

Biên soạn: TS. TRẦN CÔNG ÁN (Chủ biên)  
THS. NGUYỄN CÔNG HUY

**GIÁO TRÌNH**

**LẬP TRÌNH  
HƯỚNG ĐỐI TƯỢNG**  
(OBJECT ORIENTED PROGRAMMING)



**NHÀ XUẤT BẢN ĐẠI HỌC CẦN THƠ**  
**2016**

**BIÊN MỤC TRƯỚC XUẤT BẢN THỰC HIỆN BỞI  
TRUNG TÂM HỌC LIỆU TRƯỜNG ĐẠI HỌC CẦN THƠ**

---

Trần, Công Ân

Giáo trình lập trình hướng đối tượng / Trần Công Ân, Nguyễn Công Huy. – Cần Thơ : Nxb. Đại học Cần Thơ, 2016.

250 tr. : minh họa ; 24 cm

Sách có danh mục tài liệu tham khảo

ISBN: 9786049195884

1. Object-oriented programming (computer science) 2. Lập trình hướng đối tượng

I. Nhan đề. II. Nguyễn, Công Huy

005.133 – DDC 23

MFN 209226

A105

# LỜI GIỚI THIỆU

---

Nhằm góp phần làm phong phú nguồn tư liệu phục vụ nghiên cứu, học tập cho bạn đọc và sinh viên Khoa Công nghệ Thông tin và Truyền thông - Trường Đại học Cần Thơ, Nhà xuất bản Đại học Cần Thơ ấn hành và giới thiệu cùng bạn đọc giáo trình “Lập trình hướng đối tượng” do Tiến sĩ Trần Công Án và Thạc sĩ Nguyễn Công Huy biên soạn.

Nội dung giáo trình bao gồm 06 chương, trình bày chi tiết về phương pháp lập trình hướng đối tượng như các khái niệm lớp và đối tượng, các tính chất kế thừa, đa hình và trừu tượng hóa,... Ngôn ngữ Java được sử dụng để minh họa cho phương pháp lập trình này. Do đó, để giúp người đọc chưa từng làm quen với ngôn ngữ Java có thể sử dụng giáo trình này, các tác giả cũng trình bày kiến thức căn bản về ngôn ngữ Java. Thêm vào đó, cuối mỗi chương còn có nhiều bài tập hữu ích cho bạn đọc bao gồm các bài tập củng cố lý thuyết và các bài tập thực hành. Giáo trình này là tài liệu học tập có giá trị cho sinh viên các chuyên ngành có liên quan đến ngành Công nghệ thông tin.

Nhà xuất bản Đại học Cần Thơ chân thành cảm ơn các tác giả và sự đóng góp ý kiến của quý thầy cô trong Hội đồng thẩm định trường Đại học Cần Thơ để giáo trình “Lập trình hướng đối tượng” được ra mắt bạn đọc.

Nhà xuất bản Đại học Cần Thơ trân trọng giới thiệu đến sinh viên, giảng viên và bạn đọc giáo trình này.

NHÀ XUẤT BẢN ĐẠI HỌC CẦN THƠ



# LỜI NÓI ĐẦU

---

*Lập trình hướng đối tượng* (Object Oriented Programming, OOP) là một phương pháp lập trình được sử dụng bởi hầu hết các ngôn ngữ lập trình phổ dụng hiện nay. Đây là phương pháp lập trình được đề xuất bởi Dr. Alan Kay vào những năm 70s. Mô hình mà phương pháp lập trình này đề xuất dựa trên khái niệm *lớp* và *đối tượng*, trong đó chương trình là một tập các đối tượng tương tác lẫn nhau. Có thể nói hướng đối tượng đã mang lại một cuộc cách mạng trong phương pháp lập trình. Sở dĩ phương pháp lập trình này trở thành một phương pháp lập trình phổ biến nhất hiện nay là vì mô hình mà phương pháp lập trình này lựa chọn là một sự trừu tượng hóa mạnh mẽ và linh hoạt dựa trên khái niệm đối tượng. Bản chất của thế giới thật là sự tương tác giữa các sự vật hay đối tượng, vì vậy việc lựa chọn mô hình đối tượng để mô hình hóa các vấn đề cần giải quyết giúp cho việc mô hình hóa các bài toán cần giải quyết vào trong mô hình của máy tính dễ dàng và linh hoạt. Điều này có ý nghĩa đặc biệt quan trọng trong bối cảnh hiện nay, khi mà các hệ thống máy tính trở nên mạnh mẽ để không những được sử dụng để tính toán mà còn được sử dụng để giải quyết nhiều vấn đề trong thực tế.

