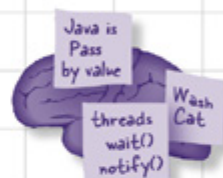


Head First Java

Легкий для сприйняття довідник

Дізнайтеся, як потоки
можуть змінити ваше життя



Грунтовно затямте
концепції Java



Уникайте безглузких
00 помилок

Побавтеся
в бібліотеці Java



Розімніть
свій мозок
за допомогою
42 Java-пазлів



Зробіть привабливими
та корисними
графічні інтерфейси

Кеті Сьерра, Берт Бейтс

Кеті Сьєрра та Берт Бейтс

Head First

Java

Друге видання

Легкий для сприйняття довідник

ВИДАВНИЧИЙ ДІМ
Ф А Б У Л А
#PRO

Кеті Сьєрра, Берт Бейтс
HEAD FIRST JAVA, друге видання

Видавничий дім «Фабула»
2023

Оригінальна назва твору: HEAD FIRST JAVA, SECOND EDITION

Copyright © 2003, 2005 by O'Reilly Media,
Inc. All rights reserved.

© Г. Якубовська, пер. з англ., 2021

© ВД «Фабула», 2023

ISBN 978-617-522-137-2 (epub)

Усі права збережено.

*Жодна частина цієї книжки не може бути відтворена
в будь-якій формі без письмового дозволу власників авторських прав.*

Даний переклад публікується та продається з дозволу O'Reilly Media, Inc,
які володіють або контролюють всі права на публікацію та продаж того ж самого перекладу.

Електронна версія створена за виданням:

Сьєрра Кеті

C96 Head First Java, друге видання / Кеті Сьєрра, Берт Бейтс / Пер. з англ. Г. Якубовська. —
Харків : ВД «Фабула», 2022. — 720 с.

ISBN 978-617-522-033-7

Технологія Java сьогодні використовується скрізь. Якщо ви розробляєте програмне забезпечення і досі не опанували Java, точно настав час заглибитися у цю книжку.

Head First Java — це вичерпний навчальний посібник, який надасть Вам досвід роботи з Java та об'єктно-орієнтованого програмування. Ця книжка допоможе вам вивчити мову Java унікальним методом, що виходить за рамки синтаксису та інструкцій з експлуатації і допомагає зрозуміти, як стати неперевершеним розробником. Ви опануєте основи мови, отримаєте загальну інформацію, засвоїте роботу з потоками, мережевою взаємодією та розподіленим програмуванням, і навіть побудуєте гру «Sink The Dot Com», а також створите барабанний музичний чат.

Окрім захопливого стилю, що одразу поглинає вас, у «Head First Java» міститься величезна кількість практичних речей, які інші посібники залишають поза увагою. Розумно, іронічно, сучасно та практично — не так уже багато існує видань, що здатні похвалитися цими якостями. Обережно — ви можете насправді чогось навчитися!

УДК 004.4:004.02

Шановний читачу!

Спасибі, що придбали цю книгу.

Нагадуємо, що вона є об'єктом Закону України «Про авторське і суміжні право», порушення якого карається за статтею 176 Кримінального кодексу України «Порушення авторського права і суміжних прав» штрафом від ста до чотирьохсот неоподатковуваних мінімумів доходів громадян або виправними роботами на строк до двох років, з конфіскацією та знищенням всіх примірників творів, матеріальних носіїв комп'ютерних програм, баз даних, виконань, фонограм, програм мовлення та обладнання і матеріалів, призначених для їх виготовлення і відтворення. Повторне порушення карається штрафом від тисячі до двох тисяч неоподатковуваних мінімумів доходів громадян або виправними роботами на строк до двох років, або позбавленням волі на той самий строк, з конфіскацією та знищенням всіх примірників, матеріальних носіїв комп'ютерних програм, баз даних, виконань, фонограм, програм мовлення, аудіо- і відеокасет, дискет, інших носіїв інформації, обладнання та матеріалів, призначених для їх виготовлення і відтворення. Кримінальне переслідування також відбувається згідно з відповідними законами країн, де зафіксовано незаконне відтворення (поширення) творів.

Книга містить криптографічний захист, що дозволяє визначити, хто є джерелом незаконного розповсюдження (відтворення) творів.

Щиро сподіваємося, що Ви з повагою поставитесь до інтелектуальної праці інших і ще раз Вам вдячні!



Що кажуть про «Head First Java»



За версією «Amazon», перше видання «Head First Java» (2003) очолило десятку книжок із комп'ютерних наук у номінації «Вибір редакції».



Журнал «Software Development» оголосив «Head First Java» фіналістом чотирнадцятої щорічної нагороди «Jolt Cola / Product Excellence».

«Head First Java» перетворює друковану сторінку на річ найбільш подібну до графічного інтерфейсу користувача який ви коли-небудь бачили. В іронічній, сучасній манері автори роблять вивчення Java захопливим, змушуючи читача постійно ставити собі питання: «Що вони утнуть наступного разу?»

Воррен Койффель, журнал «Software Development»

Єдиний спосіб визначитися з якістю підручника — це з'ясувати, наскільки добре він навчає. «Head First Java» — досконалий викладач. Отже, спочатку я подумав, що це безглуздо... а згодом зрозумів, що з легкістю засвоюю теми, просто переглядаючи книжку.

Стиль «Head First Java» насправді полегшує навчання.

