

스스로 하는  
프로그래밍 공부

# 코딩

# 자율학습



구독자 40만  
조회수 2,600만

# 나도코딩! 의

# 자바 입문

기초부터 객체지향까지,  
예제와 실습으로 완성하는  
자바 프로그래밍

나도코딩 지음

100개 예제와  
문제로 익히는  
실습 중심 학습

무료 동영상  
강의 제공



코딩  
자율학습단과  
함께 공부하기

길벗

# 코딩 자율학습 나도코딩의 자바 입문

Java for Beginners with Nadocoding

초판 발행 · 2026년 5월 4일

지은이 · 나도코딩

발행인 · 이종원

발행처 · (주)도서출판 길벗

출판사 등록일 · 1990년 12월 24일

주소 · 서울시 마포구 월드컵로 10길 56(서교동)

대표 전화 · 02)332-0931 | 팩스 · 02)323-0586

홈페이지 · [www.gilbut.co.kr](http://www.gilbut.co.kr) | 이메일 · [gilbut@gilbut.co.kr](mailto:gilbut@gilbut.co.kr)

기획 및 책임편집 · 정지연([stopy@gilbut.co.kr](mailto:stopy@gilbut.co.kr)) | 제작 · 이준호, 손일순

마케팅 · 임태호, 전선하, 박민영, 서현정, 박성웅 | 유통혁신 · 한준희 | 영업관리 · 김명자 | 독자지원 · 윤정아

교정교열 · 이미연 | 디자인 및 전산편집 · 책돼지 | 출력 및 인쇄 · 예림인쇄 | 제본 · 예림제본

- ▶ 이 책은 저작권법의 보호를 받는 저작물로 이 책에 실린 모든 내용, 디자인, 이미지, 편집 구성은 허락 없이 복제하거나 다른 매체에 옮겨 실을 수 없습니다.
- ▶ 인공지능(AI) 기술 또는 시스템을 훈련하기 위해 이 책의 전체 내용은 물론 일부 문장도 사용하는 것을 금지합니다.
- ▶ 잘못 만든 책은 구입한 서점에서 바꿔 드립니다.

ISBN 979-11-407-1868-9 93000

(길벗 도서번호 080475)

정가 29,000원

---

독자의 1초를 아껴주는 정성 길벗출판사

(주)도서출판 길벗 | IT단행본&교재, 성인어학, 교과서, 수험서, 경제경영, 교양, 자녀교육, 취미실용

[www.gilbut.co.kr](http://www.gilbut.co.kr)

길벗스쿨 | 국어학습, 수학학습, 주니어어학, 어린이단행본, 학습단행본

[www.gilbutschool.co.kr](http://www.gilbutschool.co.kr)

페이스북 · <https://www.facebook.com/gbitbook>

코딩 자율학습단 · <https://cafe.naver.com/gilbutitbook>

예제 파일과 해설 노트 · <https://github.com/gilbutITbook/080475>

동영상 강의 · <https://youtu.be//watch?v=NQq0dOoEPUM>

## 1장 Hello, 자바! 027



- 1.1 개발 환경 설정하기 ..... 028
  - 1.1.1 윈도우에 인텔리제이 설치하기 029
  - 1.1.2 맥OS에 인텔리제이 설치하기 031
- 1.2 첫 번째 자바 프로그램 만들기 ..... 034
  - 1.2.1 자바 프로젝트 생성하기 034
  - 1.2.2 자바 프로그램 실행하기 037

## 2장 자료형과 변수 039



- 2.1 실습용 패키지 만들기 ..... 040
  - 2.1.1 패키지와 파일 생성하기 040
  - 2.1.2 첫 프로그래밍, Hello World 043
  - 2.1.3 코드 수정하고 다시 실행하기 045
- 2.2 데이터의 종류를 나타내는 자료형 ..... 046
  - 2.2.1 숫자 데이터: 정수, 실수 047
  - 2.2.2 논리 데이터: true, false 048
  - 2.2.3 숫자 데이터로 계산하기 050
- 2.3 값을 저장하는 변수 ..... 052
  - 2.3.1 변수 만들기 052
  - 2.3.2 변수는 변하는 수 055
- 2.4 변수 이름 짓는 법 ..... 059
  - 2.4.1 변수 이름 짓기 규칙 059
  - 2.4.2 밑줄과 숫자 포함하기 061
  - 2.4.3 때로는 간단한 이름으로 062

