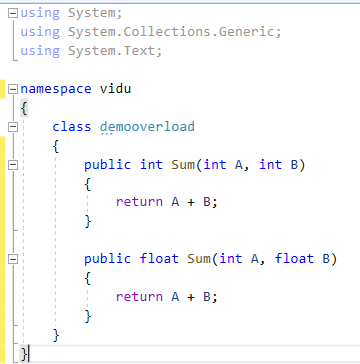
* Overriding ( ghi đè) có nghĩa là có 2 phương thức giống nhau về tên và tham số truyền vào. Một phương thức ở lớp cha, còn cái còn lại thuộc lớp con, Overriding cho phép lớp con có thể thực hiện riêng biệt cho phương thức mà lớp cha đã cung cấp.
* Phương thức ghi đè ở lớp con phải:
* Cùng tên, kiểu trả về và tham số với hàm được định nghĩa ở lớp cha.
* Không được sử dụng chỉ định từ truy xuất yếu hơn.

## Trong C#

* *So sánh 2 tính chất của phương thức Overload và Override trong Lập trình hướng đối tượng.*

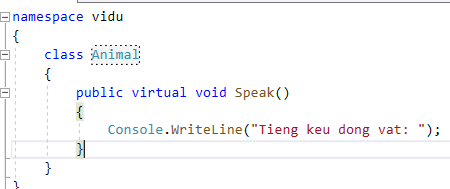
|  |  |
| --- | --- |
| ***Override*** | ***Overload*** |
| * Kiểu dữ liệu trả về , tên phương thức,danh sách tham sô của phương thức override và phương thức được override phải giống nhau * Không thể thu hẹp phạm vi truy cập(acces modifier) của phương thức được override * Không thể overriding constructor method * Chỉ thực hiện được đối với các class có quan hệ kế thừa. Do đó overriding thực hiện ở người phạm vu của một class. * Là hình thức đa hình khi chạy(runtime) (tức là chỉ khi chương trình chạy thì ,chúng ta mới biết phương thức được gọi từ lớp nào) * Không cho phép tạo ra những ngoại lệ khác loại hoặc không phải đối tượng thuộc lớp con của lớp có thể hiện là ngoại lệ từ phương thức được override | * Kiểu dữ liệu trả về của các phương thức Overload có thể giống nhau có thể giống nhau hoặc khác * Số lượng tham số hoặc kiểu dữ liệu của tham số ở các phương thức overload phải khác nhau * Có thể mở rộng hoặc truy cập(acces modifier) * Overloading được constructor method * Chỉ thực hiện trong cùng phạm vi trong nội bộ của 1 class. * Là hình thức đa hình khi biên dịch(compiler) (tức là khi biên dịch mới biết đang sử dụng phương thức ở lớp nào) * Cho phép tạo ra những ngoại lệ hoàn toàn mới so với những ngoại lệ từ phương thức được phương thức overload |

Bảng . So sánh Tính chất Override và Overload

* *Ví dụ:*
* *Overload*

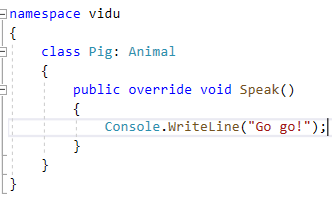
Hình . Ví dụ Overload

* *Override:*
* Ví dụ có một lớp động vật Animal, có phương thức là Speak() như sau:



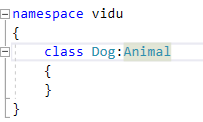
Hình . Ví dụ Override

* Cho một con động vậy cụ thể chẳng hạn như là Pig kế thừa lớp động vật Animal, với tiếng kêu cụ thể là Go go thì lúc này override lại method Speak() của lớp cha:



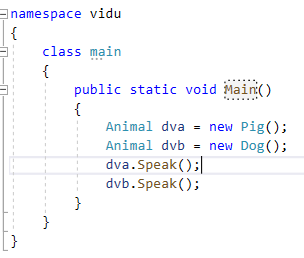
Hình . Ví dụ Override

* Và một con vật khác là Dog cũng kế thừa lớp động vật Animal, nhưng không Override lại method Speak() của lớp cha:



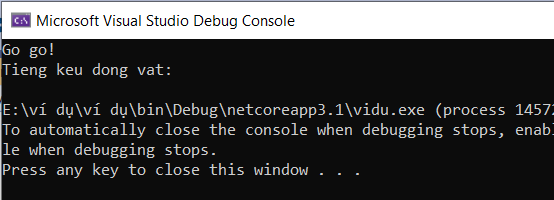
Hình . Ví dụ Override

* Lúc này nếu khởi tạo 2 đối tượng Pig và Dog thực hiện in ra tiếng kêu của mỗi con:



Hình . Ví dụ Override

* Thì chúng ta sẽ được kết quả là:



Hình . Kết quả Ví dụ Override

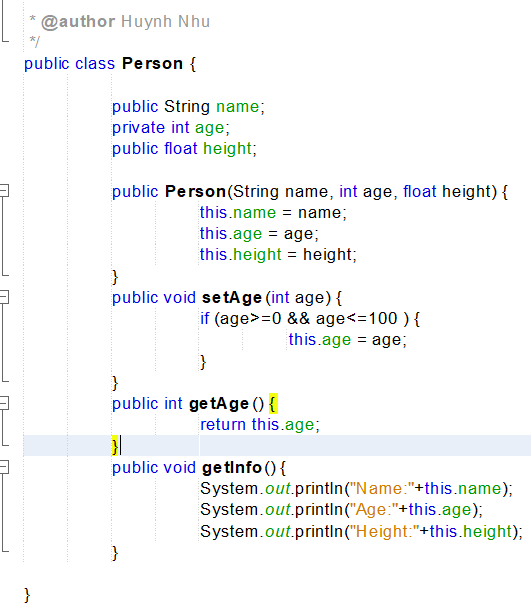
## Trong Java

* Override
* Từ khoá supper:
* Truy xuất đến phương thức cùng tên của lớp cha.
* Truy xuất đến hàm dựng của lớp cha.
* Lưu ý:
* Trong Java, chúng ta không thể thực hiện nạp chồng phương thức chỉ bằng cách thay đổi kiểu trả về của phương thức đó.
* Hàm tạo cũng có thể được nạp chồng.
* *So sánh cú pháp của Overload và Override trong JAVA. Cho ví dụ minh hoạ.*
* Sự khác nhau của overloading và override:

|  |  |  |
| --- | --- | --- |
|  | **Overloading (Nạp chồng phương thức)** | **(Overriding) Ghi đè phương thức** |
| Hành vi | Thêm hoặc mở rộng hành vi của phương thức. | Thay đổi hành vi hiện tại của phương thức. |
| Quyền truy cập | Các phương thức nạp chồng có thể có quyền truy cập khác nhau. | Phương thức ghi đè ở lớp con phải có quyền truy cập bằng hoặc lớn hơn phương thức được ghi đè ở lớp cha. |
| Phạm vi | Nạp chồng phương thức được thực hiện bên trong một class. | Ghi đè phương thức xảy ra bên trong 2 class có quan hệ kế thừa. |
| Danh sách tham số | Nạp chồng phương thức thì tham số phải khác nhau. | Ghi đè phương thức thì tham số phải giống nhau. |
| Đa hình | Nạp chồng phương thức là ví dụ về đa hình lúc biên dịch(Compile time). | Ghi đè phương thức là ví dụ về đa hình lúc runtime. |
| Giá trị trả về | Trong Java, nạp chồng phương thức không thể được thực hiện khi chỉ thay đổi kiểu giá trị trả về của phương thức. Kiểu giá trị trả về có thể giống hoặc khác nhưng tham số phải khác nhau. | Giá trị trả về phải giống nhau. |

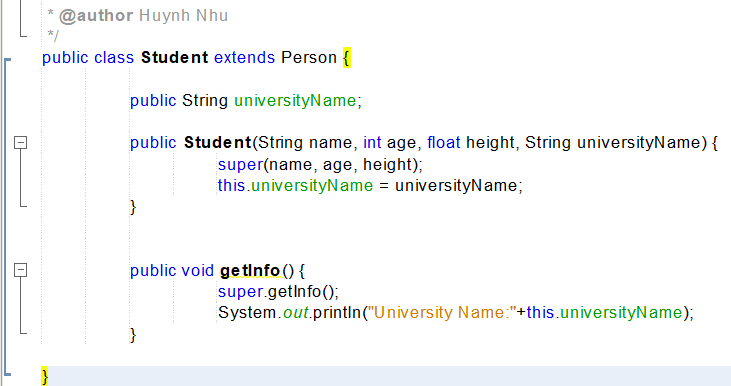
Bảng . Cú pháp Override và Overload trong Java

* Cho ví dụ minh hoạ.
* Override:
* Lớp Student kế thừa lớp Person. Với phương thức getInfo của lớp Person chỉ in được thông tin name, age, heught trong khi lớp Student còn thuộc tính universityName. Như vậy, ta sẽ overriding lại phương thức getInfo:
* Class Person:



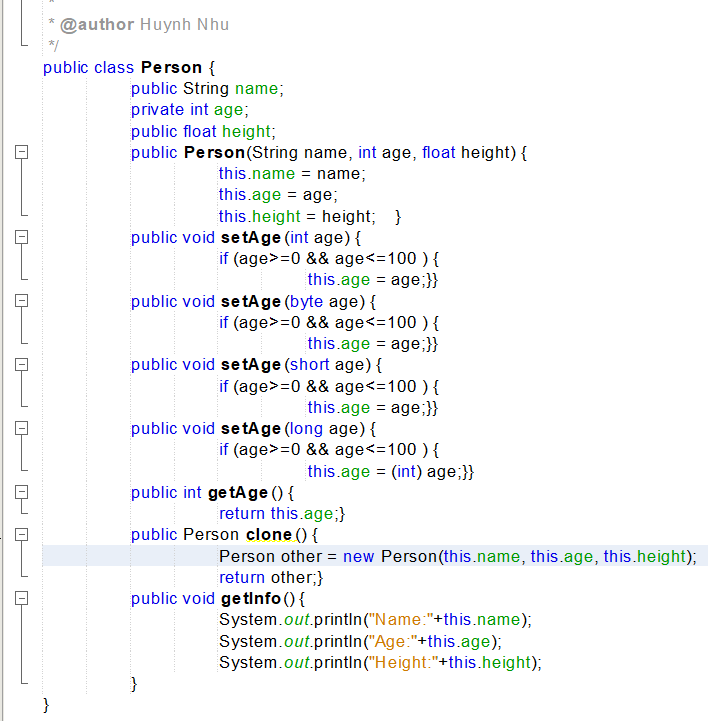
Hình . Lớp Person

* Khi ta khai báo phương thức getInfo trong lớp Student, có nghĩa ta đang overrding. Và đối tượng thuộc lớp Student sẽ gọi phương thức getInfo từ lớp Student thay vì lớp Person.
* Class Student:



Hình . Lớp Student

* Overload:
* Với setter cho thuộc tính age, có thể người dùng truyền vào tham số age kiểu int, kiểu byte, short hoặc long. Như vậy, ta sẽ Overloading nhiều phương thức setter cho thuộc tính age để đảm bảo.



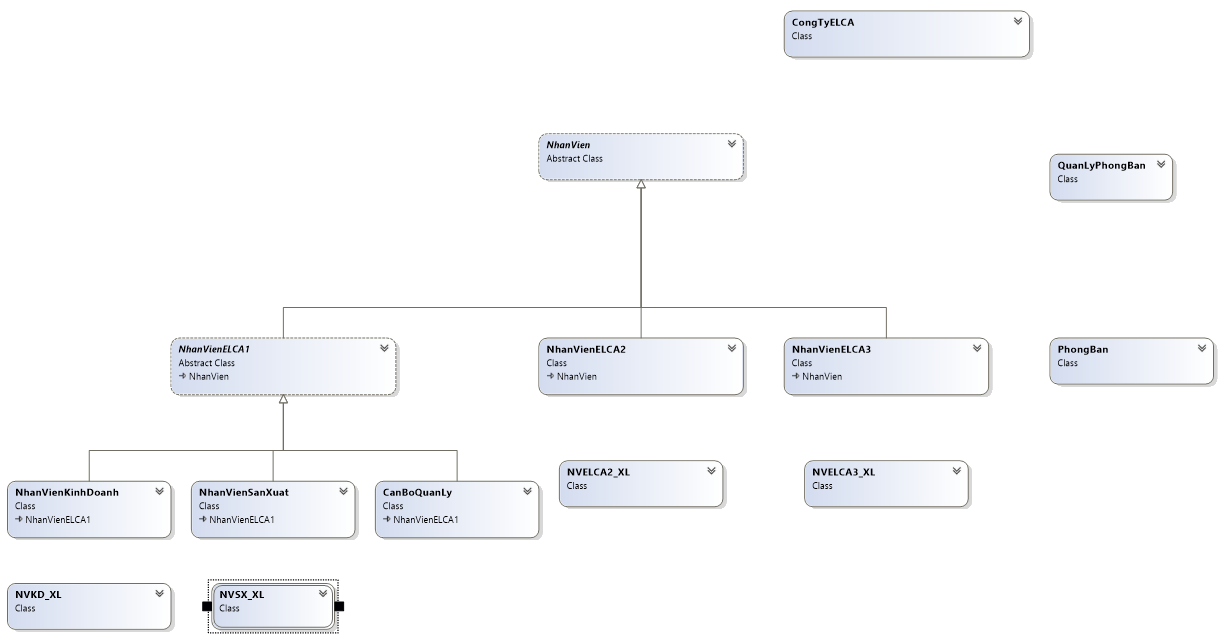
Hình . Ví dụ Overload

* *Khi nào sử dụng Overriding và overloading?*
* Overriding được sử dụng khi trong cùng một phương thức, chúng ta lại muốn thay đổi phần thân của phương thức đó.
* Overloading được dử dụng khi trong cùng một phương thứ, chúng ta lại muốn làm thêm một công việc khác.

# THỰC HIỆN CHƯƠNG TRÌNH

## Sơ đồ phân cấp cấp trúc các lớp :

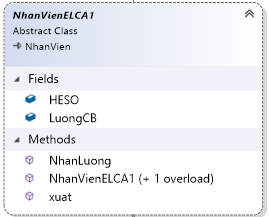
### Sơ đồ phân cấp:

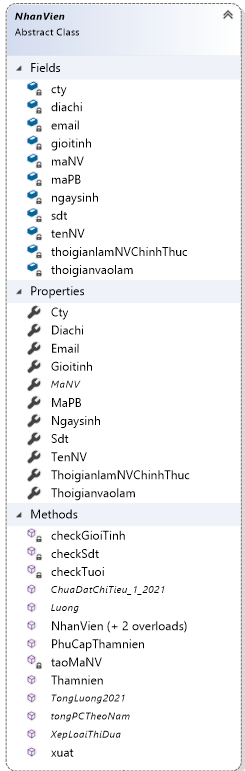


Hình . Sơ đồ phân cấp

### Sơ đồ các lớp:

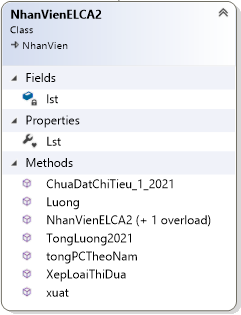
Hình . Class CongtyELCA



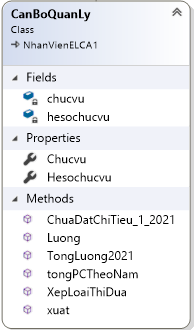
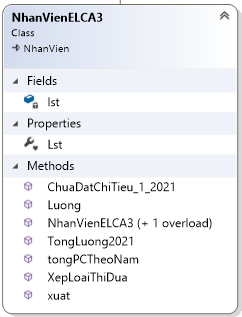


Hình . Class NhanVien

Hình . Class NhanVienELCA1

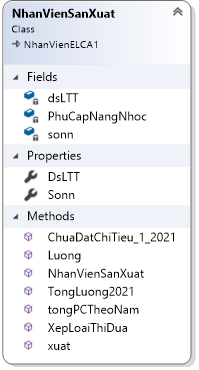


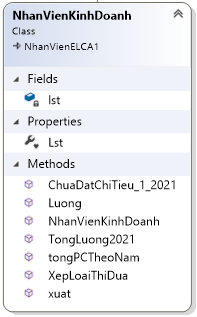
Hình . CLASS NhanVienELCA2



Hình . Class NhanVienELCA3

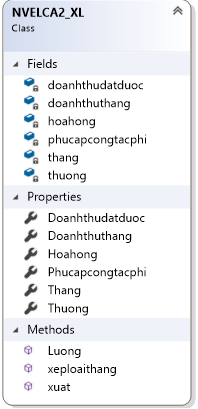
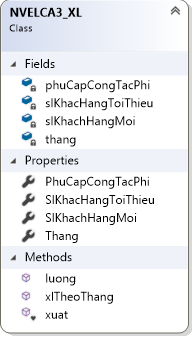
Hình . Class CanBoQuanLy

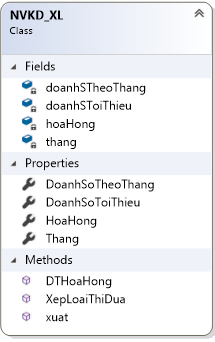




Hình . Class Nhanvienkinhdoanh

Hình . Class NhanVienSanXuat

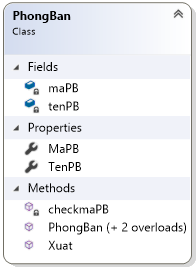
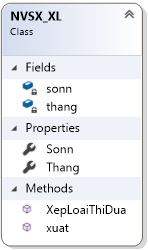
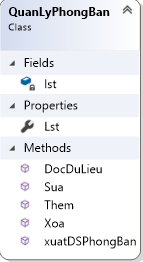




Hình . Class NVELCA2\_XL

Hình . Class NVKD\_XL

Hình . Class NVELCA3\_XL

**

Hình . Class NVSX\_XL

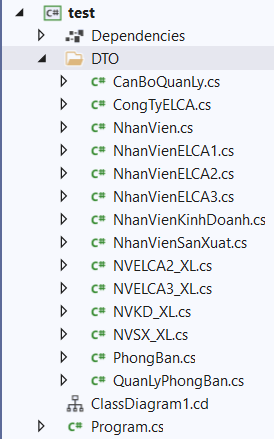
Hình . Class PhongBan

Hình . Class QuanLyPhongBan

## Chương trình cài đặt

Môi trường cài đặt: Visual studio 2019 , ngôn ngữ C#.