Із відгуків на slashdot.org

Окрім захопливого стилю, що одразу поглинає вас, у «Head First Java» міститься величезна кількість практичних питань, які інші книжки залишають у вигляді нікчемних «вправ для читача». Розумно, іронічно, сучасно та практично — не так уже багато існує підручників, що здатні похвалитися цими якостями і, до того ж, розтлумачити вам принципи серіалізації об'єктів і протоколи розгортання мережі.

Д-р Ден Расселл, директор департаменту «Almaden Research Center», викладач курсу штучного інтелекту в Стенфордському університеті

Стрімко, зухвало, весело та захопливо. Обережно — ви можете насправді чогось навчитися!

Кен Арнольд, співавтор (разом із Джеймсом Гослінгом, творцем Java) «The Java Programming Language»

Технологію Java використовують скрізь. Якщо ви розробляєте програмне забезпечення і досі не вивчили Java, напевне настав час зануритися в *Head First*.

Скотт Макнілі, голова, президент і CEO «Sun Microsystems»

«Head First Java» схожа на комік-гурт «Монті Пайтон», що зустрівся з Бандою чотирьох... Текст так вдало приправлений головоломками і розповідями, вікторинами та прикладами, що засвоєння матеріалу проходить так добре, як ніколи до цього.

Дуглас Поу, «Columbia Java Users Group»

Подяки за «Head First Java»

Прочитайте «*Head First Java*», і ви знову отримаєте задоволення від навчання... Для людей, які люблять вивчати нові мови програмування, але не мають досвіду в галузі інформатики чи програмування, ця книжка є справжньою знахідкою... Це єдина книжка, що робить вивчення складної комп'ютерної мови веселим. Я сподіваюся, що десь існують інші автори, які прагнуть відкинути зашкарублі «традиційні» форми викладення матеріалу. Вивчення комп'ютерних мов має бути веселим, а не обтяжливим.

Джудіт Тейлор, «Southeast Ohio Macromedia User Group»

Якщо ви хочете вивчити Java, то це саме те, що ви шукали: ласкаво просимо до першої технічної книжки на основі графічного інтерфейсу користувача! Цей ідеально виконаний новаторський формат надає переваги, які інші видання, присвячені Java, просто не в змозі запропонувати...

Підготуйтеся до по-справжньому захопливої подорожі країною Java!

Ніл Р. Бауман, капітан та CEO «Geek Cruises»

Якщо ви новачок у програмуванні і вас цікавить Java, ця книжка саме для вас... Охоплюючи різні питання починаючи від об'єктів до створення графічних інтерфейсів користувача (GUI), від обробки винятків (помилки) до мережевої комунікації (сокетів) і багатопотоковості, навіть якщо ви запакуєте купу класів в один інсталяційний файл, це видання досить повне... Я впевнений, що книжка вам сподобається так само, як вона сподобалася мені, і сподіваюся, що серія Head First охопить багато інших тем!

Із відгуків на LinuxQuestions.org

Я став ЗАЛЕЖНИМ від стислих оповідей цієї книжки, анотованого коду, глузливих інтерв'ю та вправ для мозку.

Майкл Юан, автор Enterprise J2ME

«*Head First Java*»... надала нове значення маркетинговій фразі «There's an O'Reilly for that». Я вибрав цю книжку, бо кілька поважних людей описали її, скориставшись виразами на кшталт «революційна» або «радикально інший підхід до посібника». Вони мали рацію... Видавництво «O'Reilly» застосувало науковий і добре обміркований підхід. Результат виявився смішним, специфічним, актуальним, інтерактивним і майже геніальним... Читати цю книжку — наче перебувати в залі конференції, вчитися і водночас сміятися разом із колегами... Якщо ви хочете РОЗУМІТИ Java, придбайте цю книжку.

Ендрю Поллак, www.thenorth.com

Кілька днів тому я отримав свій примірник «*Head First Java*» від Кеті Сьєрра та Берта Бейтса. Я лише трохи погортав книжку, але мене вразило те, що навіть у перший вечір, коли я почувався стомленим, я піймав себе на думці: «Добре, ще сторінка-друга — і тоді вже до ліжка».

Джо Літтон

Head First Java™

Друге видання

Хіба не можна помріяти про те,
аби книжка з Java була більш
стимулюючою, аніж очікування
в черзі до Департаменту авто-
транспорту Каліфорнії, щоб оновити
водійську ліцензію? Це, мабуть,
лише фантазія...



Кеті Сьєрра
Берт Бейтс

O'REILLY®

Пекін • Бостон • Фарнем • Себастопол (Каліфорнія) • Токіо

*Присвячується нашим мізкам, за те, що вони є
(незважаючи на хиткість доказів)*

Творці серії Head First

Кеті Сьєрра



Берт Бейтс



Кеті зацікавилася теорією навчання ще відтоді, коли почала створювати ігри (для Virgin, MGM і Amblin'). Переважна частина формату *Head First* була розроблена нею під час викладання курсу New Media Authoring для програми UCLA Extension's Entertainment Studies. Останнім часом вона готує фахівців для Sun Microsystems, навчає інструкторів Java мистецтву викладання новітніх Java-технологій. Водночас вона брала участь в розробці декількох сертифікаційних іспитів для Java-розробників. Разом із Бертом Бейтсом використовувала концепції *Head First* для навчання сотень викладачів, розробників і пересічних користувачів, які не мають жодного стосунку до програмування. До того ж, Кеті є засновником сайту javaranch.com і блогу «Creating passionate Users».