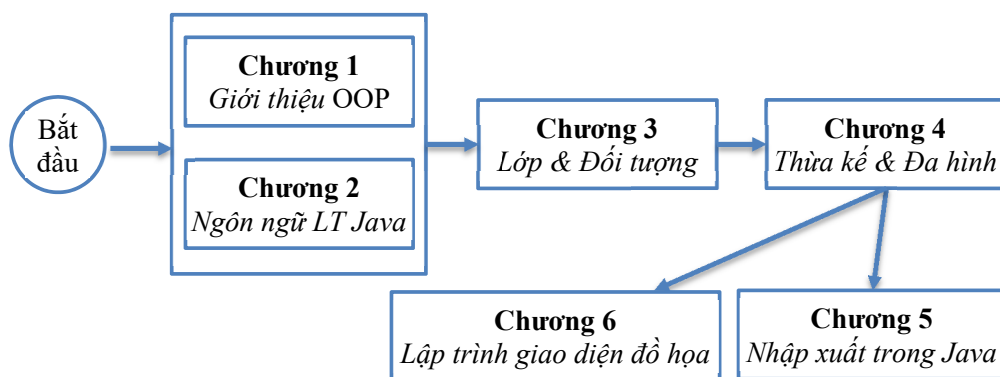
Chính vì vậy, *Lập trình hướng đối tượng* trở thành một môn học chuyên ngành rất quan trọng trong chương trình đào tạo kỹ sư ngành công nghệ thông tin. Mặc dù nguồn tài liệu tham khảo của phương pháp lập trình này rất phong phú, đa phần các tài liệu tham khảo là tiếng Anh. Vì vậy, chúng tôi đã thực hiện biên soạn quyển giáo trình *Lập trình hướng đối tượng* này, nhằm làm phong phú thêm nguồn tài liệu tiếng Việt cho môn học này. Giáo trình này đặc biệt hướng đến độc giả là sinh viên chuyên ngành công nghệ thông tin. Nội dung giáo trình sẽ đi sâu vào giới thiệu tư duy của phương pháp lập trình hướng đối tượng và dùng ngôn ngữ lập trình Java để minh họa.

Ngôn ngữ Java được lựa chọn để minh họa cho phương pháp lập trình hướng đối tượng trong giáo trình này vì đây là một ngôn ngữ đơn giản, thân thiện và đặc biệt là *thuần hướng đối tượng*. Sự đơn giản của ngôn ngữ này giúp người học có thể tập trung vào tìm hiểu phương pháp lập trình, không bị phân tán bởi rào cản của ngôn ngữ lập trình. Ngoài ra, đây cũng là ngôn ngữ lập trình được sử dụng bởi nhiều môn học khác trong chương trình đào tạo kỹ sư công nghệ thông tin.

Kiến thức và phương pháp trình bày trong giáo trình này dựa trên kinh nghiệm học tập, giảng dạy học phần *Lập trình hướng đối tượng* và *Ngôn ngữ lập trình Java* của chúng tôi trong suốt từ những năm 2000 đến nay. Nội dung của giáo trình được chia thành 5 chương như sau:

- ✓ *Chương 1 – Lập trình Hướng đối tượng*: Trình bày tư duy cùng với các khái niệm, tính chất của phương pháp lập trình hướng đối tượng.
- ✓ *Chương 2 – Ngôn ngữ lập trình Java*: Trình bày các thành phần cơ bản của ngôn ngữ lập trình Java.
- ✓ *Chương 3 – Lớp và đối tượng*: Trình bày cách tạo lớp, đối tượng và việc khởi tạo cho các đối tượng trong Java. Chương này cũng trình bày phương pháp xử lý ngoại lệ trong Java.
- ✓ *Chương 4 – Thừa kế và đa hình*: Trình bày phương pháp sử dụng tính thừa kế trong Java để sử dụng lại mã và tổ chức lớp. Đồng thời, giải thích về ứng dụng của tính đa hình trong quản lý các đối tượng.
- ✓ *Chương 5 – Nhập xuất trong Java*: Trình bày các phương pháp nhập xuất trong Java, qua đó giúp nâng cao kỹ năng sử dụng lớp, đối tượng.
- ✓ *Chương 6 – Lập trình giao diện đồ họa*: Trình bày cách thức xây dựng giao diện đồ họa ở mức độ cơ bản. Mục đích chính của chương này giúp củng cố một số khái niệm trong lập trình hướng đối tượng như thừa kế, giao diện, lớp trừu tượng.

Các chương của giáo trình có thể được đọc theo thứ tự được trình bày trong giáo trình. Tuy nhiên, người đọc cũng có thể tiếp cận ngôn ngữ Java trước rồi mới đến phương pháp lập trình hướng đối tượng. Thứ tự tiếp cận các chương của giáo trình được đề xuất như sau:



Chúng tôi gửi lời cảm ơn sâu sắc đến quý Thầy Cô khoa Công nghệ thông tin và Truyền thông đã nhiệt tình góp ý cho nội dung của giáo trình này.

Do thời gian có hạn và lần đầu tiên biên soạn, tài liệu không thể tránh khỏi những thiếu sót. Chúng tôi mong nhận được sự góp ý chân thành từ quý độc giả để quyền giáo trình ngày càng hoàn thiện hơn.