Chương trình có cấu trúc và gồm các lớp (class) như sau:



Hình . Cây thư mục chương trình

### Lớp CongTyELCA

* Đây là lớp cha trong chương trình xây dựng:

|  |
| --- |
| **CongTyELCA** |
| * root : XmlDocument * CongTyELCA() * docFile * CapNhatThongTinNhanVien (NhanVien nv) * XoaNhanVien(maNV):String * ThemNhanVien(nv) * Xuat(List<>) * TimNhanVienTheoCTy(CongTy):String * demsoNVmoiCTy() * NhanVienCoThamNienTren10Nam() * NhanVienLaChienSiThiDua() * TngPCTNTheoNam(nam):int * TongLuong2021() * xuatDSKhongDatChiTieu\_01\_2021() |

Bảng . Sơ đồ lớp CongTyELCA

#### Phương thức

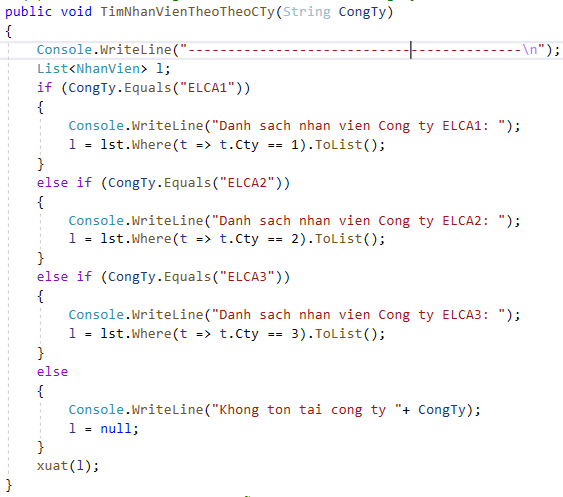
* CongTyELCA() : overload :Đọc file từ Xml.
* docFile : Đọc file từ Xml.
* CapNhatThongTinNhanVien (NhanVien nv) : Chỉnh sửa thông tin nhân viên được đưa vào.
* XoaNhanVien(maNV):String : Xoá thông tin nhân viên.
* ThemNhanVien(nv): Thêm mới 1 nhân viên của các công ty.
* Xuat(List<>): Xuất thông tin.
* TimNhanVienTheoCTy(CongTy):String: Tìm kiếm thông tin nhân viên của mỗi công ty.
* demsoNVmoiCTy(): Đếm số nhân viên của mỗi công ty.
* NhanVienCoThamNienTren10Nam(): Xuất danh sách nhân viên các công ty thành viên có thâm niên trên 10 năm.
* NhanVienLaChienSiThiDua(): Xuất thông tin các nhân viên là chiến sĩ thi đua của tổng công ty ELCA.
* TongPCTNTheoNam(nam):int: Xuất tổng phụ cấp thâm niên trong năm 2021 của ELCA 1, 2, 3.
* TongLuong2021() : Xuất tổng lương năm 2021 của ELCA 1, 2, 3.
* xuatDSKhongDatChiTieu\_01\_2021(): Xuất danh sách các nhân viên không đạt chỉ tiêu trong tháng 1/2021 của công ty ELCA.

#### Cài đặt phương thức

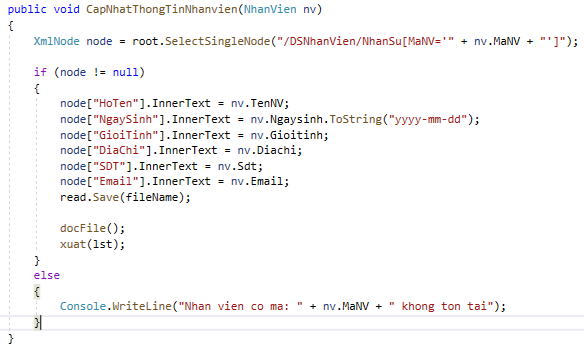
Hình . Sơ đồ lớp CongTyELCA



Hình . Cài đặt phương thức ThemNhanVien (nv)\_2



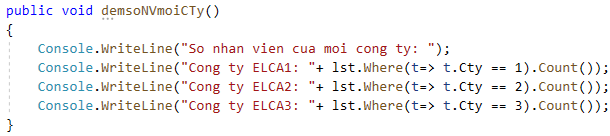
Hình . Cài đặt phương thức TimNhanVienTheoCTy(CongTy)



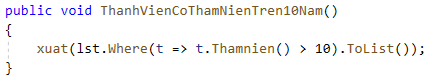
Hình . Cài đặt phương thức CapNhatThongTinNhanVien (NhanVien nv)



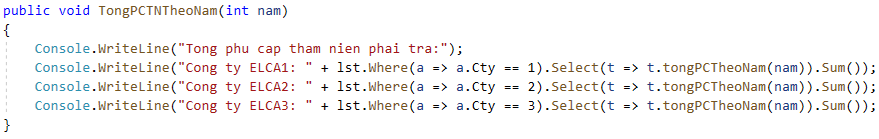
Hình . Cài đặt phương thức XoaNhanVien(maNV)



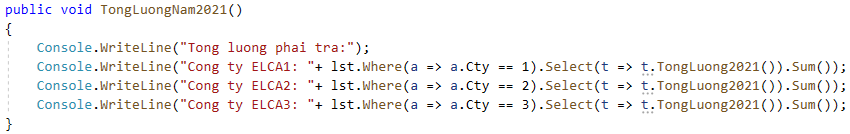
Hình . Cài đặt phương thức demsoNVmoiCTy()



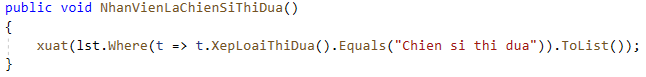
Hình . Cài đặt phương thức NhanVienCoThamNienTren10Nam()



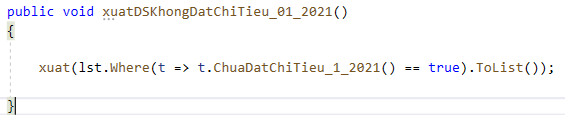
Hình . Cài đặt phương thức TongPCTNTheoNam(nam)



Hình . Cài đặt phương thức TongLuong2021()



Hình . Cài đặt phương thức NhanVienLaChienSiThiDua()



Hình . Cài đặt phương thức xuatDSKhongDatChiTieu\_01\_2021()

### Lớp NhanVien

**-** Đây là lớp con của lớp CongTyELCA:

|  |
| --- |
| **NhanVien** |
| + maNV: String  + tenNV: String  + gioitinh: String  + diachi: String  + maPB: String  + email: String  + sdt: String  + ngaysinh: Datetime  + thoigianvaolam: Datetime  + thoigianlamNVChinhThuc: Datetime  + cty:int  + taoMaNV(): string  + (abstract) Luong(): float  + (abstract) XepLoaiThiDua(): string  + (abstract) TongLuong2021(): float  + (abstract) ChuaDatChiTieu\_1\_2021(): bool  + (abstract) tongPCTheoNam(nam): float  + NhanVien()  + NhanVien(maNV,tenNV,gioitinh, diachi, sdt, thoigianvaolam, thoigianlamNVChinhThuc)  + checkGioiTinh(gt): string  + checkSdt(sdt): string  + checkTuoi(ns): bool  + Thamnien(float) : float  + PhuCapThamnien(): float  + xuat(): virtual |

Bảng . Sơ đồ lớp NhanVien

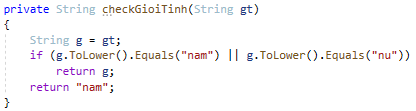
#### Biến

* maNV: Mã nhân viên.
* tenNV: Tên nhân viên.
* gioitinh: Giới tính nhân viên.
* diachi: Địa chỉ nhân viên.
* maPB: Mã phòng ban nhân viên.
* email: Email nhân viên.
* sdt: Số điện thoại nhân viên.
* ngaysinh: Ngày sinh nhân viên.
* thoigianvaolam: Thời gian vào làm.
* thoigianlamNVChinhThuc: Thời gian làm nhân viên chính thức.
* cty: Loại công ty.

#### Phương thức

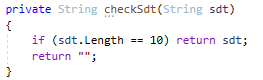
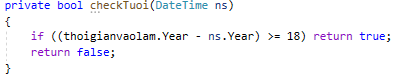
* getMaNV(): Lấy giá trị mã nhân viên.
* setMaNV(): Gán giá trị mã nhân viên.
* getTenNV: Lấy giá trị tên nhân viên.
* setTenNV: Gán giá trị tên nhân viên.
* setGioitinh: Lấy giá trị giới tính nhân viên.
* getGioitinh: Gán giá trị giới tính nhân viên.
* getDiachi: Lấy giá trị địa chỉ nhân viên.
* setDiachi: Gán giá trị địa chỉ nhân viên.
* getMaPB: Lấy giá trị mã phòng ban nhân viên.
* setMaPB: Gán giá trị mã phòng ban nhân viên.
* getEmail: Lấy giá trị Email nhân viên.
* setEmail: Gán giá trị Email nhân viên.
* getSdt: Lấy giá trị số điện thoại nhân viên.
* setSdt: Gán giá trị số điện thoại nhân viên.
* getNgaysinh: Lấy giá trị ngày sinh nhân viên.
* setNgaysinh: Gán giá trị ngày sinh nhân viên.
* getThoigianvaolam: Lấy giá trị thời gian vào làm.
* setThoigianvaolam: Gán giá trị thời gian vào làm.
* getThoigianlamNVChinhThuc: Lấy giá trị thời gian làm nhân viên chính thức.
* setThoigianlamNVChinhThuc: Gán giá trị thời gian làm nhân viên chính thức.
* getCty(): Lấy giá trị loại công ty.
* setCty(): Gán giá trị loại công ty.
* taoMaNV(): string : Tạo random mã nhân viên.
* NhanVien(): Phương thức khởi tạo không tham số
* NhanVien(maNV,tenNV,gioitinh, diachi, sdt, thoigianvaolam, thoigianlamNVChinhThuc): Phương thức khỏi tạo có tham số.
* (abstract) Luong(): float: Phương thức tính lương
* (abstract) XepLoaiThiDua(): string : Xếp loại thi đua.
* (abstract) TongLuong2021(): float : T Xuất tổng lương năm 2021 của **ELCA** 1, 2, 3.
* (abstract) ChuaDatChiTieu\_1\_2021(): bool : Kiểm tra danh sách các nhân viên không đạt chỉ tiêu trong tháng 1/2021 của công ty **ELCA** 1.
* (abstract) tongPCTheoNam(nam): float : Xuất tổng phụ cấp thâm niên trong năm 2021 của **ELCA** 1, 2, 3.
* checkGioiTinh(gt): string : Chỉ nhận thông tin giới tính nam và nữ.
* checkSdt(sdt): string : Chỉ nhận thông tin số điện thoại có đúng 10 số.
* checkTuoi(ns): bool : Chỉ nhận nhưng thông tin của nhân viên có tuổi >18.
* Thamnien(float) : float : Thâm niên.
* PhuCapThamnien(): float: Phụ cấp thâm niên.
* xuat(): virtual : Xuất thông tin.