Окрім цієї книжки, Кеті також є співавтором видань *Head First Servlets*, *Head First EJB* та *Head First Design Patterns*.

У вільний від роботи час Кеті займається верховою їздою на своїй ісландській конячці, катається на лижах і бігає, а також захоплюється швидкістю свігла.

Пишіть їй за адресою kathy@wickedlysmart.com.

Кеті та Берт намагаються не залишати жодного листа без відповіді, але, з огляду на кількість листів і напружений графік відряджень, робити це доволі складно. Найкращий (і найшвидший) спосіб отримати технічну допомогу щодо цієї книжки — звернутися до дуже жвавого форуму для новачків у Java: javaranch.com.

Берт — досвідчений розробник і архітектор програмного забезпечення, але його десятирічні дослідження в сфері штучного інтелекту викликали в нього підвищений інтерес до теорії і технології навчання. Відтоді він займається підвищенням кваліфікації розробників. Останнім часом був у групі розробки декількох сертифікаційних іспитів із мови Java для корпорації «Sun».

У перше десятиліття своєї кар'єри, Берт подорожував по всьому світу, допомагаючи ЗМІ на кшталт «Radio New Zealand», «Weather Channel» та «Arts & Entertainment Network» (A&E). Одним із його улюблених проектів того часу стала побудова повного симулятора залізничної системи для «Union Pacific Railroad».

Берт — безнадійно захоплений грою в Go і давно працює над написанням власної версії гри Go. Непогано грає на гітарі та пробує себе в ролі банджо-музики. Полюбить катання на лижах, біг, а також пробує навчати свого ісландського коня Анді (чи вчитися в нього).

Берт пише книжки у співавторстві з Кеті. Наразі вони працюють над новою серією видань (стежте за оновленнями у блозі).

Шукайте Берта на [javaranch](http://javaranch.com) або на Go-сервері IGS (під псевдонімом *jackStraw*). Ви також можете написати йому листа за адресою terrapin@wickedlysmart.com.

Зміст (коротко)

Вступ	21
1. Відкидаючи зайве. Швидке занурення	33
2. Подорож до Об'єктивіля. Класи й об'єкти	53
3. Знайте свої змінні. Примітиви та посилання	81
4. Як поведуться об'єкти. Методи використовують змінні екземпляра	103
5. Надпотужні методи. Створення програм	127
6. Використання бібліотеки Java. Знайомтеся: Java API	157
7. Покращене життя в Об'єктивілі. Наслідування та поліморфізм	197
8. Серйозний поліморфізм. Інтерфейси та абстрактні класи	229
9. Життя і смерть об'єктів. Конструктори та збирання сміття	267
10. Числа мають значення. Числа і статичні члени класу	305
11. Небезпечна поведінка. Обробка винятків	347
12. Дуже графічна історія. Створення графічного користувацького інтерфейсу (GUI)	385
13. Попрацюйте над Swing. Робота з бібліотекою Swing	431
14. Збереження об'єктів. Серіалізація та I/O (ввід/вивід)	461
15. Встановлюємо зв'язок. Мережева та потоки	503
16. Структури даних. Колекції та шаблони	561
17. Випуск коду. Пакети, jar-архіви та розгортання	613
18. Розподілені обчислення. Віддалене розгортання за допомогою RMI	639
Додаток А. Фінальна «Кухня Коду»	681
Додаток Б. Десять важливих речей, що не потрапили до цієї книжки	709
Покажчик	709

Зміст (докладно)

Вступ

Ваш мозок і Java. Коли ви намагаєтеся щось вивчити, ваш мозок намагається надати вам послугу, переконуючи, що все це не варто уваги. Він думає: «Краще перейматися важливішими речами, наприклад, небезпечними дикими тваринами або тим, що катання голяка на сноуборді — погана ідея». Як же переконати свій мозок у тому, що від знання Java залежить ваше життя?

Для кого ця книжка?	22
Кому швидше за все слід триматися якнайдалі від цієї книги?	22
Ми знаємо, про що ви подумали	23
І ми знаємо, чим переймається ваш мозок	23
Ми вважаємо читача «Head First» учнем	24
Метапізнання: мислення про мислення	25
Ось що зробили МИ	26
Ось що ви можете зробити, аби змусити свій мозок підкорятися	27
Що необхідно для читання цієї книги	28
Деякі останні нюанси, що їх варто знати	29
Технічні редактори	30
Подяки	31

1 Відкидаючи зайве

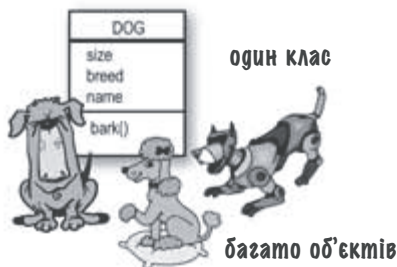
Java відкриває нові МОЖЛИВОСТІ. Від часів скромного релізу відкритої (і доволі вбогої) версії 1.02, ця мова підкорила розробників своїм дружнім синтаксисом, об'єктно-орієнтованими функціями, управлінням пам'яттю і, що найважливіше, перспективою перенесення на різні платформи. Ми швиденько зануримося, напишемо код, скомпілюємо й запустимо його. Маємо на увазі синтаксис, цикли, розгалуження і все, що робить Java такою кльовою. Отже, починаємо.