Cần Thơ, tháng 6 năm 2016

**NHÓM TÁC GIẢ**

# MỤC LỤC

---

<b>Lời nói đầu .....</b>	<b>i</b>
<b>Mục lục .....</b>	<b>iii</b>
<b>Danh mục hình ảnh .....</b>	<b>vii</b>
<b>Danh mục bảng .....</b>	<b>xi</b>
<b>Danh mục thuật ngữ tiếng anh .....</b>	<b>xiii</b>
<b>Chương 1 Lập trình Hướng đối tượng .....</b>	<b>1</b>
1.1 Lịch sử của ngôn ngữ và phương pháp lập trình .....	1
1.1.1 <i>Ngôn ngữ máy</i> .....	1
1.1.2 <i>Hợp ngữ</i> .....	2
1.1.3 <i>Ngôn ngữ lập trình cấp cao</i> .....	2
1.1.4 <i>Ngôn ngữ lập trình hướng đối tượng</i> .....	4
1.2 Giới thiệu phương pháp lập trình hướng đối tượng .....	5
1.2.1 <i>Lịch sử ra đời</i> .....	5
1.2.2 <i>Ý tưởng cơ bản</i> .....	5
1.2.3 <i>Các đặc trưng cơ bản của lập trình hướng đối tượng</i> .....	9
1.3 Các khái niệm cơ bản của lập trình hướng đối tượng .....	10
1.3.1 <i>Đối tượng</i> .....	11
1.3.2 <i>Trừu tượng hóa</i> .....	12
1.3.3 <i>Thông điệp và truyền thông điệp</i> .....	13
1.3.4 <i>Lớp</i> .....	14
1.4 Các tính chất của Lập trình Hướng đối tượng .....	17
1.4.1 <i>Tính bao gói</i> .....	17
1.4.2 <i>Tính kế thừa</i> .....	18
1.4.3 <i>Tính đa hình</i> .....	21
1.5 Tổng kết .....	23
1.6 Bài tập .....	24
Tài liệu tham khảo .....	26
<b>Chương 2 Ngôn ngữ Lập trình Java .....</b>	<b>27</b>
2.1 Giới thiệu .....	27
2.2 Kiến trúc của nền tảng Java .....	28
2.2.1 <i>Các thành phần của nền tảng Java</i> .....	28
2.2.2 <i>Cài đặt môi trường Java</i> .....	30
2.3 Dịch và thực thi một chương trình Java .....	32
2.3.1 <i>Qui trình phát triển một chương trình Java</i> .....	32
2.3.2 <i>Cấu trúc một chương trình Java</i> .....	32
2.3.3 <i>Biên dịch và thực thi một chương trình Java</i> .....	34
2.4 Các thành phần cơ bản của Java .....	37

2.4.1	<i>Câu lệnh và khối lệnh</i> .....	37
2.4.2	<i>Biến và kiểu dữ liệu</i> .....	38
2.4.3	<i>Kiểu dữ liệu kiểu chuỗi</i> .....	40
2.4.4	<i>Hằng giá trị</i> .....	42
2.4.5	<i>Biểu thức</i> .....	44
2.4.6	<i>Ép kiểu</i> .....	45
2.4.7	<i>Các toán tử trong Java</i> .....	45
2.4.8	<i>Chú thích</i> .....	51
2.5	<i>Căn bản về nhập xuất trong Java</i> .....	52
2.5.1	<i>Xuất dữ liệu ra màn hình</i> .....	52
2.5.2	<i>Nhập dữ liệu từ bàn phím</i> .....	52
2.6	<i>Cấu trúc điều khiển</i> .....	54
2.6.1	<i>Cấu trúc rẽ nhánh</i> .....	54
2.6.2	<i>Cấu trúc lặp</i> .....	58
2.7	<i>Mảng</i> .....	61
2.7.1	<i>Khai báo biến mảng</i> .....	61
2.7.2	<i>Khởi tạo mảng</i> .....	62
2.7.3	<i>Sử dụng mảng</i> .....	62
2.7.4	<i>Mảng nhiều chiều</i> .....	63
2.8	<i>Tổng kết</i> .....	64
2.9	<i>Bài tập</i> .....	65
	<i>Tài liệu tham khảo</i> .....	69
	<b>Chương 3 Lớp và đối tượng</b> .....	<b>71</b>
3.1	<i>Tạo lớp và đối tượng</i> .....	71
3.1.1	<i>Tạo lớp</i> .....	71
3.1.2	<i>Tạo đối tượng</i> .....	76
3.1.3	<i>Tái định nghĩa phương thức</i> .....	78
3.1.4	<i>Các thành phần tĩnh</i> .....	79
3.1.5	<i>Biến tham chiếu this</i> .....	80
3.1.6	<i>Truyền tham số cho phương thức</i> .....	83
3.2	<i>Mảng các đối tượng</i> .....	86
3.3	<i>Khởi tạo và hủy đối tượng</i> .....	87
3.3.1	<i>Phương thức xây dựng</i> .....	87
3.3.2	<i>Phép gán đối tượng, sao chép cạn và sao chép sâu</i> .....	89
3.3.3	<i>Phương thức xây dựng sao chép</i> .....	98
3.3.4	<i>Hủy đối tượng</i> .....	99
3.4	<i>Gói</i> .....	99
3.4.1	<i>Tạo gói</i> .....	100
3.4.2	<i>Sử dụng gói</i> .....	102
3.4.3	<i>Biên dịch và thực thi chương trình có sử dụng gói</i> .....	103
3.4.4	<i>Chỉ định truy cập với gói</i> .....	105