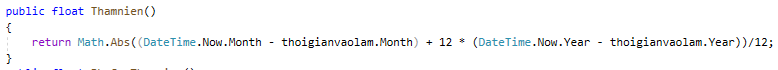
#### Cài đặt phương thức



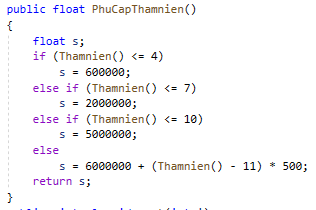
Hình . Cài đặt phương thức checkSdt(sdt)

Hình . Cài đặt phương thức checkGioiTinh(gt)

Hình . Cài đặt theo phương thức checkTuoi(ns)



Hình . Cài đặt phương thức Thamnien()



Hình . Cài đặt phương thức PhuCapThamnien()



Hình . Cài đặt phương thức xuat()

### Lớp NhanVienELCA1

* Đây là lớp kế thừa của lớp NhanVien

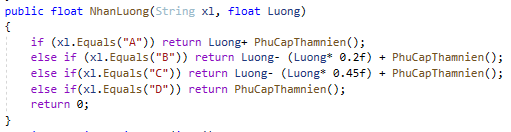
|  |
| --- |
| **NhanVienELCA1** |
| + HESO={ 2.34f, 2.67f, 3f, 3.33f, 3.66f, 3.99f, 4.32f, 4.65f} :float  + LuongCB = 1500000f :float  + NhanVienELCA1: base()  + NhanLuong(xl,Luong):float  + xuat(): override |

Bảng . Sơ đồ lớp NhanVien

#### Phương thức

* HESO={ 2.34f, 2.67f, 3f, 3.33f, 3.66f, 3.99f, 4.32f, 4.65f} :float :Hệ số lương.
* LuongCB = 1500000f :float : Lương cơ bản.
* NhanVienELCA1: base() : Hàm khởi tạo kế thừa lớp NhanVien.
* NhanLuong(xl,Luong):float : Lương nhận được.
* xuat(): override :Xuất thông tin.

#### Cài đặt phương thức



Hình . Cài đặt phương thức NhanLuong(xl,Luong)



Hình . Cài đặt phương thức xuat()

### Lớp NhanVienSanXuat

* Đây là lớp kế thừa lớp NhanVienELCA1:

|  |
| --- |
| **NhanVienSanXuat** |
| + sonn: int  + (static) PhuCapNangNhoc =0.1f : float  + NhanVienSanXuat():base  + (override)Luong(): float  + (override) ChuadatChiTieu\_1\_2021(): bool  + (override) xuat()  + (override) XepLoaiThiDua() : string  + (override) TongLuong2021(): float  + (override) tongPCTheoNam(nam):float |

Bảng . Sơ đồ lớp NhanVienSanXuat

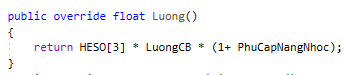
#### Biến

* sonn: Số ngày nghĩ.

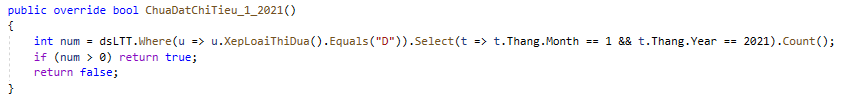
#### Phương thức

* getSonn: Lấy số ngày nghỉ.
* setSonn: Gán số ngày nghỉ.
* (static) PhuCapNangNhoc =0.1f : float : Gán vào phụ cấp nặng nhọc.
* NhanVienSanXuat():base : Hàm khỏi tạo kế thừa class NhanVienELCA1.
* (override)Luong(): float : Xây dựng phương thức tính lương.
* (override) ChuadatChiTieu\_1\_2021(): bool: Xây dựng phương thức xuất danh sách các nhân viên không đạt chỉ tiêu trong tháng 1/2021 của công ty **ELCA** 1.
* (override) xuat(): Xây dựng phương thức xuất thông tin.
* (override) XepLoaiThiDua() : string : Xây dựng phương thức xếp loại thi đua.
* (override) TongLuong2021(): float: Xây dựng phương thức xuất tổng lương năm 2021 của **ELCA** 1, 2, 3.
* (override) tongPCTheoNam(nam):float: Xây dựng phương thức xuất tổng phụ cấp thâm niên trong năm 2021 của **ELCA** 1, 2, 3.

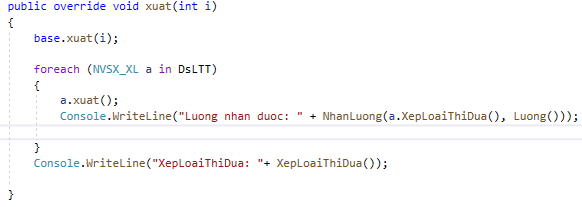
#### Cài đặt phương thức



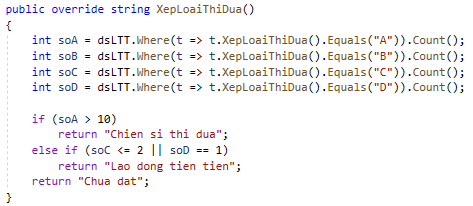
Hình . Cài đặt phương thức Luong()



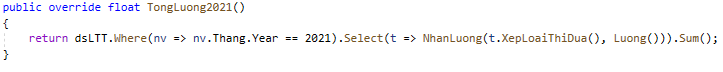
Hình . Cài đặt phương thức ChuaDatChiTieu\_1\_2021()



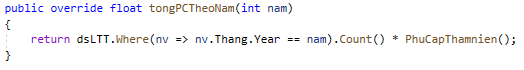
Hình . Cài đặt phương thức xuat()



Hình . Cài đặt phương thức XepLoaiThiDua()



Hình . Cài đặt phương thức TongLuong2021



Hình . Cài đặt phương thức tongPCTheoNam(nam)

### Lớp NhanVienKinhDoanh

* Đây là lớp kế thừa lớp NhanVienELCA1:

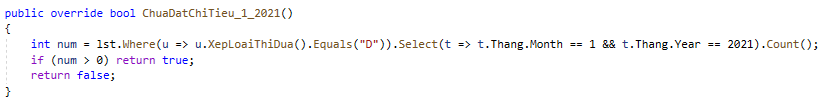
|  |
| --- |
| **NhanVienKinhDoanh** |
| + NhanVienKinhDoanh():base  + (override) ChuadatChiTieu\_1\_2021(): bool  + (override)Luong(): float  + (override) TongLuong2021(): float  + (override) tongPCTheoNam(nam):float  + (override) XepLoaiThiDua() : string  + (override) xuat() |

Bảng . Sơ đồ lớp NhanVienKinhDoanh

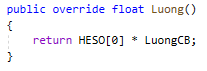
#### Phương thức

* NhanVienKinhDoanh():base : Hàm khỏi tạo kế thừa class NhanVienELCA1.
* (override)Luong(): float : Xây dựng phương thức tính lương.
* (override) ChuadatChiTieu\_1\_2021(): bool: Xây dựng phương thức xuất danh sách các nhân viên không đạt chỉ tiêu trong tháng 1/2021 của công ty **ELCA** 1.
* (override) TongLuong2021(): float: Xây dựng phương thức xuất tổng lương năm 2021 của **ELCA** 1, 2, 3.
* (override) tongPCTheoNam(nam):float: Xây dựng phương thức xuất tổng phụ cấp thâm niên trong năm 2021 của **ELCA** 1, 2, 3.
* (override) XepLoaiThiDua() : string : Xây dựng phương thức xếp loại thi đua.
* (override) xuat(): Xây dựng phương thức xuất thông tin.