- Як працює Java34
- Що ви робитимете в Java35
- Дуже стисла історія Java36
- Структура коду в Java39
- Анатомія класу40
- Створення класу з методом main41
- Що можна сказати всередині main методу?42
- Зациклювання, зациклювання і...43
- Умовне розгалуження45
- Створення серйозного бізнес-застосунку46
- Понеділковий ранок Боба47
- Генератор фраз49
- Вправи і головоломки52

2 Подорож до Об'єктвіля

Мені казали, що там будуть об'єкти. У Розділі 1 ми розмістили весь свій код у методі main(), але це не зовсім об'єктно-орієнтований підхід. Тепер ми маємо залишити позаду цей світ процедур і почати розробляти власні типи об'єктів. Ми розглянемо, чим цікаве об'єктно-орієнтоване (ОО) програмування за допомогою мови Java. Також розглянемо різницю між класом та об'єктом і дізнаємося, як об'єкти можуть покращити ваше життя.



- Війна за крісло60
- Що вам подобається в ОО?65
- Проектуючи клас, поміркуйте про об'єкти, які буде створено з цього типу класу66
- Створюємо перший об'єкт68
- Створення і тестування об'єктів Movie69
- Швидше! Забирайтеся з головного методу!70
- Запускаємо гру «Відгадування»72
- Вправи і головоломки74

3

Знайте свої змінні

Змінні бувають двох видів: примітиви і посилання.

У житті має бути щось окрім чисел, рядків і масивів. Як бути з об'єктом PetOwner у якого є змінна типу Dog? Або Car зі змінною Engine? У цьому розділі ми розкриємо завісу таємниці над типами в мові Java й розглянемо, що саме можна оголошувати як змінну, які значення присвоювати їм і що взагалі можна з ними зробити. І нарешті, розглянемо як насправді виглядає життя в кучі (heap), яка контролюється сміттєзбиральним механізмом (garbage-collectible heap).

Тут heap, або «купа», — це спеціальна ділянка пам'яті, яка контролюється автоматично і створена для впорядкування об'єктів.



Посилання на об'єкт Dog

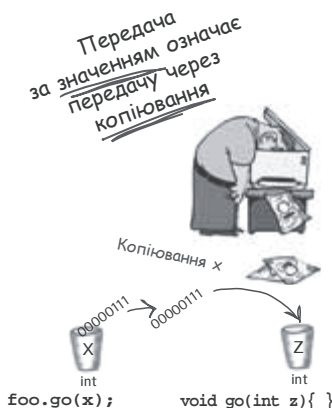
Оголошення змінної.....	82
«Мені, будь ласка, double мокко. Хоча ні, зробіть int»	83
Пильнуйте, щоби нічого не розлити... ..	84
Тримайтеся подалі від цього ключового слова!.....	85
Керуємо об'єктом Dog.....	86
Посилання на об'єкт — це лише ще одне значення змінної.....	87
Масив як підставка для склянок	91
Масиви теж є об'єктами.....	91
Створюємо масив об'єктів Dog.....	92
Контролюйте свій об'єкт Dog (за допомогою посилання).....	93
Приклад класу Dog.....	94
Вправи і головоломки	95

4

Як поведуться об'єкти

Стан впливає на поведінку, а поведінка — на стан.

Нам відомо, що об'єкти характеризуються **станом** і **поведінкою**, що представлені **змінними екземпляра** і **методами**. Зараз ми обговоримо зв'язок між станом і поведінкою. Поведінка об'єкта використовує унікальний стан об'єкта. Інакше кажучи, **методи використовують значення змінних екземпляра**. Скажімо, «якщо собака важить менше 14 фунтів, треба видати звук *yirru*, інакше...» **Давайте змінимо стан!**

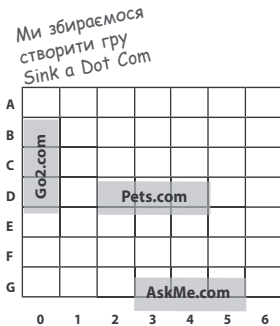


Пам'ятайте: клас описує те, що об'єкт знає і робить	104
Розмір впливає на гучність гавкоту.....	105
Ви можете передавати методу різні значення.....	106
Ви можете отримувати значення як результат виконання методу	107
Ви можете передати методу одразу декілька значень.....	108
У Java все передається за значенням. Це означає, що значення копіюється під час передавання.....	109
Круті штуки які можна зробити з параметрами і типами повернених значень ...	111
Інкапсуляція	112
Інкапсуляція класу GoodDog.....	114
Як поведуться об'єкти всередині масиву?.....	115
Оголошення та ініціалізація змінних екземпляра	116
Різниця між змінними екземпляра та локальними змінними.....	117
Порівнюємо змінні (примітиви або посилання)	118
Вправи і головоломки.....	120

5

Надпотужні методи

Давайте зробимо метод потужнішим. Ви вже побавилися зі змінними, погралися з декількома об'єктами і написали невеличкий код. Але для повноцінної роботи потрібно більше інструментів. На кшталт **операторів і циклів**. Це може бути корисним для **генерації випадкових чисел**. І перетворення об'єкта String на Int — так, це було б круто. Чому б вам не створити щось справжнє, аби на власні очі побачити, як з нуля пишуться (і тестуються) програми? **Можливо, це буде гра** на кшталт Sink a Dot Com (подібна до «Морського бою»).