<b>2.5 특별한 숫자 변수</b> .....	064
<b>2.5.1</b> 메모리를 절약하는 알뜰한 실수: float	064
<b>2.5.2</b> int의 한계를 넘어서는 거대한 정수: long	065
<b>2.6 값을 바꿀 수 없는 상수</b> .....	067
<b>2.6.1</b> 변수를 상수로 만드는 마법: final	067
<b>2.6.2</b> 상수를 만들고 사용하는 규칙	069
<b>2.7 코드에 남기는 메모, 주석</b> .....	071
<b>2.7.1</b> 한 줄 주석	071
<b>2.7.2</b> 여러 줄 주석	072
<b>2.7.3</b> 코드에 설명 더하기	074
<b>2.8 자료형을 바꾸는 형 변환</b> .....	076
<b>2.8.1</b> 숫자끼리 자료형 바꾸기	076
<b>2.8.2</b> 숫자와 문자열 사이 자료형 바꾸기	080
<b>1분 퀴즈</b>	083
<b>실습 문제</b>	084
<b>마무리</b>	085
<b>셀프체크</b>	087

## 3장 연산자 089



<b>3.1 사칙연산의 기본, 산술 연산자</b> .....	090
<b>3.1.1</b> 정수 나눗셈의 비밀	091
<b>3.1.2</b> 나머지 연산자	092
<b>3.1.3</b> 연산자의 우선순위	093
<b>3.1.4</b> 변수를 사용해 연산하기	094
<b>3.1.5</b> 증감 연산자로 1씩 더하고 빼기	095
<b>3.1.6</b> 실습 예제: 은행 대기 번호표 뽑기	099

<b>3.2 값을 저장하고 갱신하는 대입 연산자</b> .....	101
<b>3.2.1</b> 변수 값 갱신하기	101
<b>3.2.2</b> 복합 대입 연산자로 코드를 더 짧게 만들기	102
<b>3.3 참과 거짓을 가리는 비교 연산자</b> .....	105
<b>3.3.1</b> 크고 작음 비교하기	105
<b>3.3.2</b> 같은지 다른지 비교하기	106
<b>3.4 여러 조건을 한 번에 판단하는 논리 연산자</b> .....	109
<b>3.4.1</b> 또는 조건의 OR 연산자	109
<b>3.4.2</b> 그리고 조건의 AND 연산자	111
<b>3.4.3</b> 결과를 뒤집는 NOT 연산자	112
<b>3.5 한 줄로 끝내는 조건문, 삼항 연산자</b> .....	114
<b>3.5.1</b> 삼항 연산자란	114
<b>3.5.2</b> 두 수 중 더 큰 값과 더 작은 값 찾기	115
<b>3.5.3</b> 숫자뿐만 아니라 무엇이든 가능	116
<b>1분 퀴즈</b> 118	
<b>실습 문제</b> 119	
<b>마무리</b> 120	
<b>셀프체크</b> 122	

## 4장 문자열 123



<b>4.1 문자열 가지고 놀기</b> .....	124
<b>4.1.1</b> 문자열 길이 재기: length()	124
<b>4.1.2</b> 대소문자 마음대로 바꾸기: toUpperCase()와 toLowerCase()	126
<b>4.1.3</b> 특정 문자열 포함 여부 확인: contains()	127
<b>4.1.4</b> 단어의 위치 찾기: indexOf(), lastIndexOf()	127
<b>4.1.5</b> 특정 문자열로 시작하는지 확인: startsWith()	129
<b>4.1.6</b> 특정 문자열로 끝나는지 확인: endsWith()	130