3.5 Xử lý ngoại lệ .....	105
3.5.1 <i>Xử lý ngoại lệ</i> .....	105
3.5.2 <i>Truyền tiếp ngoại lệ</i> .....	110
3.5.3 <i>Phát sinh ngoại lệ và định nghĩa ngoại lệ</i> .....	112
3.6 Tổng kết .....	114
3.7 Bài tập .....	115
Tài liệu tham khảo .....	120
<b>Chương 4 Thừa kế, đa hình và trừu tượng hóa .....</b>	<b>121</b>
4.1 Thừa kế trong Java.....	121
4.1.1 <i>Khai báo thừa kế</i> .....	122
4.1.2 <i>Nạp đè phương thức và từ khóa super</i> .....	125
4.1.3 <i>Cây thừa kế và lớp Object</i> .....	127
4.2 Phương thức xây dựng trong thừa kế.....	129
4.3 Quan hệ thành phần và quan hệ thừa kế .....	131
4.4 Đa hình và liên kết động .....	133
4.5 Phương thức trừu tượng và Lớp trừu tượng.....	137
4.5.1 <i>Phương thức trừu tượng</i> .....	137
4.5.2 <i>Lớp trừu tượng</i> .....	137
4.6 Giao diện.....	140
4.7 Đa thừa kế.....	142
4.8 Áp dụng Thừa kế và Lớp trừu tượng trong thiết kế lớp.....	145
4.9 Tổng kết .....	149
4.10 Bài tập .....	150
Tài liệu tham khảo .....	156
<b>Chương 5 Nhập xuất (I/O) trong Java .....</b>	<b>157</b>
5.1 Giới thiệu .....	157
5.2 Lớp File.....	157
5.3 Các dòng nhập xuất.....	159
5.3.1 <i>Khái niệm về dòng</i> .....	159
5.3.2 <i>Phân loại dòng nhập xuất</i> .....	160
5.4 Các dòng nhập xuất theo byte.....	161
5.4.1 <i>Lớp InputStream</i> .....	162
5.4.2 <i>Lớp OutputStream</i> .....	163
5.4.3 <i>Nhập xuất tập tin</i> .....	163
5.4.4 <i>Nối kết các dòng nhập xuất</i> .....	165
5.5 Các dòng nhập xuất theo ký tự .....	167
5.5.1 <i>Lớp Reader và Writer</i> .....	167
5.5.2 <i>Lớp BufferedReader và BufferedWriter</i> .....	168
5.5.3 <i>Lớp PrintWriter</i> .....	169
5.6 Nhập xuất tuần tự đối tượng .....	170

5.7 Tập tin truy cập ngẫu nhiên .....	172
5.8 Tổng kết .....	174
5.9 Bài tập .....	175
Tài liệu tham khảo .....	178
<b>Chương 6 Lập trình giao diện đồ họa .....</b>	<b>179</b>
6.1 Giới thiệu lập trình giao diện trong Java .....	179
6.2 Tạo một ứng dụng GUI.....	180
6.3 Các lớp vật chứa .....	181
6.3.1 <i>JFrame</i> .....	182
6.3.2 <i>JDialog</i> .....	184
6.3.3 <i>Các lớp vật chứa cấp cao khác</i> .....	185
6.4 Các thành phần giao diện Swing.....	187
6.4.1 <i>JLabel</i> .....	189
6.4.2 <i>JTextField</i> .....	190
6.4.3 <i>JTextArea</i> .....	190
6.4.4 <i>JButton</i> .....	191
6.4.5 <i>JCheckBox</i> .....	192
6.4.6 <i>JRadioButton</i> .....	194
6.4.7 <i>JList</i> .....	194
6.4.8 <i>JComboBox</i> .....	196
6.5 Sắp xếp bố cục .....	198
6.5.1 <i>FlowLayout</i> .....	199
6.5.2 <i>BorderLayout</i> .....	199
6.5.3 <i>GridLayout</i> .....	201
6.5.4 <i>Kết hợp nhiều cách sắp xếp bố cục bằng cách sử dụng JPanel</i> .....	202
6.6 Xử lý sự kiện.....	204
6.7 Trình đơn, thanh công cụ và hộp thoại .....	209
6.7.1 <i>Trình đơn</i> .....	209
6.7.2 <i>Thanh công cụ</i> .....	212
6.7.3 <i>Hộp thoại</i> .....	213
6.8 Mô hình MVC.....	215
6.9 Tổng kết .....	218
6.10 Bài tập .....	219
Tài liệu tham khảo .....	222
<b>Phụ lục 1 – Các lệnh cơ bản của Java .....</b>	<b>223</b>
<b>Phụ lục 2 – Java Collection Framework .....</b>	<b>227</b>

# DANH MỤC HÌNH ẢNH

---

Hình 1.1	Ví dụ về một đoạn mã trong ngôn ngữ máy .....	1
Hình 1.2	Một chương trình viết bằng hợp ngữ (hiển thị “Hello, world!”).....	2
Hình 1.3	Chương trình hiển thị các dấu “*” viết bằng ngôn ngữ Basic.....	2
Hình 1.4	Chương trình tính giá trị tuyệt đối viết bằng ngôn ngữ lập trình C .....	3
Hình 1.5	Chương trình giải phương trình bậc 2 được chia thành nhiều hàm.....	6
Hình 1.6	Một hình ảnh trong trò chơi Plants and Zombies (PopCap®) .....	7
Hình 1.7	Một số đối tượng với các thuộc tính và hành vi.....	8
Hình 1.8	Cấu tạo của đèn huỳnh quang .....	9
Hình 1.9	Các đối tượng với thuộc tính, hành vi và định danh .....	12
Hình 1.10	Trừu tượng hóa tập trung vào các đặc tính mô tả bản chất của đối tượng, từ góc nhìn của người sử dụng [1].....	13
Hình 1.11	Việc truyền thông điệp “Tăng_tốc” đến đối tượng myBMW chỉ ảnh hưởng đến tốc độ của đối tượng này.....	14
Hình 1.12	Lớp xe Ôtô với các thuộc tính và phương thức.....	15
Hình 1.13	Thuộc tính của một đối tượng là một đối tượng khác.....	16
Hình 1.14	Lớp Diem2D với các thành phần được gán chỉ định truy cập .....	16
Hình 1.15	Lớp Stack với các thành phần dữ liệu riêng.....	17
Hình 1.16	Thừa kế để dùng lại mã – xây dựng lớp mới từ những lớp đã có .....	19
Hình 1.17	Các thành phần của một đối tượng thuộc lớp ColorWindow.....	20
Hình 1.18	Dùng tính kế thừa để tổ chức lớp .....	21
Hình 1.19	Các lớp con nạp đề phương thức lớp cha.....	22
Hình 2.1	Các ngôn ngữ phổ biến nhất năm 2015 (Nguồn: IEEE) .....	27
Hình 2.2	Kiến trúc của nền tảng Java (Nguồn: Oracle).....	29
Hình 2.3	Kiểm tra phiên bản của trình biên dịch và máy ảo Java.....	31
Hình 2.4	Quá trình phát triển một chương trình Java .....	32
Hình 2.5	Cấu trúc một chương trình Java .....	33
Hình 2.6	Một chương trình Java bao gồm nhiều hàm.....	33
Hình 2.7	Một chương trình Java có sử dụng các lớp trong thư viện Swing.....	34
Hình 2.8	Quá trình biên dịch và thực thi một chương trình Java.....	35
Hình 2.9	Giao diện môi trường phát triển tích hợp NetBeans .....	36
Hình 2.10	Thực thi chương trình với đối số dòng lệnh.....	37
Hình 3.1	Cấu trúc các đối tượng trong bộ nhớ.....	81
Hình 3.2	Truyền tham số kiểu nguyên thủy .....	84
Hình 3.3	Truyền tham số kiểu tham chiếu .....	85
Hình 3.4	Phép gán tham chiếu .....	89
Hình 3.5	So sánh kết quả của sao chép tham chiếu và sao chép cạn .....	92