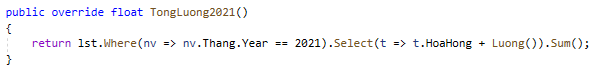
#### Cài đặt phương thức



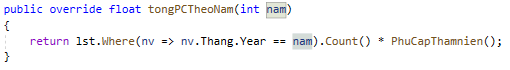
Hình . Cài đặt phương thức ChuaDatChiTieu\_1\_2021()



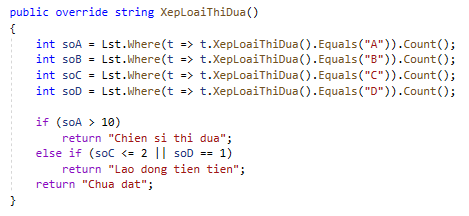
Hình . Cài đặt phương thức Luong()

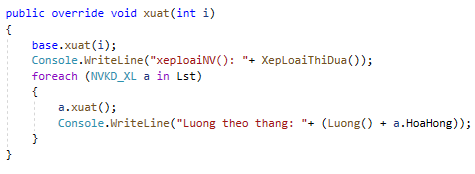


Hình . Cài đặt phương thức TongLuong2021()



Hình . Cài đặt phương thức tongPCTheoNam(nam)



Hình . Cài đặt phương thức XepLoaiThiDua()

Hình . Cài đặt phương thức xuat()

### Lớp CanBoQuanLy

* Đây là lớp kế thừa lớp NhanVienELCA1:

|  |
| --- |
| **CanBoQuanLy** |
| + chucvu: string  + hesochucvu: float  + (override) ChuadatChiTieu\_1\_2021(): bool  + (override)Luong(): float  + (override) TongLuong2021(): float  + (override) tongPCTheoNam(nam):float  + (override) XepLoaiThiDua() : string  + (override) xuat() |

Bảng . Sơ đồ lớp CanBoQuanLy

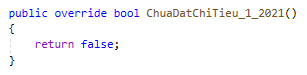
#### Biến

* chucvu: Chức vụ.
* hesochucvu: Hệ số chức vụ.

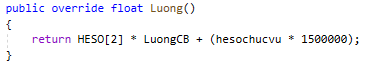
#### Phương thức

* getChucvu(): Gán chức vụ.
* setChucvu(): Lấy chức vụ.
* getHesochucvu(): Gán hệ số chức vụ.
* setHesochucvu(): Lấy hệ số chức vụ.
* (override) ChuadatChiTieu\_1\_2021(): bool: Xây dựng phương thức xuất danh sách các nhân viên không đạt chỉ tiêu trong tháng 1/2021 của công ty **ELCA** 1.
* (override)Luong(): float : Xây dựng phương thức tính lương.
* (override) TongLuong2021(): float: Xây dựng phương thức xuất tổng lương năm 2021 của **ELCA** 1, 2, 3.
* (override) tongPCTheoNam(nam):float: Xây dựng phương thức xuất tổng phụ cấp thâm niên trong năm 2021 của **ELCA** 1, 2, 3.
* (override) XepLoaiThiDua() : string : Xây dựng phương thức xếp loại thi đua.
* (override) xuat(): Xây dựng phương thức xuất thông tin.

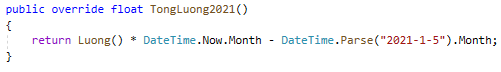
#### Cài đặt phương thức



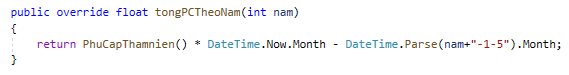
Hình . Cài đặt phương thức ChuaDatChiTieu\_1\_2021()



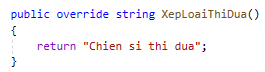
Hình . Cài đặt phương thức Luong()



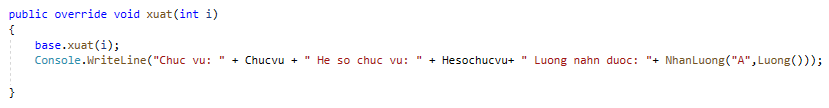
Hình . Cài đặt phương thức TongLuong2021()



Hình . Cài đặt phương thức tongPCTheoNam(nam)



Hình . Cài đặt phương thức XepLoaiThiDua()



Hình . Cài đặt phương thức xuat()

### Lớp NhanVienELCA2

* Đây là lớp kế thừa lớp NhanVien:

|  |
| --- |
| **NhanVienELCA2** |
| + NhanVienELCA2():base()  + (override) ChuadatChiTieu\_1\_2021(): bool  + (override)Luong(): float  + (override) xuat()  + (override) XepLoaiThiDua() : string  + (override) TongLuong2021(): float  + (override) tongPCTheoNam(nam):float |

Bảng . Sơ đồ lớp NhanVienELCA2

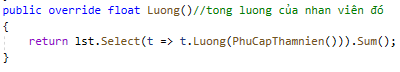
#### Phương thức

* NhanVienELCA2():base() : Hàm khởi tạo kế thừa từ lớp NhanVien.
* (override) ChuadatChiTieu\_1\_2021(): bool: Xây dựng phương thức xuất danh sách các nhân viên không đạt chỉ tiêu trong tháng 1/2021 của công ty **ELCA** 1.
* (override)Luong(): float : Xây dựng phương thức tính lương.
* (override) xuat(): Xây dựng phương thức xuất thông tin.
* (override) XepLoaiThiDua() : string : Xây dựng phương thức xếp loại thi đua.
* (override) TongLuong2021(): float: Xây dựng phương thức xuất tổng lương năm 2021 của **ELCA** 1, 2, 3.
* (override) tongPCTheoNam(nam):float: Xây dựng phương thức xuất tổng phụ cấp thâm niên trong năm 2021 của **ELCA** 1, 2, 3.

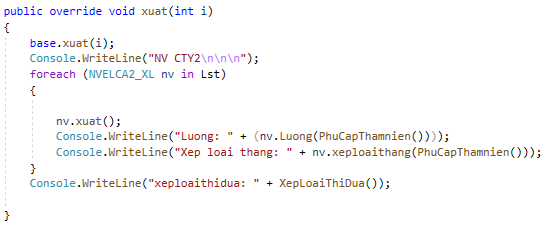
#### Cài đặt phương thức



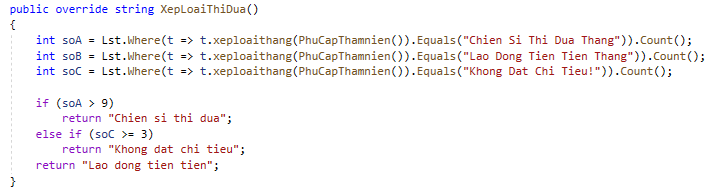
Hình . Cài đặt phương thức ChuaDatChiTieu\_1\_2021()



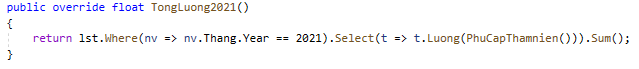
Hình . Cài đặt phương thức Luong()



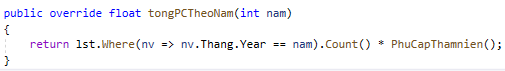
Hình . Cài đặt phương thức xuat()



Hình . Cài đặt phương thức XepLoaiThiDua()



Hình . Cài đặt phương thức TongLuong2021()



Hình . Cài đặt phương thức tongPCTheoNam(nam)

### Lớp NhanVienELCA3

Đây là lớp kế thừa lớp NhanVien:

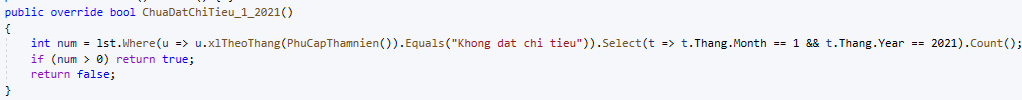
|  |
| --- |
| **NhanVienELCA3** |
| + NhanVienELCA3():base()  + (override) ChuadatChiTieu\_1\_2021(): bool  + (override)Luong(): float  + (override) xuat()  + (override) XepLoaiThiDua() : string  + (override) TongLuong2021(): float  + (override) tongPCTheoNam(nam):float |

Bảng . Sơ đồ lớp NhanVienELCA3

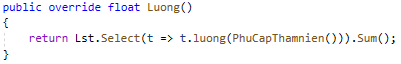
#### Phương thức

* NhanVienELCA3():base() : Hàm khởi tạo kế thừa từ lớp NhanVien.
* (override) ChuadatChiTieu\_1\_2021(): bool: Xây dựng phương thức xuất danh sách các nhân viên không đạt chỉ tiêu trong tháng 1/2021 của công ty **ELCA** 1.
* (override)Luong(): float : Xây dựng phương thức tính lương.
* (override) xuat(): Xây dựng phương thức xuất thông tin.
* (override) XepLoaiThiDua() : string : Xây dựng phương thức xếp loại thi đua.
* (override) TongLuong2021(): float: Xây dựng phương thức xuất tổng lương năm 2021 của **ELCA** 1, 2, 3.
* (override) tongPCTheoNam(nam):float: Xây dựng phương thức xuất tổng phụ cấp thâm niên trong năm 2021 của **ELCA** 1, 2, 3.