<b>4.2 문자열 자르고 바꾸고 붙이기</b> .....	131
<b>4.2.1</b> 원하는 글자로 바꾸기: <code>replace()</code>	131
<b>4.2.2</b> 필요한 부분만 잘라내기: <code>substring()</code>	132
<b>4.2.3</b> 양 끝의 불필요한 공백 없애기: <code>trim()</code>	134
<b>4.2.4</b> 문자열 이어 붙이기: + 연산자, <code>concat()</code>	135
<b>4.3 문자열 비교하기</b> .....	137
<b>4.3.1</b> 문자열이 같은지 확인하기: <code>equals()</code>	137
<b>4.3.2</b> 대소문자 무시하고 비교하기: <code>equalsIgnoreCase()</code>	139
<b>4.3.3</b> <code>equals()</code> 와 <code>==</code> 의 차이	140
<b>4.4 이스케이프 문자로 문자열에 의미 부여하기</b> .....	143
<b>4.4.1</b> 여러 줄 출력하기: <code>\n</code>	143
<b>4.4.2</b> 줄 맞춰 출력하기: <code>\t</code>	144
<b>4.4.3</b> 문법 기호를 문자 그대로 출력하기	145
<b>1분 퀴즈</b>	148
<b>실습 문제</b>	149
<b>마무리</b>	150
<b>셀프체크</b>	152

## 5장 조건문 153



<b>5.1 상황에 따라 흐름을 바꾸는 if 문</b> .....	154
<b>5.1.1</b> if 문의 기본 구조	154
<b>5.1.2</b> 중괄호 사용법	156
<b>5.1.3</b> 여러 조건을 한 번에 검사하기	158
<b>5.2 조건에 따라 결과가 달라질 때는 if-else</b> .....	160
<b>5.2.1</b> 선택지가 두 개일 때	160
<b>5.2.2</b> 조금 더 복잡한 조건 만들기	162
<b>5.3 또 다른 조건이 있을 때는 else if</b> .....	164
<b>5.3.1</b> 여러 선택지 중 하나만 고르기	164
<b>5.3.2</b> 조건 순서의 중요성	165

5.3.3	else if 확장하기	167
5.3.4	else 선택적으로 사용하기	168
<b>5.4</b>	<b>똑똑한 조건문, switch-case 문</b>	170
5.4.1	if 문으로 조건에 따라 다른 결과 출력하기	170
5.4.2	switch-case 문으로 값에 따라 분기하기	172
5.4.3	같은 결과를 처리하는 case 묶기	174
5.4.4	break 생략하기	175
5.4.5	if 문과 switch-case 문의 선택 기준	176
<b>1분 퀴즈</b>	<b>실습 문제</b>	<b>마무리</b>
177	179	181
		<b>셀프체크</b>
		183

## 6장 반복문 185



<b>6.1</b>	<b>정해진 횟수만큼 같은 일을 반복하는 for 문</b>	186
6.1.1	for 문의 기본 구조	186
6.1.2	for 문 사용하기	189
6.1.3	실습 예제: 1부터 10까지 합계 구하기	190
<b>6.2</b>	<b>조건이 참인 동안 같은 일을 반복하는 while 문</b>	192
6.2.1	while 문의 기본 구조	192
6.2.2	무한 루프에 빠지지 않기	194
<b>6.3</b>	<b>일단 한 번 실행해보고 판단하는 do-while 문</b>	196
6.3.1	판단을 먼저 하는 while 문	196
6.3.2	while 문의 한계	197
6.3.3	do-while 문의 기본 구조	198
<b>6.4</b>	<b>반복문 속에 반복문, 중첩 반복문</b>	201
6.4.1	반듯한 사각형 그리기	201
6.4.2	왼쪽으로 기울어진 삼각형 그리기	203
6.4.3	오른쪽으로 기울어진 삼각형 그리기	205

<b>6.5 실습 예제: 구구단 출력하기</b> .....	208
<b>6.5.1</b> 간단한 반복문 맛보기 208	
<b>6.5.2</b> 중첩 반복문으로 구구단 완성하기 209	
<b>6.6 반복문을 탈출할 때는 break</b> .....	211
<b>6.6.1</b> for 문에서 break 사용하기 211	
<b>6.6.2</b> while 문에서 break 사용하기 212	
<b>6.7 반복문 다음 차례로 넘어갈 때는 continue</b> .....	214
<b>6.7.1</b> for 문에서 continue 사용하기 214	
<b>6.7.2</b> while 문에서 continue 사용하기 216	
<b>1분 퀴즈</b> 218 <b>실습 문제</b> 219 <b>마무리</b> 221 <b>셀프체크</b> 223	