Hình 3.6 Minh họa phép sao chép cạn .....	93
Hình 3.7 So sánh kết quả các phương pháp sao chép đối tượng.....	96
Hình 3.8 Các lớp sử dụng lồng ghép nhau nhiều cấp.....	96
Hình 3.9 Cấu trúc phân cấp một số gói của Apache Software Foundation .....	101
Hình 3.10 Cấu trúc thư mục tương ứng với các gói trong Hình 3.9 .....	101
Hình 3.11 Biên dịch các lớp trong Ví dụ 3.22 .....	104
Hình 3.12. Cây phân cấp các loại ngoại lệ trong Java .....	107
Hình 3.13 Kết quả thực thi Ví dụ 3.23 .....	107
Hình 3.14 Kết quả thực thi Ví dụ 3.24 .....	108
Hình 4.1 Quan hệ giữa lớp Côn trùng với lớp Ong vò vè và Châu chấu .....	121
Hình 4.2 Lớp SinhVien thừa kế từ lớp ConNguoi .....	122
Hình 4.3 Cây phân cấp lớp và lớp Object .....	128
Hình 4.4 Ứng xử của các đối tượng thuộc các lớp khác nhau cho cùng 1 phương thức .....	133
Hình 4.5 Kết quả thực thi Ví dụ 4.13 .....	140
Hình 4.6 Ví dụ về quan hệ đa thừa kế.....	142
Hình 4.7 Vấn đề kim cương trong đa thừa kế .....	143
Hình 4.8 Giả lập đa thừa kế sử dụng quan hệ thành phần và giao diện .....	144
Hình 4.9 Dùng giao diện để thiết đặt ràng buộc tính năng của các lớp .....	146
Hình 4.10 Kết quả thực thi Ví dụ 4.16 .....	149
Hình 5.1 Kết quả thực thi chương trình trong Ví dụ 5.2 .....	159
Hình 5.2 Dòng nhập .....	160
Hình 5.3 Dòng xuất .....	160
Hình 5.4 Phân loại dòng nhập xuất .....	161
Hình 5.5 Sơ đồ thừa kế các dòng nhập xuất theo byte.....	161
Hình 5.6 Kết quả ghi file dùng FileOutputStream .....	164
Hình 5.7 Nối kết các dòng nhập xuất.....	165
Hình 5.8 Sơ đồ thừa kế các dòng nhập xuất theo ký tự .....	167
Hình 5.9 Con trỏ file dùng trong RandomAccessFile.....	173
Hình 6.1 Vật chứa và các thành phần giao diện.....	180
Hình 6.2 Giao diện của chương trình trong Ví dụ 6.1.....	181
Hình 6.3 Sơ đồ thừa kế Container.....	182
Hình 6.4 Cấu trúc tổng quát của một JFrame.....	182
Hình 6.5 Một ứng dụng JFrame đơn giản .....	184
Hình 6.6 Cửa sổ ứng dụng JFrame khi không đặt kích thước.....	184
Hình 6.7 Một ứng dụng JDialog đơn giản.....	185
Hình 6.8 Kết quả ứng dụng sử dụng JFileChooser .....	186
Hình 6.9 Hộp thoại JColorChooser.....	187
Hình 6.10 Sơ đồ thừa kế các thành phần giao diện.....	188

Hình 6.11	Giao diện ứng dụng minh họa cách tạo và sử dụng checkbox .....	193
Hình 6.12	Cách sắp xếp bố cục BorderLayout .....	199
Hình 6.13	Cách sắp xếp bố cục BorderLayout .....	200
Hình 6.14	Cách sắp xếp bố cục GridLayout .....	201
Hình 6.15	Cách sắp xếp bố cục GridLayout với hgap và vgap khác 0 .....	202
Hình 6.16	Vận dụng nhiều cách sắp xếp bố cục .....	203
Hình 6.17	Sơ đồ thừa kế của các lớp sự kiện .....	205
Hình 6.18	Xử lý sự kiện click nút bấm .....	208
Hình 6.19	Xử lý sự kiện chuột di chuyển và click .....	209
Hình 6.20	Hệ thống trình đơn .....	209
Hình 6.21	Kết quả của chương trình minh họa JMenu .....	212
Hình 6.22	Kết quả của chương trình minh họa JToolBar .....	213
Hình 6.23	Hộp thoại dạng JOptionPane.showMessageDialog() .....	214
Hình 6.24	Hộp thoại dạng JOptionPane.showConfirmDialog() .....	214
Hình 6.25	Hộp thoại dạng JOptionPane.showInputDialog() .....	214
Hình 6.26	Hộp thoại dạng JOptionPane.showOptionDialog() .....	215
Hình 6.27	Mô hình MVC (Nguồn: Oracle) .....	215
Hình 6.28	Kết quả của chương trình minh họa mô hình MVC .....	218