Давайте створимо аналог «Морського бою» — гру «Sink a Dot Com».....	128
Передусім — високорівневе проектування	129
Плавний вступ у спрощену версію гри	130
Розробка класу	131
Пишемо реалізацію методу.....	133
Тестовий код для класу SimpleDotCom.....	134
Метод checkYourself()	136
Про нові речі.....	137
Псевдокод для класу SimpleDotComGame	140
Методи main(), random() та getUserInput() у гри	142
Останній клас GameHelper.....	144
Зіграємо! Що це? Помилка?.....	145
Детальніше про цикли for.....	146
Приведення примітивів	149
Вправи і головоломки.....	150

6

Використання бібліотеки Java

Разом із Java постачаються сотні готових класів. Можете не витратити час на винахід власного велосипеда, якщо знаєте, як знайти необхідне в бібліотеці Java (її також називають Java API). Гадаю, у вас знайдуться важливіші справи. При написанні коду зосередьтеся на тій частині, що є унікальною для вашого застосунку. Базова бібліотека Java являє собою гігантський набір класів, готових до застосування як будівельні блоки.

«Приємно знати, що ArrayList міститься в пакеті java.util. Але яким чином я змогла б додуматися до цього самостійно?»

Джулія, 31 рік, модель

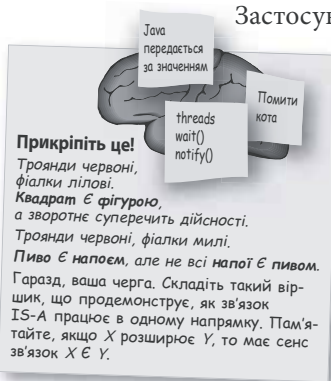


У попередньому розділі ми зупинились на найцікавішому місці. На помилці ...	158
То що ж трапилось?	159
Яким чином це виправити?.....	160
Перший варіант занадто незграбний	161
Другий варіант трохи кращий, але все ще незграбний	161
Прокиньтесь та відчуйте дух бібліотеки.....	164
Деякі приклади застосування ArrayList	165
Порівняння ArrayList зі звичайним масивом	168
Виправляємо код класу DotCom	170
Новий і вдосконалений клас DotCom	171
Давайте створимо справжню гру «Sink a Dot Com»	172
Що треба змінити?.....	173
Хто що робить у гри DotComBust (і коли)	174
Псевдокод для справжнього класу DotComBust	176
Остаточна версія класу DotCom	182
Суперпотужні логічні вирази	183
Використання бібліотеки (Java API).....	186
Як працювати з API.....	190
Вправи і головоломки.....	193

7 Покращене життя в Об'єктивлі

Плануйте свої програми з прицілом на майбутнє. Чи легко написати код так, щоби хтось інший міг би його розширити? Чи зацікавить вас створення гнучкого коду, що не боїться змін у технічному завданні, що виникають в останню мить? Долучившись до планування поліморфізму, ви дізнаєтеся про п'ять кроків вправного проектування класів, три прийоми задля досягнення поліморфізму і вісім способів створення гнучкого коду, а якщо діятимете енергійно — отримаєте бонусний урок із чотирма порадами щодо використання наслідування.

Війни за крісло переглянуто.....	198
Наслідування в деталях.....	200
Давайте розробимо дерево наслідування для програми-симулятора життя тварин.....	202
Застосування наслідування для запобігання дублюванню коду в підкласах.....	203
Чи всі тварини харчуються однаково?.....	204
Шукаємо нові можливості наслідування.....	205
Який метод викликається?.....	207
Проектування дерева наслідування.....	208
Використання відносин Є (IS-A) і МІСТИТЬ (HAS-A).....	209
Але стривайте! Це ще не все!.....	210
Як дізнатися, що наслідування зроблено правильно?.....	211
Наслідування— добре чи погано?.....	213
У чому ж полягає справжня цінність наслідування?.....	214
Дотримання умов контракту: правила переописання.....	222
Перевантаження (overloading) методу.....	223
Вправи і головоломки.....	224

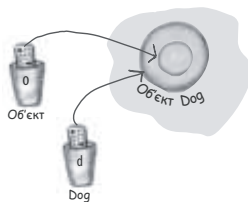


8 Серйозний поліморфізм

Наслідування — це лише початок. Аби задіяти поліморфізм, необхідні інтерфейси. Нам слід вийти за рамки простого наслідування, щоби ви могли отримати гнучкість лише проектуючи та кодуючи інтерфейси. Що таке інтерфейс? Це стовідсотково абстрактний клас. Що таке абстрактний клас? Це клас, для якого не можна створити екземпляр. Навіщо це потрібно? Читайте розділ...