## 7장 배열 225



<b>7.1 여러 데이터를 한 번에 다룰 때</b> .....	226
<b>7.1.1</b> 배열이 필요한 이유 226	
<b>7.1.2</b> 배열 선언과 초기화 227	
<b>7.1.3</b> 배열 값 사용하기 229	
<b>7.1.4</b> 다른 자료형의 배열 사용하기 230	
<b>7.2 반복문으로 배열 다루기</b> .....	232
<b>7.2.1</b> for 문으로 배열 순회하기 232	
<b>7.2.2</b> 배열 길이만큼 반복하기 233	
<b>7.2.3</b> 향상된 for 문 사용하기 235	
<b>7.2.4</b> 반복문 선택 기준 236	
<b>7.3 여러 줄의 데이터를 담은 다차원 배열</b> .....	238
<b>7.3.1</b> 2차원 배열 만들기 238	
<b>7.3.2</b> 2차원 배열의 값에 접근하기 240	
<b>7.3.3</b> 줄마다 길이가 다른 2차원 배열 241	

<b>7.4 반복문으로 2차원 배열 순회하기</b> .....	244
<b>7.4.1</b> 2차원 배열의 모든 칸 방문하기	244
<b>7.4.2</b> 줄마다 길이가 다른 2차원 배열 순회하기	245
<b>7.4.3</b> 빈 2차원 배열에 데이터 채우기	247
<b>7.4.4</b> 실습 예제: 예매 좌석 처리하기	248
<b>7.5 컴퓨터가 문자를 이해하는 방식, 아스키 코드</b> .....	250
<b>7.5.1</b> char 변수로 아스키 코드 출력하기	250
<b>7.5.2</b> char 변수로 숫자 연산하기	251
<b>7.5.3</b> 실습 예제: 아스키 코드로 영화관 좌석표 만들기	252
<b>1분 퀴즈</b> 255 <b>실습 문제</b> 256 <b>마무리</b> 257 <b>셀프체크</b> 259	

## 8장 메서드 261



<b>8.1 하나의 일을 하는 코드 묶음, 메서드</b> .....	262
<b>8.1.1</b> 메서드란 262	
<b>8.1.2</b> 첫 메서드 만들기: 정의와 호출 263	
<b>8.1.3</b> 한번 만들어 계속 사용하기 265	
<b>8.2 메서드에 생명을 불어넣는 전달값</b> .....	267
<b>8.2.1</b> 메서드에 값 하나 전달하기 267	
<b>8.2.2</b> 메서드에 여러 값 전달하기 269	
<b>8.3 값을 반환하는 메서드</b> .....	272
<b>8.3.1</b> 반환값이 있는 메서드 만들기 272	
<b>8.3.2</b> 반환값 사용하기 273	
<b>8.3.3</b> 더 간단하게 반환하고 사용하기 274	
<b>8.4 전달값과 반환값이 있는 메서드</b> .....	277
<b>8.4.1</b> 숫자를 받아 제곱을 반환하기 277	
<b>8.4.2</b> 숫자 2개로 거듭제곱하기 279	

<b>8.5 같은 이름으로 다양한 작업을 하는 메서드 오버로딩</b> .....	281
<b>8.5.1</b> 같은 이름의 메서드 여러 개 정의하기 281	
<b>8.5.2</b> 메서드 오버로딩 사용 규칙 284	
<b>8.6 메서드가 필요한 이유</b> .....	286
<b>8.6.1</b> 같은 계산을 반복해 작성한 코드 286	
<b>8.6.2</b> 메서드 하나로 해결하기 287	
<b>8.7 변수의 사용 범위</b> .....	290
<b>8.7.1</b> 지역 변수의 사용 범위 290	
<b>8.7.2</b> 중괄호가 만드는 새로운 스코프 291	
<b>8.7.3</b> 안쪽 스코프와 바깥쪽 스코프의 관계 292	
<b>8.8 프로그램의 시작점, main 메서드</b> .....	294
<b>8.8.1</b> main 메서드의 역할 294	
<b>8.8.2</b> 프로그램을 실행할 때 값 전달하기 295	
<b>8.8.3</b> 실습: 전달값으로 메뉴판 만들기 298	
<b>1분 퀴즈</b> 301 <b>실습 문제</b> 302 <b>마무리</b> 304 <b>셀프체크</b> 306	