# DANH MỤC BẢNG

---

Bảng 2.1	Một số công cụ phát triển của Java.....	30
Bảng 2.2	Các kiểu dữ liệu nguyên thủy trong Java.....	39
Bảng 2.3	Một số phương thức của lớp String .....	41
Bảng 2.4	Một số ký tự escape thông dụng .....	43
Bảng 2.5	Các toán tử số học.....	46
Bảng 2.6	Toán tử tăng, giảm .....	46
Bảng 2.7	Các toán tử so sánh trong Java.....	47
Bảng 2.8	Toán tử luận lý.....	47
Bảng 2.9	Các toán tử trên bit.....	48
Bảng 2.10	Các toán tử dịch bit.....	49
Bảng 2.11	Độ ưu tiên của các toán tử .....	51
Bảng 3.1	Các chỉ định truy cập trong Java.....	72
Bảng 3.2	Các hình thức truyền tham số .....	83
Bảng 3.3	Tác dụng của chỉ định truy cập với gói .....	105
Bảng 4.1	Một số phương thức của lớp Object .....	128





# DANH MỤC THUẬT NGỮ TIẾNG ANH

---

OOP	Object Oriented Programming
MVC	Model – View – Controller
ADT	Abstract Data Type
Abstraction	Trừu tượng hóa
Data abstraction	Trừu tượng hóa dữ liệu
Functional abstraction	Trừu tượng hóa chức năng
Access modifier	Chỉ định truy cập
Attribute	Thuộc tính
Behaviour	Hành vi
Binding	Nối kết
Dynamic binding	Nối kết động
Static binding	Nối kết tĩnh
Class	Lớp
Base class	Lớp cơ sở
Class hierarchy	Cây phân cấp lớp
Derived class	Lớp dẫn xuất
Subclass	Lớp con
Superclass	Lớp cha
Constructor	Phương thức xây dựng
Copy constructor	Phương thức xây dựng sao chép
Default constructor	Phương thức xây dựng mặc nhiên
Declaration	Khai báo
Encapsulation	Tính bao gói
Exception	Ngoại lệ
Garbage collector	Bộ thu hồi tài nguyên
Generalize	Tổng quát hóa
Identifier	Định danh
Implementation	Sự cài đặt
Inheritance	Tính thừa kế
Multiple inheritance	Đa thừa kế
Instance	Thể hiện
Interface	Giao diện
Message	Thông điệp
Message passing	Truyền thông điệp

Method	Phương thức / Hàm
Method signature	Chữ ký phương thức
Method overloading	Nạp chồng/Tái định nghĩa phương thức
Method overriding	Nạp đè phương thức
Method prototype	Nguyên mẫu của phương thức
Static method	Phương thức tĩnh
Object	Đối tượng
Package	Gói
Polymorphism	Tính đa hình
Property	Thuộc tính
Static property	Thuộc tính tĩnh
Reference	Tham chiếu
Reusability	Sự sử dụng lại
Specialize	Đặc trưng hóa
Stream	Dòng
Byte stream	Dòng nhập xuất theo byte
Character stream	Dòng nhập xuất theo ký tự
Input stream	Dòng nhập
Output stream	Dòng xuất
Type casting	Ép kiểu
Implicit type casting	Ép kiểu ngầm (không tường minh)
Explicit type casting	Ép kiểu tường minh
Variable	Biến
Variable declaration	Khai báo biến
Variable of primitive type	Biến kiểu nguyên thủy
Variable of reference type	Biến kiểu tham chiếu

# CHƯƠNG 1

## LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

---

*Chương này giới thiệu phương pháp lập trình Hướng đối tượng bao gồm tư duy, các khái niệm quan trọng và các tính chất của phương pháp lập trình hướng đối tượng. Nội dung của chương bắt đầu bằng việc giới thiệu sơ lược về lịch sử của ngôn ngữ lập trình và so sánh sự khác nhau giữa phương pháp lập trình hướng đối tượng với phương pháp lập trình cấu trúc nhằm làm rõ ý tưởng và sự cần thiết của phương pháp lập trình này. Sau đó, chúng tôi sẽ trình bày các khái niệm quan trọng trong lập trình bao gồm đối tượng, lớp, thông điệp và truyền thông điệp. Sau cùng, các tính chất của lập trình hướng đối tượng sẽ được giới thiệu gồm tính bao gói, tính kế thừa và tính đa hình. Chương này cung cấp kiến thức cơ sở cho các Chương 3 – 6.*

### 1.1 LỊCH SỬ CỦA NGÔN NGỮ VÀ PHƯƠNG PHÁP LẬP TRÌNH

#### 1.1.1 Ngôn ngữ máy

Máy tính dùng để tính toán, giải quyết các vấn đề trong cuộc sống thông qua các *chương trình máy tính*. Mỗi hệ thống máy tính có một ngôn ngữ riêng gọi là *ngôn ngữ máy*. Đây là một tập các *chỉ thị* mà các *bộ xử lý* của máy tính có thể thực thi được.