Здається ми щось забули, коли розробляли це?.....	230
Компілятор не дозволить вам створити екземпляр для абстрактного класу.....	233
Абстрактний проти Конкретного.....	234
Абстрактні методи.....	235
Ви ПОВИННІ реалізовувати всі абстрактні методи.....	236
Поліморфізм у дії.....	238
Ох-ох! Тепер нам потрібно зберігати ще й об'єкти типу Cat!.....	239
Чому б не зробити клас настільки універсальним, аби він міг приймати будь-що? ...	240
Що ж містить цей ультра-супер-мегаклас Object?.....	241
Використання поліморфних посилань типу Object має свою ціну... ..	243
Коли Dog поводить себе не як Dog.....	244
Установіть зв'язок зі своїм внутрішнім Object.....	246
Що робити, якщо вам необхідно змінити контракт?.....	250
Розгляньмо варіанти проектування для повторного використання існуючих класів у програмі PetShop.....	251
Інтерфейс поспішає на допомогу!.....	256
Створення та реалізація інтерфейсу Pet.....	257
Вправи і головоломки.....	262

```
Object o = al.get(index);
Dog d = (Dog) o;
d.bark();
```

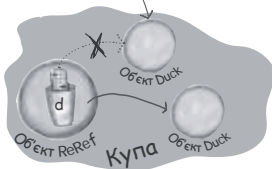


Приводимо
Об'єкт до Dog,
який, як ми знає-
мо, ним і є.

9 Життя і смерть об'єктів

Об'єкти народжуються і вмирають. Ви головний. Ви вирішуєте, коли і як саме створювати їх. І ви вирішуєте, коли відмовитися від них. **Збирач сміття (Garbage Collector — gc)** звільняє від них пам'ять. Ми розглянемо, як створюються об'єкти, де розміщуються, як їх ефективно використовувати і звільняти пам'ять від них, коли вони стають непотрібними. Тобто ми обговорюватимемо купу (heap), стек (stack), зону впливу (scope), конструктори, суперконструктори, null посилання і вимоги gc.

Коли хтось викликає метод `do()`, цей об'єкт `Duck` стає недоступним. Його єдине посилання перепрограмується на інший об'єкт `Duck`.



Змінний «d» призначено новий об'єкт `Duck`, при цьому оригінальний (початковий) об'єкт `Duck` стає недосяжним. Та перша качка відтепер засмажена.

Стек і купа: місця для життя. 268
 Як щодо локальних змінних, які є об'єктами? 270
 Якщо локальні змінні мешкають у стеці, то де перебувають змінні екземпляра? . . . 271
 Таїнство створення об'єкта 272
 Створюємо `Duck`. 274
 Ініціалізація стану нового об'єкта `Duck` 275
 Чи завжди компілятор створює конструктор без аргументів? Ні! 278
 Наногляд: чотири речі, які слід пам'ятати про конструктори 281
 Роль конструкторів суперкласу в житті об'єкта 283
 Як викликати конструктор суперкласу? 285
 Чи може дитина існувати до появи своїх батьків? 286
 Конструктори суперкласів з аргументами. 287
 Виклик одного перевантаженого конструктора з іншого. 288
 Тепер ми знаємо, як народжується об'єкт. Але скільки він живе? 290
 Як щодо змінних посилань? 292
 Вправи і головоломки 298

10 Числа мають значення

Займемося математикою. Java API має методи для роботи з числами на кшталт визначення абсолютного значення, округлення, пошук мінімуму / максимуму тощо. А як щодо форматування? Можливо, ви захочете, щоби числа друкувалися із точністю до двох десяткових знаків або з комами в усіх необхідних місцях. Ви також можете друкувати дати і маніпулювати ними. А як щодо розбору рядка в число? Або перетворення числа на рядок? Ми почнемо цей розділ із вивчення особливостей *статичної* змінної або методу.

Статичні змінні є загальними для всіх екземплярів класу.

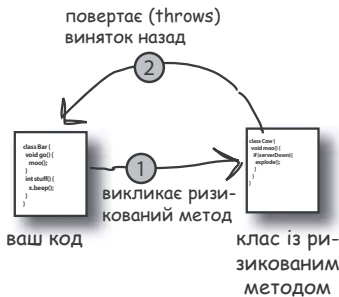


Методи класу `Math` — найближчі до глобальних. 306
 Різниця між звичайними (не статичними) і статичними методами (`static`) . . . 307
 Статичні методи 308
 Статична змінна 311
 Модифікатор `final` придатний не лише для статичних змінних... 315
 Математичні методи. 318
 Обгортка для примітивів 319
 Автоматичне пакування 321
 Стривайте! Це ще не все! 324
 А тепер навпаки... перетворення простого числового значення в `String`. . . 325
 Форматування чисел 326
 Про числа ми поговорили. А як щодо дат? 333
 Переміщення назад і вперед у часі 335
 Отримання об'єкта, що розширює `Calendar`. 336
 Робота з об'єктами `Calendar` 337
 Ще більше статички! `static imports` 339
 Вправи і головоломки 342

11

Небезпечна поведінка

Трапляються різні речі. То файл зникає. То сервер падає. Хай би яким вправним розробником ви були, неможливо контролювати геть усе. При створенні ризикованого методу вам знадобиться код, що буде обробляти можливі нестандартні ситуації. Але яким чином дізнатися, чи ризиковий цей метод? І куди помістити код для обробки *непередбаченої* ситуації? У цьому розділі ми розробимо музичний MIDI-плеєр, що використовує ризикове JavaSound API.