## 9장 클래스 전반전 307



<b>9.1 나만의 자료형을 만드는 클래스</b> .....	308
<b>9.1.1</b> 객체 지향 프로그래밍이란 308	
<b>9.1.2</b> 클래스가 없을 때 생기는 문제 309	
<b>9.1.3</b> 클래스로 문제 해결하기 310	
<b>9.2 객체의 속성을 담은 인스턴스 변수</b> .....	313
<b>9.2.1</b> 재료 준비하기: 인스턴스 변수 정의 313	
<b>9.2.2</b> 설계도로 제품 만들기: 객체 생성과 사용 314	
<b>9.2.3</b> 같은 설계도로 다른 제품 만들기 315	

---

<b>9.3 모두에게 공유되는 클래스 변수</b> .....	318
<b>9.3.1</b> 각자의 변수, 인스턴스 변수 318	
<b>9.3.2</b> 모두의 변수, 클래스 변수 319	
<b>9.3.3</b> 클래스 변수 사용하기 320	
<b>9.4 클래스에 동작을 추가하는 메서드</b> .....	323
<b>9.4.1</b> 전달값도 반환값도 없는 메서드 323	
<b>9.4.2</b> 값을 전달받는 메서드 325	
<b>9.4.3</b> 값을 반환하는 메서드 326	
<b>9.5 하나의 이름으로 다양한 동작을 만드는 메서드 오버로딩</b> .....	328
<b>9.5.1</b> 매개변수를 활용해 메서드 동작 제어하기 328	
<b>9.5.2</b> 메서드 오버로딩으로 기본 설정 메서드 만들기 330	
<b>9.6 객체 없이도 바로 사용하는 클래스 메서드</b> .....	332
<b>9.6.1</b> 모두가 함께 쓰는 클래스 메서드 만들기 332	
<b>9.6.2</b> 객체 생성 없이 메서드 바로 호출하기 333	
<b>9.6.3</b> 클래스 메서드를 사용할 때 알아야 할 규칙 334	
<b>9.6.4</b> 클래스 메서드를 사용하는 기준 335	
<b>9.7 나 자신을 가리키는 this</b> .....	337
<b>9.7.1</b> 인스턴스 변수와 매개변수의 이름 충돌 문제 337	
<b>9.7.2</b> this로 인스턴스 변수 명확히 가리키기 339	
<b>9.8 객체를 초기화하는 생성자</b> .....	341
<b>9.8.1</b> 생성자와의 첫 만남: 기본 생성자 341	
<b>9.8.2</b> 객체를 맞춤 설정하기: 사용자 정의 생성자 343	
<b>9.8.3</b> 생성자 간 역할 분리하기: this() 344	
<b>9.9 객체의 데이터를 보호하는 Getter와 Setter</b> .....	347
<b>9.9.1</b> 데이터에 직접 접근할 때의 문제점 347	
<b>9.9.2</b> 게터와 세터로 데이터 보호하기 348	

<b>9.10 외부 접근을 제어하는 접근 제어자</b> .....	352
9.10.1 접근 제어자란	352
9.10.2 변수에 직접 접근할 때의 문제점	353
9.10.3 private으로 철벽 방어하기	354
9.10.4 다른 패키지에서 접근하기	355
9.10.5 메서드에 적용하는 접근 제어자	357
9.10.6 클래스에 적용하는 접근 제어자	358
9.10.7 캡슐화와 정보 은닉의 필요성	359
<b>9.11 클래스를 정리하는 패키지</b> .....	361
9.11.1 패키지와의 import의 역할	361
9.11.2 난수를 생성하는 Random 클래스	363
9.11.3 다양한 자료형의 난수 생성하기	364
9.11.4 실습: 로또 번호 뽑기	366
9.11.5 자바에서 제공하는 유용한 클래스	366
<b>1분 퀴즈</b>	368
<b>실습 문제</b>	369
<b>마무리</b>	371
<b>셀프체크</b>	373