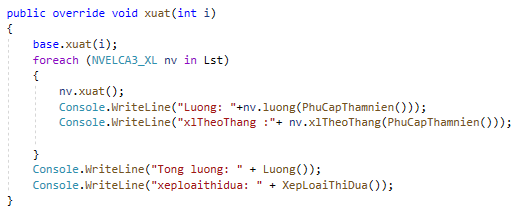
#### Cài đặt phương thức



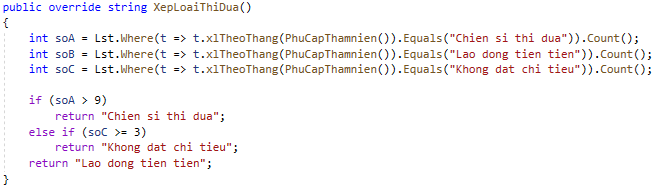
Hình . Cài đặt phương thức ChuaDatChiTieu\_1\_2021()



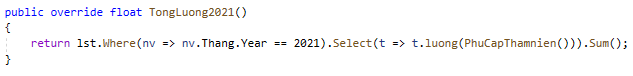
Hình . Cài đặt phương thức Luong()



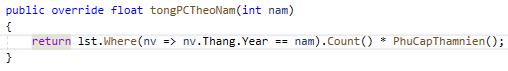
Hình . Cài đặt phương thức xuat()



Hình . Cài đặt phương thức XepLoaiThiDua()



Hình . Cài đặt hương thức TongLuong2021()



Hình . Cài đặt phương thức tongPCTheoNam(nam)

### Lớp QuanLyPhongBan

* Đây là lớp con của lớp CongTyELCA:

|  |
| --- |
| **QuanLyPhongBan** |
| + Them()  + Xoa()  + Sua()  + DocDuLieu()  +xuatDSPhongBan |

Bảng . Sơ đồ lớp QuanLyPhongBan

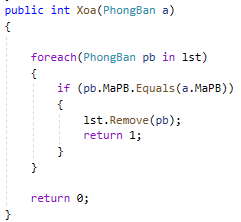
#### Phương thức

* Them(): Thêm dữ liệu.
* Xoa(): Xoá dữ liệu.
* Sua(): Sửa dữ liệu.
* DocDuLieu(): Đọc dữ liệu.
* xuatDSPhongBan: Xuất danh sách phòng ban.

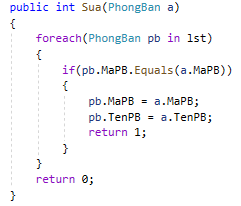
#### Cài đặt phương thức



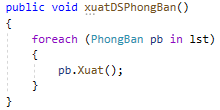
Hình . Cài đặt phương thức Them()



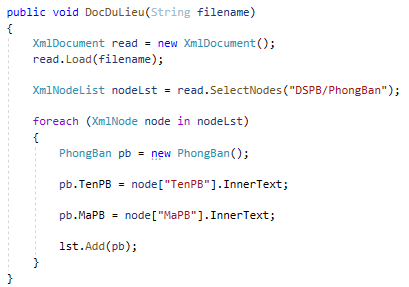
Hình . Cài đặt phương thức Xoa()



Hình . Cài đặt phương thức Sua()



Hình . Cài đặt phương thức xuatDSPhongBan()



Hình . Cài đặt phương thức DocDuLieu()

### Lớp PhongBan

* Đây là lớp con của lớp QuanLyPhongBan

|  |
| --- |
| **PhongBan** |
| + maPB: string  + tenPB: string  + PhongBan(maPB, tenPB)  + checkmaPB(maPB): bool  + xuat() |

Bảng . Sơ đồ lớp PhongBan

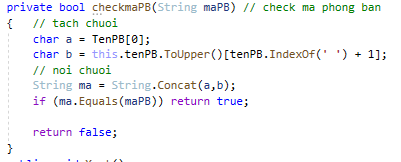
#### Biến

* maPB: Mã phòng ban.
* tenPB: Tên phòng ban

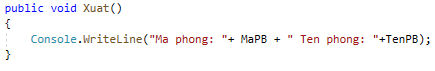
#### Phương thức

* getMaPB: Lấy giá trị mã phòng ban.
* setMaPB: Gán giá trị mã phòng ban.
* getTenPB: Lấy giá trị tên phòng ban.
* setTenPB: Gán giá trị tên phòng ban.
* PhongBan(maPB, tenPB) : Xây dựng hàm khởi tạo.
* checkmaPB(maPB): bool : Kiểm tra lại ràng buộc mã phòng ban
* xuat(): Xuất thông tin.

#### Cài đặt phương thức



Hình . Cài đặt phương thức checkmaPB()



Hình . Cài đặt phương thức xuat()

### Lớp NVELCA2\_XL

* Đây là lớp con của lớp NhanVienELCA2:

|  |
| --- |
| **NVELCA2\_XL** |
| + doanhthudatduoc : float  + thuong: float  + hoahong: float  + thang: Datetime  + doanhthuthang: float  + phucapcongtacphi: float  + xeploaithang(pcThamnien): string  + Luong(phucapthamnien): float  + xuat() |

Bảng . Sơ đồ lớp NVELCA2\_XL

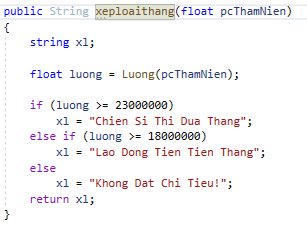
#### Biến

* doanhthudatduoc : Doanh thu đạt được.
* thuong: Tiền thưởng.
* hoahong: Tiền hoa hồng.
* thang: Tháng truyền vào.
* doanhthuthang: Doanh thu tháng.
* phucapcongtacphi: Phụ cấp công tác phí.

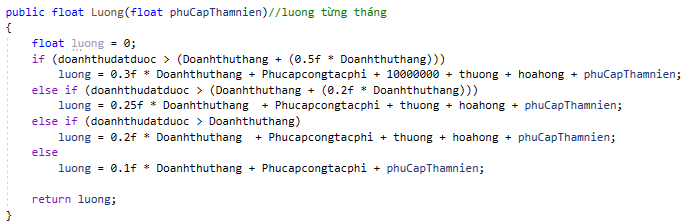
#### Phương thức

* getDoanhthudatduoc : Lấy doanh thu đạt được.
* setDoanhthudatduoc : Gán doanh thu đạt được.
* getThuong: Lấy giá trị tiền thưởng.
* setThuong: Gán giá trị tiền thưởng.
* getHoahong: Lấy giá trị tiền hoa hồng.
* setHoahong: Gán giá trị tiền hoa hồng.
* getThang: Lấy giá trị tháng truyền vào.
* setThang: Gán giá trị tháng truyền vào.
* getDoanhthuthang: Lấy giá trị doanh thu tháng.
* setDoanhthuthang: Gán giá trị doanh thu tháng.
* getPhucapcongtacphi: Lấy giá trị phụ cấp công tác phí.
* setPhucapcongtacphi: Gán giá trị phụ cấp công tác phí.
* xeploaithang(pcThamnien): Xếp loại theo tháng.
* Luong(phucapthamnien): float : Mức lương nhận được.
* xuat(): Xuất thông tin.

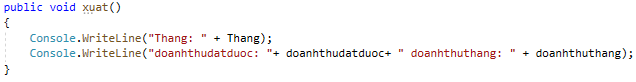
#### Cài đặt phương thức



Hình . Cài đặt phương thức xeploaithang()



Hình . Cài đặt phương thức Luong()



Hình . Cài đặt phương thức xuat()

### Lớp NVELCA3\_XL

* Đây là lớp con của lớp NhanVienELCA3:

|  |
| --- |
| **NVELCA3\_XL** |
| + slKhachHangMoi: int  + slKhachHangToiThieu: int  + phuCapCongTacPhi: float  + thang: Datetime  + xuat()  + xeploaithang(pcThamnien): string  + Luong(phucapthamnien): float |

Hình . Sơ đồ lớp NVELCA3\_XL

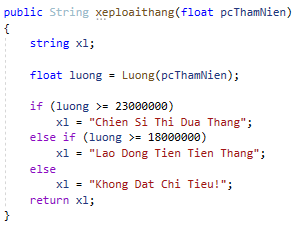
#### Biến

* slKhachHangMoi: Số lượng khách hàng mới.
* slKhachHangToiThieu: Số lượng khách hàng tối thiểu.
* phuCapCongTacPhi: Phụ cấp công tác phí.
* thang: Tháng truyền vào.

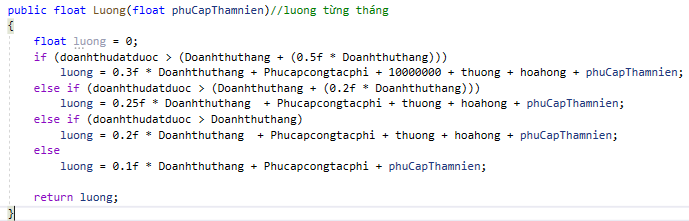
#### Phương thức

* getSlKhachHangMoi: Lấy giá trị khách hàng mới.
* setSlKhachHangMoi: Gán giá trị khách hàng mới.
* getSlKhachHangToiThieu: Lấy giá trị khách hàng tối thiểu.
* setSlKhachHangToiThieu: Gán giá trị khách hàng tối thiểu.
* getThang: Lấy giá trị tháng truyền vào.
* setThang: Gán giá trị tháng truyền vào.
* getPhucapcongtacphi: Lấy giá trị phụ cấp công tác phí.
* setPhucapcongtacphi: Gán giá trị phụ cấp công tác phí.
* xeploaithang(pcThamnien): Xếp loại theo tháng.
* Luong(phucapthamnien): float : Mức lương nhận được.
* xuat(): Xuất thông tin.