Давайте створимо Музичну Машину.....	348
Розпочнемо з основ.....	349
Для початку нам потрібен Sequencer.....	350
Ризикові методи.....	351
Виняток — це об'єкт... типу Exception.....	354
Керування програмним потоком у блоках try/catch.....	358
Finally: для дій, що їх потрібно виконати НЕЗВАЖАЮЧИ НІ НА ЩО... ..	359
Винятки є поліморфними.....	362
Оминаючи (оголошуючи) виняток, ви лише відкладаєте неминуче.....	368
Повертаємося до нашого музичного коду.....	370
Відтворення звуку.....	372
Створення MIDI-події (даних про композицію).....	375
MIDI-повідомлення: серце MidiEvent.....	376
Змінюємо повідомлення.....	377
Версія 2. Використання аргументів командного рядка для експериментів зі звуками... ..	378
Вправи і головоломки.....	380

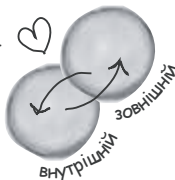
12

Дуже графічна історія

Готуйтеся, вам доведеться створювати GUI (*graphical user interface* — графічний користувацький інтерфейс). Навіть якщо ви впевнені, що все життя будете писати код для серверних програм, рано чи пізно вам доведеться створювати інструменти, і тоді виникне потреба у використанні графічного інтерфейсу. Роботі над GUI присвячені два розділи, із яких ви дізнаєтеся про ключові особливості мови Java, включно з **обробкою подій і внутрішніми класами**. Ми розмістимо кнопку на екрані, малюватимемо на екрані, додаватимемо зображення у форматі JPEG і навіть створюватимемо анімацію.

```
class MyOuter {
    class MyInner {
        void go() {
        }
    }
}
```

Зовнішня та внутрішня об'єкти відтепер тісно пов'язані.

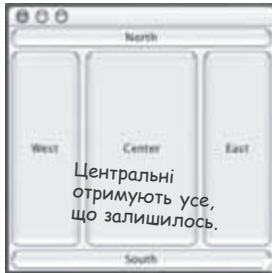


Ці два об'єкти на купі мають особливий зв'язок. Внутрішній може використовувати змінні зовнішнього (і навпаки).

Усе починається з вікна.....	386
Отримуємо користувацьку подію.....	389
Повернемося до графіки... ..	395
Створіть власний віджет для малювання.....	396
Що цікавого можна зробити в paintComponent ().....	397
За кожним об'єктом Graphics стоїть об'єкт Graphics2D.....	398
Компонування графічних елементів: розміщуємо декілька віджетів у фреймі... ..	402
Спробуємо зробити це із ДВОМА кнопками.....	404
Отже, зараз нам знадобляться ЧОТИРИ віджети.....	404
І нам потрібно отримати ДВІ події.....	404
Внутрішній клас поспішає на допомогу!.....	408
Те, чого ми дійсно прагнемо, являє собою щось на кшталт... ..	415
Відстежуємо подію, не пов'язану з GUI.....	419
Легкий спосіб створення повідомлень/подій.....	420
Приклад: використовуємо новий статичний метод makeEvent ().....	421
Версія друга: реєструємо і отримуємо ControllerEvents.....	422
Вправи і головоломки.....	426

13 Попрацюйте над Swing

Компоненти на півночі та півдні отримують бажану висоту.



Компоненти на заході та сході отримують бажану ширину.

Swing — це просто. Доти, доки вам байдуже, де ви опинитесь. Код для роботи зі Swing виглядає просто, але, скопіювавши його, запустивши і подивившись на екран, ви подумаете: «Агов, цей об'єкт має бути в іншому місці!» Інструмент, що *спрощує створення коду і водночас ускладнює управління ним*, — це **менеджер макету інтерфейсу (Layout Manager)**. Проте, доклавши невеликих зусиль, ви зможете підкорити його. У цьому розділі ми будемо працювати зі Swing і ближче познайомимося з віджетами.

Компоненти Swing	432
Менеджери компоновання (Layout managers)	433
Як менеджер компоновання ухвалює рішення	434
Три головні менеджери компоновання: border, flow та box	435
Пограємося зі Swing-компонентами	445
Створюємо BeatBox	451
Вправи і головоломки	456

14 Збереження об'єктів

Об'єкти можуть бути стиснутими та відновленими. Вони характеризуються станом і поведінкою. Поведінка міститься в класі, а стан визначається кожним об'єктом окремо. Якщо вашій програмі необхідно зберегти стан, ви можете зробити це складним способом, опитуючи кожен об'єкт і старанно записуючи значення кожної змінної екземпляра. Або ви можете **зробити це легким ОО-способом** — просто застосуйте суху заморозку (серіалізуйте об'єкт) та відновіть (десеріалізуйте), аби отримати його знов.

Лови ритм	462
Збереження стану	463
Запис серіалізованого об'єкта у файл	464
Що насправді відбувається з об'єктом під час серіалізації	466
Але що конкретно є станом об'єкта? Що слід зберігати?	467
Інтерфейс Serializable	469
Десеріалізація: відновлення об'єкта	473
Збереження і відновлення ігрових персонажів	476
Клас GameCharacter	477
Серіалізація об'єктів	478
Запис рядка в текстовий файл	479
Приклад текстового файлу: електронні флеш-карти	480
Quiz Card Builder, структура коду	481
Клас java.io.File	484
Читання з текстового файлу	486
Quiz Card Player, структура коду	487
Розбір тексту за допомогою методу split() із класу String	490
Ідентифікатор версії: «великий капкан» серіалізації	492
Використання serialVersionUID	493
Збереження патерну BeatBox	495
Відновлення схеми BeatBox	496
Вправи і головоломки	498

Серіалізований



Маєте питання?