## 10장 클래스 후반전 375



<b>10.1 공통 기능을 물려주는 상속</b> .....	376
10.1.1 같은 코드가 반복되는 문제	376
10.1.2 부모와 자식 관계로 중복 코드 줄이기	380
10.1.3 상속 동작 확인하기	382
10.1.4 상속으로 코드 한 번에 수정하기	383
<b>10.2 상속받은 동작을 바꾸는 메서드 오버라이딩</b> .....	385
10.2.1 상속받은 메서드를 그대로 사용할 때의 문제	385
10.2.2 메서드 오버라이딩으로 동작 재정의하기	387
10.2.3 실수를 줄여주는 @Override 애너테이션	388

---

<b>10.3 부모 타입을 유연하게 다루는 다형성</b> .....	391
<b>10.3.1 IS-A 관계로 이해하는 다형성</b> 391	
<b>10.3.2 부모 클래스의 이름으로 객체 생성하기</b> 392	
<b>10.3.3 다형성으로 코드 구조 단순화하기</b> 393	
<b>10.3.4 자식 클래스의 고유 기능 다시 사용하기</b> 394	
<b>10.4 부모를 가리키는 키워드, super</b> .....	398
<b>10.4.1 super로 부모 메서드 호출하기</b> 398	
<b>10.4.2 부모 클래스의 생성자 호출하기: super()</b> 401	
<b>10.5 값과 주소로 구분하는 참조</b> .....	404
<b>10.5.1 값 자체를 저장하는 기본 자료형과 주소를 저장하는 참조 자료형</b> 404	
<b>10.5.2 값이 그대로 복사되는 기본 자료형</b> 405	
<b>10.5.3 주소가 복사되는 참조 자료형</b> 406	
<b>10.5.4 메서드 매개변수로 전달되는 참조</b> 407	
<b>10.5.5 연결을 끊는 값, null과 NullPointerException</b> 408	
<b>10.6 더 이상 바꿀 수 없는 제한, final</b> .....	410
<b>10.6.1 절대 바꿀 수 없는 값, final 변수</b> 410	
<b>10.6.2 바뀌지 않는 동작을 만드는 final 메서드</b> 413	
<b>10.6.3 더 이상 확장할 수 없는 final 클래스</b> 415	
<b>10.7 정해진 값만 사용하는 열거형</b> .....	417
<b>10.7.1 열거형 만들기</b> 417	
<b>10.7.2 열거형 변수 사용하기</b> 418	
<b>10.7.3 열거형과 함께 사용하는 switch-case 문</b> 419	
<b>10.7.4 열거형을 더 똑똑하게 활용하기</b> 419	
<b>10.7.5 열거형에 나만의 값 부여하기</b> 421	
<b>1분 퀴즈</b> 423	
<b>실습 문제</b> 424	
<b>마무리</b> 426	
<b>셀프체크</b> 428	

## 11장 추상 클래스와 인터페이스 429



<b>11.1 추상 클래스</b> .....	430
<b>11.1.1</b> 아직 완성되지 않은 설계도: 추상 클래스 430	
<b>11.1.2</b> 구현을 자식 클래스에 맡기기: 추상 메서드 432	
<b>11.1.3</b> 추상 클래스 완성하기: 상속과 오버라이딩 432	
<b>11.1.4</b> 완성된 클래스로 객체 만들기 434	
<b>11.2 인터페이스</b> .....	436
<b>11.2.1</b> 인터페이스가 필요한 이유 436	
<b>11.2.2</b> 기능의 뼈대를 정의하는 인터페이스 437	
<b>11.2.3</b> 인터페이스 구현하기: 실제 기능을 가진 부품 만들기 439	
<b>11.2.4</b> 인터페이스 적용하기: 카메라에 기능 장착하기 442	
<b>11.2.5</b> 구현 교체로 성능 확장하기 443	
<b>1분 퀴즈</b> 445 <b>실습 문제</b> 446 <b>마무리</b> 449 <b>셀프체크</b> 450	