Một *chương trình trong ngôn ngữ máy* là một tập các chỉ thị dưới dạng nhị phân, bao gồm một dãy các con số 0 và 1. Những người lập trình ở trong thời kỳ đầu những năm 40s, khi máy tính mới ra đời, phải sử dụng trực tiếp ngôn ngữ máy để viết chương trình cho máy tính trên các *thẻ đục lỗ*.

```
0000 1010 0011 1100 1110 1111 0010 0011 0001
1010 1111 1001 1110 0001 1110 1111 0010 0011
0000 1111 1010 0101 1110 1111 1010 0101 1011
```

Hình 1.1 Ví dụ về một đoạn mã trong trong ngôn ngữ máy

Tuy nhiên, viết chương trình trực tiếp bằng ngôn ngữ máy rất khó khăn và mất nhiều thời gian vì các chỉ thị máy không có tính gợi nhớ. Một chỉ thị trong ngôn ngữ máy thường chỉ thực hiện được những thao tác rất đơn giản và phụ thuộc vào từng hệ thống máy tính. Ngoài ra, việc sử dụng ngôn ngữ máy đòi hỏi người lập trình phải nắm vững kiến trúc của hệ thống máy tính (như thanh ghi, ngắt, tổ chức bộ nhớ,...). Vì vậy, loại ngôn ngữ này không phù hợp để viết các chương trình có xử lý phức tạp, có kích thước lớn (nhiều mã lệnh).

### 1.1.2 Hợp ngữ

Đầu những năm 50s, các ngôn ngữ lập trình cấp cao hơn bắt đầu được sử dụng rộng rãi, được gọi là các *hợp ngữ* (assembler languages). Các chỉ thị hợp ngữ có tính gọi nhớ và một chỉ thị trong các ngôn ngữ này có thể tương ứng với một hoặc một số lệnh trong ngôn ngữ máy. Các ngôn ngữ này làm cho việc viết các chương trình máy tính trở nên dễ dàng hơn, tốn ít thời gian và công sức hơn. Các chương trình viết bằng các ngôn ngữ này không thể được thực thi một cách trực tiếp bởi máy tính vì các bộ xử lý không hiểu được các ngôn ngữ này. Vì vậy, cần có một chương trình để dịch các chương trình viết bằng các ngôn ngữ này sang mã máy, ở dạng mà máy tính có thể “hiểu” được. Các chương trình này được gọi là các *trình biên dịch* (compiler). Hình 1.2 minh họa một chương trình hiển thị chuỗi “Hello, world!” ra màn hình viết bằng hợp ngữ (x86 DOS).

```
.code
org 100h
main proc
    mov     ah, 9
    mov     dx, offset hello_message
    int     21h
    retn
    hello_message db 'Hello, world!$'
main endp
end main
```

Hình 1.2 Một chương trình viết bằng hợp ngữ (hiển thị “Hello, world!”)

### 1.1.3 Ngôn ngữ lập trình cấp cao

Khi phần cứng máy tính ngày càng mạnh hơn, các bài toán ngày càng phức tạp hơn đòi hỏi các ngôn ngữ lập trình phải thay đổi để đáp ứng các nhu cầu này. Các ngôn ngữ phải cho phép thực hiện các thao tác, chỉ thị phức tạp dễ dàng hơn. Ví dụ, các thao tác nhập, xuất phải đơn giản hơn so với chương trình viết bằng hợp ngữ trong Hình 1.2. Ngoài ra, phương pháp mô hình hóa bài toán từ *không gian của vấn đề* (nơi vấn đề tồn tại) sang *không gian giải quyết vấn đề* (chương trình máy tính) cũng đóng vai trò rất quan trọng.

```
10 INPUT "How many stars do you want: ", N
20 S$ = ""
30 FOR I = 1 TO N
40 S$ = S$ + "*"
50 NEXT I
60 PRINT S$
70 INPUT "Do you want more stars? ", A$
80 IF LEN(A$) = 0 THEN GOTO 70
90 A$ = LEFT$(A$, 1)
100 IF A$ = "Y" OR A$ = "y" THEN GOTO 10
110 END
```

Hình 1.3 Chương trình hiển thị các dấu “\*” viết bằng ngôn ngữ Basic

Sau hợp ngữ thì một số ngôn ngữ cấp cao hơn (gần với ngôn ngữ tự nhiên của con người hơn, xa kiến trúc và ngôn ngữ máy tính hơn) cũng được giới thiệu như COBOL, Basic, FOCAL,... Một lệnh trong các ngôn ngữ này có thể tương đương với nhiều lệnh mã máy hoặc hợp ngữ, có thể thực thi nhiều tác vụ phức tạp hơn. Điều này giúp giảm rất nhiều công sức và thời gian viết chương trình. Đây là các ngôn ngữ lập trình phi cấu trúc trong đó chương trình là một khối duy nhất. Hình 1.3 minh họa một chương trình viết bằng ngôn ngữ lập trình Basic dùng để hiển thị các dấu "\*" ra màn hình.