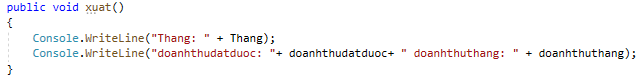
#### Cài đặt phương thức



Hình . Cài đặt phương thức xeploaithang()



Hình . Cài đặt phương thức Luong()



Hình . Cài đặt phương thức xuat()

### Lớp NVKD\_XL

* Đây là lớp con của lớp NhanVienKinhDoanh:

|  |
| --- |
| **NVKD\_XL** |
| **+** doanhSToiThieu: float  + doanhSTheoThang: float  + hoaHong: float  + thang: Datetime  + XepLoaiThiDua(): string  + DTHoaHong(): float  + xuat() |

Bảng . Sơ đồ lớp NVKD\_XL

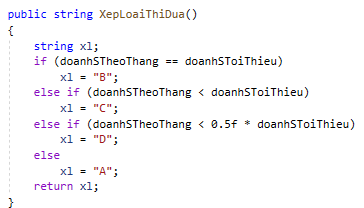
#### Biến

* doanhSToiThieu: Doanh số tối thiểu.
* doanhSTheoThang: Doanh số theo tháng
* hoaHong: Hoa hồng đạt được.
* - thang: Tháng truyền vào.

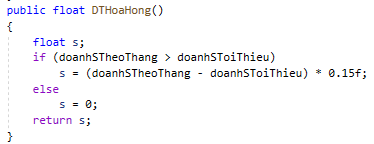
#### Phương thức

* getDoanhSToiThieu: Gán vào giá trị doanh số tối thiểu.
* setDoanhSToiThieu: Lấy giá trị doanh số tối thiểu.
* getDoanhSTheoThang: Gán vào giá trị doanh số theo tháng.
* setDoanhSTheoThang: Lấy giá trị doanh số theo tháng.
* getHoaHong: Gán vào giá trị hoa hồng đạt được.
* setHoaHong: set giá trị hoa hồng đạt đượcc.
* getThang: Lấy giá trị tháng truyền vào.
* setThang: Gán giá trị tháng truyền vào.
* XepLoaiThiDua(): string : Xếp loại thi đua.
* DTHoaHong(): Doanh thu hoa hồng đạt được.
* xuat(): Xuất thông tin.

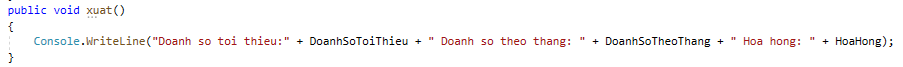
#### Cài đặt phương thức



Hình . Cài đặt phương thức XepLoaiThiDua()



Hình . Cài đặt phương thức DTHoaHong()



Hình . Cài đặt phương thức xuất

### Lớp NVSX\_XL

* Đây là lớp con của lớp NhanVienSanXuat:

|  |
| --- |
| **NVSX\_XL** |
| + sonn: int  + thang: Datetime  + XepLoaiThiDua(): string  + xuat() |

Bảng . Sơ đồ lớp NVSX\_XL

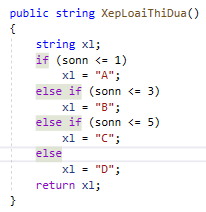
#### Biến

* sonn: Số ngày nghỉ.
* thang: Giá trị tháng truyền vào.

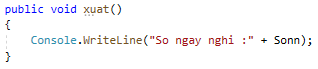
#### Phương thức

* getSonn: Gán vào giá trị số ngày nghỉ.
* setSonn: Lấy giá trị số ngày nghỉ.
* getThang: Lấy giá trị tháng truyền vào.
* setThang: Gán giá trị tháng truyền vào.
* XepLoaiThiDua(): Xếp loại thi đua.
* xuat(): Xuất thông tin.

#### Cài đặt phương thức



Hình . Cài đặt phương thức XepLoaiThiDua()

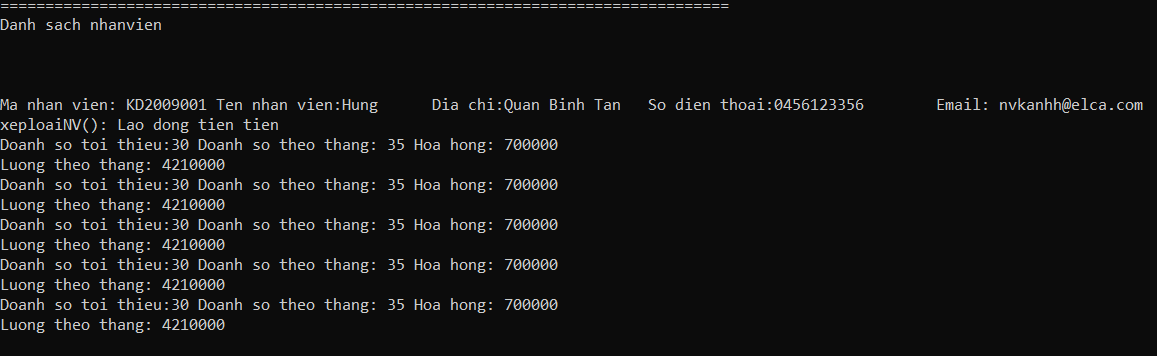


Hình . Cài đặt phương thức Xuat()

### Lớp Program

* Đây là lớp để thực thi chương trình:
* Yêu cầu 1: Thêm mới đơn vị, phòng ban, nhân viên của các công ty. Lưu dữ liệu vào file XML. Đọc dữ liệu từ file XML.

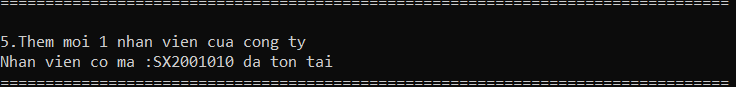
Phía dưới đây là hình ảnh 1 phần kết quả đọc dữ liệu từ file Xml.



Hình . Chạy yêu cầu\_1

* Yêu cầu 2: Thêm mới 1 nhân viên của các công ty

Khi truyền vào 1 mã nhân viên bất kì, nếu mã đã tồn tại -> Thông báo “Nhân viên đã tồn tại”, nếu mã chưa có -> “Thông tin nhân viên được đưa vào”

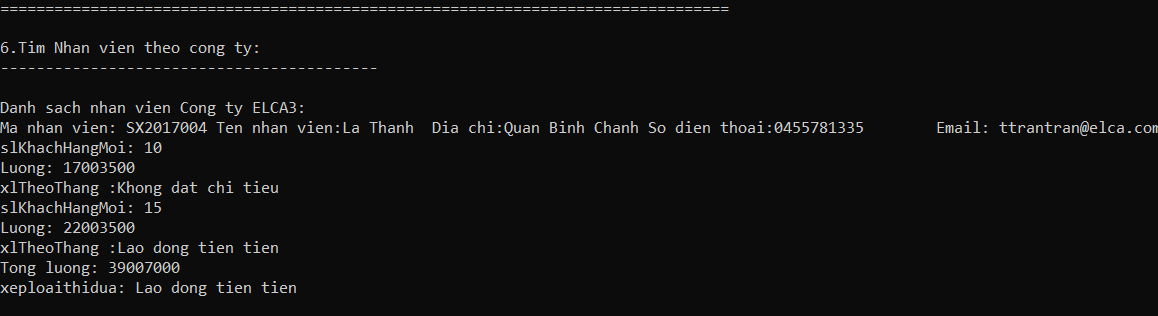


Hình . Chạy yêu cầu\_2

* Yêu cầu 3: Tìm kiếm thông tin nhân viên của mỗi công ty.

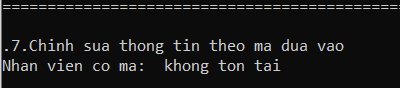
Khi truyền vào tên của một trong ba công ty được hệ thống xây dựng. Thông tin danh sách nhân viên sẽ được xuất ra.

Dưới đây là 1 phần kết quả khi thực thi chương trình:



Hình . Chạy yêu cầu\_3

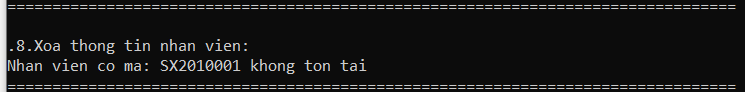
* Yêu cầu 4: Chỉnh sửa thông tin nhân viên theo mã nhân viên đưa vào.



Hình . Chạy yêu cầu\_4

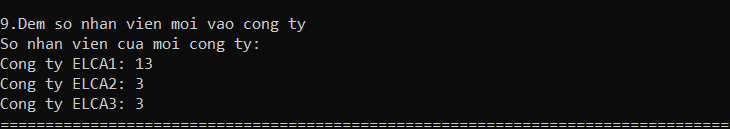
* Yêu cầu 5: Xoá thông tin nhân viên.

Khi chọn một mã nhân viên bất kỳ.Nếu mã tồn tại=>Xoá tất cả thông tin nhân viên. Nếu mã không tồn tại => Xuất thông báo “Mã nhân viên không tồn tại”.



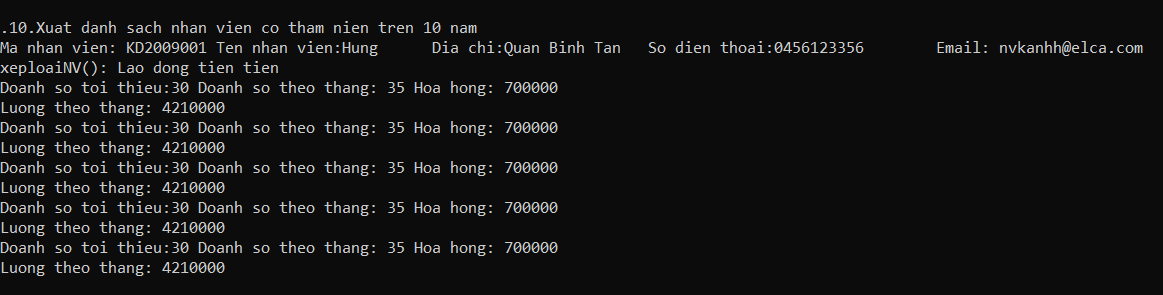
Hình . Chạy yêu cầu\_5

* Yêu cầu 6: Đếm số nhân viên của mỗi công ty .