## 12장 컬렉션 프레임워크 453



<b>12.1 List 인터페이스</b> .....	454
<b>12.1.1</b> 크기가 변하는 ArrayList 454	
<b>12.1.2</b> 연결로 관리하는 LinkedList 463	
<b>12.2 Set 인터페이스</b> .....	472
<b>12.2.1</b> 중복을 허용하지 않는 HashSet 472	
<b>12.2.2</b> 순서가 필요할 때는 LinkedHashSet 476	
<b>12.3 Map 인터페이스</b> .....	478
<b>12.3.1</b> 키-값 쌍으로 데이터를 관리하는 HashMap 478	
<b>12.3.2</b> 맵에 담긴 데이터 한눈에 살펴보기 483	
<b>12.3.3</b> 키가 중복되면 값이 수정되는 HashMap 485	

<b>12.4 컬렉션을 순회하는 Iterator</b> .....	487
<b>12.4.1</b> 데이터 하나씩 꺼내며 순회하기	487
<b>12.4.2</b> 반복문으로 모든 요소 순회하기	489
<b>12.4.3</b> 데이터 삭제하기: remove()	490
<b>12.4.4</b> Set과 Map에서 Iterator 활용하기	492
<b>1분 퀴즈</b>	494
<b>실습 문제</b>	496
<b>마무리</b>	498
<b>셀프체크</b>	500

## 13장 예외 처리 501

<b>13.1 예외 처리로 프로그램을 안전하게 만들기</b> .....	502
<b>13.1.1</b> 프로그램 오류의 종류	502
<b>13.1.2</b> 프로그램의 구급상자, try-catch	504
<b>13.1.3</b> 다양한 예외 상황 처리하기	506
<b>13.2 catch 블록으로 예외 구분하기</b> .....	509
<b>13.2.1</b> 다중 catch 블록의 작동 원리	509
<b>13.2.2</b> 예외 상황에 맞춰 catch 블록 추가하기	511
<b>13.2.3</b> 여러 예외를 하나의 catch 블록으로 묶어 처리하기	513
<b>13.3 예외를 직접 발생시켜 흐름 제어하기</b> .....	516
<b>13.3.1</b> 사용자 규칙 위반을 if 문으로 처리하기	516
<b>13.3.2</b> throw로 직접 예외 던지기	517
<b>13.4 finally로 무조건 실행하기</b> .....	519
<b>13.4.1</b> 항상 마지막에 실행되는 finally	519
<b>13.4.2</b> 예외가 없어도 실행되는 finally	520
<b>13.4.3</b> try-catch 문의 다양한 조합	521
<b>13.5 Try with resources로 자원 자동 정리하기</b> .....	524
<b>13.5.1</b> 뒷정리가 번거로운 finally의 한계	524
<b>13.5.2</b> try-with-resources 사용하기	526
<b>13.5.3</b> AutoCloseable 인터페이스	527

<b>13.6 사용자 정의 예외로 규칙을 명확히 표현하기</b> .....	529
<b>13.6.1</b> 사용자 정의 예외가 필요한 이유	529
<b>13.6.2</b> 나만의 예외 클래스 만들기	530
<b>13.6.3</b> 직접 만든 예외 사용하기	530
<b>13.7 예외 처리를 호출자에게 미루기</b> .....	532
<b>13.7.1</b> try-catch 문으로 예외 직접 처리하기	532
<b>13.7.2</b> throws로 예외 처리 미루기	533
<b>13.7.3</b> try-catch와 throws 비교	534
<b>1분 퀴즈</b>	536
<b>실습 문제</b>	537
<b>마무리</b>	539
<b>셀프체크</b>	541

## 14장 입출력과 파일 543



<b>14.1 사용자 입력을 받아 프로그램과 대화하기</b> .....	544
<b>14.1.1</b> 입력을 위한 도구: Scanner	544
<b>14.1.2</b> 원하는 값 입력받기: next(), nextInt(), nextDouble()	545
<b>14.1.3</b> 단어와 문장 입력받기: next(), nextLine()	548
<b>14.2 출력 형식을 지정해 결과를 보기 좋게 만들기</b> .....	551
<b>14.2.1</b> 서식 지정자로 출력 형식 지정하기	551
<b>14.2.2</b> 형식 지정해 숫자 출력하기	552
<b>14.2.3</b> 형식 지정해 문자열 출력하기	555
<b>14.3 실습: 학생 성적표 만들기</b> .....	557
<b>14.3.1</b> 문자열 연결해 출력하기	557
<b>14.3.2</b> printf()로 서식 정렬하기	558
<b>14.4 파일 다루기</b> .....	560
<b>14.4.1</b> File 클래스의 역할과 사용 목적	560
<b>14.4.2</b> 실제 파일 만들기: createNewFile()	561
<b>14.4.3</b> 파일 정보 확인하기: 이름, 경로, 크기	561