Khi mà máy tính chủ yếu được sử dụng để tính toán những bài toán đơn giản thì một chương trình phi cấu trúc như Basic, COBOL hay hợp ngữ<sup>1</sup> là phù hợp để giải quyết vấn đề. Tuy nhiên, đối với những bài toán tính toán phức tạp thì đòi hỏi các ngôn ngữ phải cung cấp các phương tiện để người lập trình có thể chia một bài toán ra thành nhiều bài toán nhỏ hơn (phương pháp chia để trị), giúp cho việc giải quyết vấn đề đơn giản và rõ ràng hơn. Đây chính là nguyên lý của các *ngôn ngữ lập trình hướng thủ tục* (procedural programming languages) hay các *ngôn ngữ lập trình có cấu trúc* (structural programming languages). Một chương trình viết bằng phương pháp hướng thủ tục được xem như là một tập hợp của các hàm (function/procedure) tương tác lẫn nhau. Một số ngôn ngữ lập trình hướng thủ tục như Pascal, C, Fortran,... Hình 1.4 minh họa một chương trình viết bằng ngôn ngữ lập trình C bao gồm hai hàm, trong đó hàm main() là chương trình chính có gọi đến hàm abs() để tính giá trị tuyệt đối của một biến a.

```
#include <stdio.h>
float abs(float n) {
    if (n < 0)
        return -n;
    return n;
}

int main() {
    int a = -5.5;
    printf("abs(a) = %f", abs(a));
    return 0;
}
```

Hình 1.4 Chương trình tính giá trị tuyệt đối viết bằng ngôn ngữ lập trình C

---

<sup>1</sup> Thực tế thì trong hợp ngữ vẫn cho phép tạo các hàm (function), là một đoạn mã để thực hiện một tác vụ nào đó, có thể được gọi đến nhiều lần trong chương trình. Tuy nhiên, hợp ngữ chưa có khái niệm hàm một cách tường minh như các ngôn ngữ lập trình cấp cao hiện tại.

### 1.1.4 Ngôn ngữ lập trình hướng đối tượng

Phương pháp lập trình hướng thủ tục và các ngôn ngữ lập trình hướng thủ tục tồn tại trong một thời gian dài và hiện tại vẫn còn đang được sử dụng. Tuy nhiên, khi máy tính ngày càng mạnh hơn thì ứng dụng của máy tính ngày càng phong phú và phức tạp hơn. Ngoài việc sử dụng máy tính cho nhu cầu công việc như tính toán, quản lý nhân sự,... thì máy tính còn được sử dụng cho nhu cầu giải trí như nghe nhạc, xem phim, trò chơi,... Điều này đòi hỏi các ngôn ngữ phải cung cấp phương tiện (mô hình) để người lập trình có thể mô hình hóa các vấn đề trong thực tế (nơi mà bài toán tồn tại) vào trong chương trình (nơi mà bài toán sẽ được giải quyết) dễ dàng hơn. Ví dụ, các ngôn ngữ lập trình cấu trúc không có những khái niệm để người lập trình có thể mô hình các nhân vật trong trò chơi như các cầu thủ, các người nông dân,... một cách trực tiếp. Vì vậy, để mô hình các nhân vật này thì đòi hỏi người lập trình phải thực hiện một cách gián tiếp thông qua các khái niệm biến, hàm,... Điều này đòi hỏi người lập trình phải bỏ ra nhiều thời gian và công sức. Ngoài ra, khi các vấn đề ngày càng phức tạp thì yêu cầu về việc sử dụng lại mã cũng là một yêu cầu quan trọng đối với các phương pháp và ngôn ngữ lập trình. Đây chính là tiền đề cho sự ra đời của phương pháp lập trình hướng đối tượng.

Mô hình mà phương pháp lập trình hướng đối tượng lựa chọn để biểu diễn vấn đề là mô hình dựa trên *đối tượng (object)*. Một đối tượng trong phương pháp lập trình này là sự mô hình hóa của một sự vật trong thực tế. Một đối tượng bao gồm *thuộc tính (property/attribute)* và *hành vi (behaviour)* của đối tượng. Thuộc tính thể hiện cho các đặc điểm hay trạng thái của một đối tượng. Còn hành vi của một đối tượng thể hiện thao tác hay hoạt động mà đối tượng đó có thể thực hiện được. Khái niệm đối tượng trong lập trình xuất hiện đầu tiên vào cuối những năm 50s, đầu những năm 60s. Ngôn ngữ lập trình đầu tiên có hỗ trợ phương pháp này là ngôn ngữ Simula 67. Đến đầu những năm 70s, Alan Kay và các cộng sự của ông đã giới thiệu ngôn ngữ lập trình Smalltalk, được coi là ngôn ngữ lập trình hướng đối tượng đầu tiên. Hiện nay, hầu như tất cả các ngôn ngữ lập trình đều hỗ trợ phương pháp hướng đối tượng như Java, C++, C#, PHP, Python.

Ngoài mô hình hướng đối tượng thì còn một số mô hình lập trình khác như mô hình dựa trên danh sách (list) của ngôn ngữ LISP hoặc dựa trên vị từ (predicate) của ngôn ngữ Prolog,... Tuy nhiên, các mô hình này chỉ phù hợp cho một miền ứng dụng (application domain) cụ thể nên không được sử dụng như một mô hình lập trình “đa dụng” (multi-purpose programming model).

Sự phát triển của ngôn ngữ lập trình có thể được xem như là một quá trình *trừu tượng hóa (abstraction)*. Đây là một quá trình loại bỏ hay làm mờ đi những gì không thuộc bản chất của các sự vật để làm nổi bật lên những cái thể hiện cho bản chất của sự vật *trong ngữ cảnh (context) trong miền ứng dụng*. Điều này cho phép người lập trình tập trung vào giải quyết bài toán, thay vì mất thời gian, công sức để mô hình hóa bài toán vào trong mô hình của ngôn ngữ