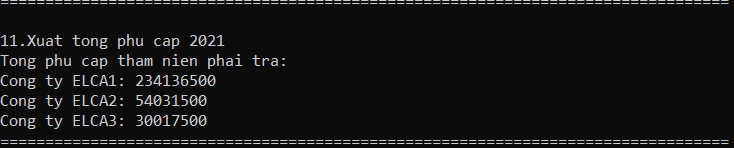
Hình . Chạy yêu cầu\_6

* Yêu cầu 7: Xuất danh sách nhân viên các công ty thành viên có thâm niên trên 10 năm.

Dưới đây là một phần kết quả chạy thử:

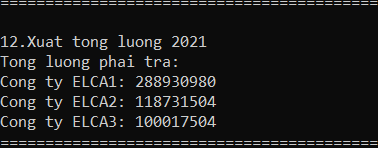
Hình . Chạy yêu cầu\_7

* Yêu cầu 8: Xuất tổng phụ cấp thâm niên trong năm 2021 của **ELCA** 1, 2, 3.



Hình . Chạy yêu cầu\_8

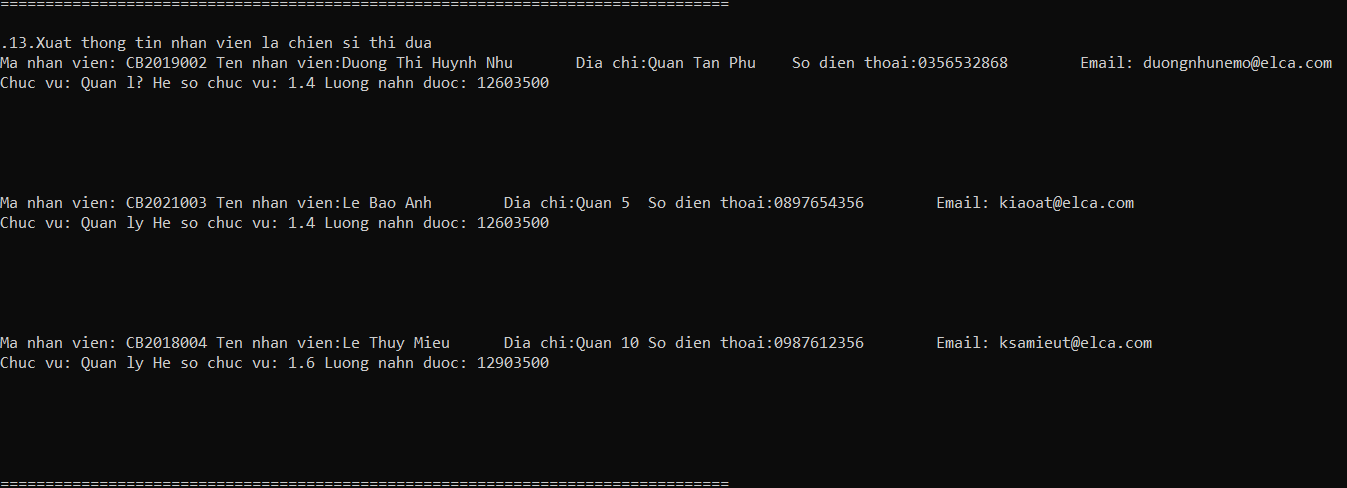
* Yêu cầu 9: Xuất tổng lương năm 2021 của **ELCA** 1, 2, 3.



Hình . Chạy yêu cầu\_9

* Yêu cầu 10: Xuất thông tin các nhân viên là chiến sĩ thi đua của tổng công ty **ELCA**.

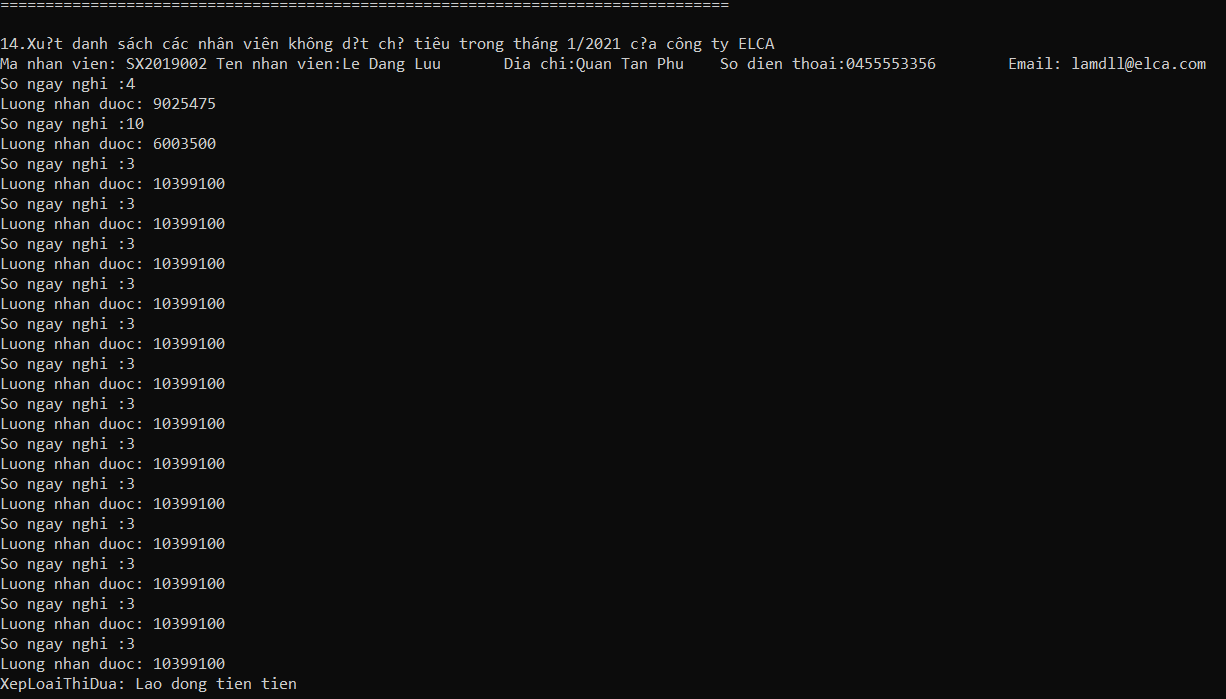
Dưới đây là một phần kết quả chạy thử:



Hình . Chạy yêu cầu\_10

* Yêu cầu 11: Xuất danh sách các nhân viên không đạt chỉ tiêu trong tháng 1/2021 của công ty **ELCA** 1.

Dưới đây là một phần kết quả chạy thử:



Hình . Chạy yêu cầu\_11

* Phía trên em xin chạy thử 1 số yêu cầu do nhưng chương trình code khá dài, để rõ hơn xin thầy hãy xem và nhận xét giúp chúng em source code ạ.

# TỔNG KẾT VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

Qua quá trình học môn Lập trình Hướng đối tượng và việc thực hiện đề tài này, chúng em đã hiểu thêm sâu hơn về việc ứng dụng các phương thức kế thừa và đa hình trong việc xây dựng các ứng dụng mang tính thực tế cao. Tuy nhiên do hạn chế về kiến thức và thời gian nên đề tài của nhóm thực sự chưa được như mong đợi. Chúng em mong nhận được sự đóng góp ý của thầy để đề tài được hoàn thiện hơn nữa.

Chúng em xin chân thành cảm ơn.

* Kết quả đạt được:
* Tìm hiểu về phương thức override và overload trong lập trình hướng đối tượng, phân biệt được tính chất của chúng trong C# và trong Java và nêu được ví dụ của chúng.
* Áp dụng tất cả kiến thức đã học trong việc giải bài toán “Xây dựng chương trình quản lý nhân sự ở công ty”.
* Hầu hết trong giải bài toán đề ứng dụng các thư viên – Mang tính tối ưu cao cho bài toán.
* Hạn chế:
* Kiến thức tìm hiểu vẫn chưa sâu, nhiều khả năng chưa đạt được.
* Chương trình xây dựng còn rối mắt. Chưa hoàn thành hết tất cả các yêu cầu trong chương trình.

# TÀI LIỆU THAM KHẢO

1. Baduyflex (2020). *Truyền tham chiếu và tham trị trong C# như thế nào*. Truy cập ngày 13/09/2021, từ <https://codelearn.io/sharing/>.
2. *Lập trình Java*. Truy cập ngày 15/09/2021, từ <https://www.howkteam.vn/>.
3. Minh Hoàng (2020). *Lập trình C#*. Truy cập ngày 15/09/20201, từ <http://minhhn.com/lap-trinh-c-sharp>.
4. *Slide Bài giảng Công nghệ Java*. Trường Đại học công nghiệp thực phẩm thành phố Hồ Chí Minh.
5. *Slide Bài giảng Lý thuyết & Thực hành Lập trình hướng đối tượng*. Trường Đại học công nghiệp thực phẩm thành phố Hồ Chí Minh.