---

<b>14.5 폴더 다루기</b> .....	564
<b>14.5.1</b> 폴더 생성하기: File 클래스, mkdir() 564	
<b>14.5.2</b> 여러 하위 폴더 한 번에 만들기: mkdirs() 565	
<b>14.6 파일과 폴더 조회하기</b> .....	568
<b>14.6.1</b> 경로 지정하기: 상대 경로와 절대 경로 568	
<b>14.6.2</b> 폴더 내 목록 가져오기: listFiles() 569	
<b>14.6.3</b> 파일과 폴더 구분하기: isFile(), isDirectory() 570	
<b>14.7 파일 읽고 쓰기</b> .....	572
<b>14.7.1</b> 새로운 파일 만들고 내용 쓰기 572	
<b>14.7.2</b> 기존 파일에 내용 이어 쓰기 573	
<b>14.7.3</b> 파일 내용 읽어오기 574	
<b>14.8 파일 삭제하기</b> .....	576
<b>14.8.1</b> 파일 삭제하기: delete() 576	
<b>14.8.2</b> 안전하게 파일 삭제하기: exists() 577	
<b>14.8.3</b> 폴더 삭제하기 577	
<b>14.8.4</b> 하위 내용을 포함한 폴더 삭제하기 578	

**1분 퀴즈** 581      **실습 문제** 582      **마무리** 584      **셀프체크** 586

**INDEX** 588



**어렵게만 느껴졌던 자바,  
이 책 하나로 막힘없이 이해할 수 있다!**

### '왜'까지 이해하도록 친절하게 설명합니다

문법과 기능을 단순히 나열하기보다 필요한 상황과 예시를 들어 친절하게 설명합니다. 단순히 사용 방법을 익히는 데서 더 나아가 왜 그렇게 사용하는지 이해할 수 있으며, 입문자도 흐름을 파악하며 코드를 작성할 수 있습니다.

### 기초부터 객체지향까지 단계적으로 학습합니다

자바 기초 문법에서 시작해 객체지향까지 자연스럽게 이어지도록 구성했습니다. 각 내용이 순서대로 연결되어 있어 처음 배우는 사람도 부담 없이 따라 할 수 있고, 이미 학습한 경험이 있는 독자도 개념을 다시 정리하는 데 도움이 됩니다.

### 직접 만들며 익힐 수 있도록 실습 중심으로 구성했습니다

개념을 설명한 뒤 바로 예제 코드를 통해 확인하고, 1분 퀴즈와 실습 문제, 셀프체크로 반복 연습합니다. 직접 실행해 보고 결과를 확인하는 과정에서 개념을 자연스럽게 이해할 수 있고, 스스로 코드를 작성하는 과정에서 문제를 해결하는 힘도 함께 기를 수 있습니다.

### 이 책을 먼저 본 베타 학습단의 한마디

- 예제 중심 구성 덕분에 초보자도 쉽게 따라 할 수 있습니다. **\_김선예**
- 구성과 설명 방식이 뛰어나 자바를 처음 배우는 사람에게 적합합니다. **\_김동우**
- 1:1로 과외를 받듯이 기본기를 탄탄하게 다질 수 있는 입문서의 정석입니다. **\_손지선**
- 단계별 실습과 촘촘한 퀴즈 덕분에 기초 개념을 정리하는 데 도움이 됩니다. **\_윤경민**
- 큰 장벽인 객체지향을 직관적이고 친절하게 설명해 쉽게 이해할 수 있습니다. **\_심주현/심은섭**
- 자바를 처음 배우는 분은 물론, 개념을 다시 정리하고 싶은 분께도 추천할 만한 책입니다. **\_박찬웅**

난이도

● ○ ○ ○  
입문 초급 중급 고급

코딩 자율학습  
나도코딩의 자바 입문

Java for Beginners  
with Nadocodiong

정가 29,000원



9 791140 718689 93000  
ISBN 979-11-407-1868-9