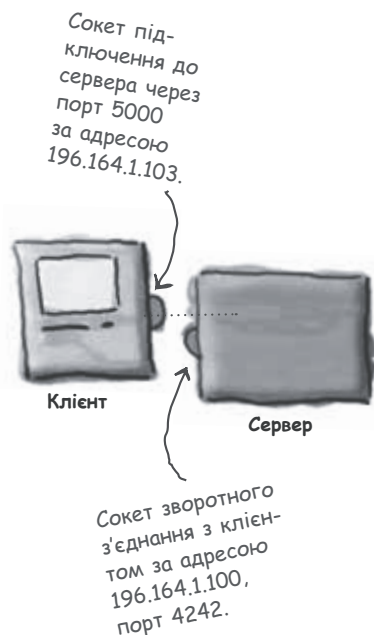


Десеріалізований

15

Встановлюємо зв'язок

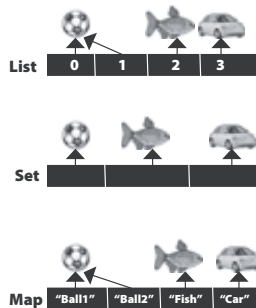
Зв'яжіться із зовнішнім світом. Це просто. Про всі низькорівневі мережеві компоненти піклуються класи з бібліотеки `java.net`. Однією з найкращих особливостей Java є відправка й отримання даних, використовуючи мережу,— це традиційне «введення/виведення» (I/O), хіба що з трохи зміниним з'єднувальним потоком (connection stream) наприкінці ланцюга викликів. У цьому розділі ми створимо клієнтські сокети. А також серверні сокети. Зрештою ми матиме клієнтів і сервери. А до завершення цього розділу ми матимемо повноцінний багатопотоковий клієнт-чат. Здається, ми щойно сказали «багатопотоковий»?



Чат Beat Vox у режимі реального часу	504
Підключення, надсилання та отримання	506
Мережеві сокети	507
TCP-порт — це просто число. 16-бітове число, що розпізнає конкретну програму на сервері	508
Зчитування даних із Socket (за допомогою <code>BufferedReader</code>) ...	510
Запис даних до сокета (за допомогою <code>PrintWriter</code>)	511
<code>DailyAdviceClient</code>	512
Код <code>DailyAdviceClient</code>	513
Створення простого сервера	515
Код застосунку <code>DailyAdviceServer</code>	516
Пишемо чат-клієнт	518
Java підтримує численні потоки, але має лише один клас <code>Thread</code>	522
Що значить мати більше, ніж один стек викликів	523
Щоби виконати завдання для вашого потоку, реалізуйте інтерфейс <code>Runnable</code>	526
Планувальник потоків	529
Поклавши потік спати	533
Використання <code>sleep()</code> для створення більш передбачуваної програми	534
Створення і запуск двох потоків	535
Що станеться?	535
Хм, так. Існує зворотна сторона медалі. Використання потоків може призвести до проблем «конкурентності»	536
Проблема дій Раяна та Моніки в коді	538
Необхідно зробити так, щоб метод <code>makeWithdrawal()</code> виконувався атомарно	542
Використання механізму блокування з об'єкту	543
Жахлива проблема під назвою «Втрачене оновлення (Lost Update)»	544
Давайте запустимо цей код	545
Зробімо метод <code>increment()</code> атомарним. Синхронізуймо його!	546
Смертельна сторона синхронізації	548
Багатопотоковий код <code>ChatClient</code>	550
Справді дуже простий чат-сервер	552
Вправи і головоломки	556

16 Структури даних

Сортування структур даних у Java. Це дуже просто. Ви вже маєте всі необхідні інструменти для збирання даних і керування ними, тому не має потреби писати власні алгоритми сортування. Фреймворк для роботи з колекціями в Java (Java collections framework) містить структури даних, які підійдуть практично для всього, що вам може знадобитися. Бажаєте отримати перелік, до якого можна додавати елементи? Вам необхідно знайти щось за ім'ям? Хочете створити перелік, що автоматично видаляє всі дублікати? Бажаєте впорядкувати своїх приятелів за кількістю ударів, що вони нанесли вам у спину?..



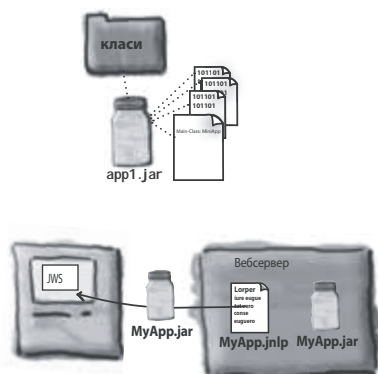
Відстеження популярності пісень у музичному автоматі	562
Ось що ви маєте на поточний момент без сортування	563
Але клас ArrayList НЕ має методу sort()!.....	564
Ви можете скористатися TreeSet... Метод Collections.sort()	566
Використання Songs замість Strings	569
Оголошення методу sort()	571
Вивчаємо шаблони.....	573
Використання шаблонних класів	574
Використання параметрів типу в ArrayList	575
Використання шаблонних методів	576
Повертаємося до методу sort()	579
У контексті шаблонів «extends» означає «extends або implements»	580
Нарешті ми дізналися, що не так.....	581
Новий, удосконалений клас Song, чий об'єкти можна порівнювати	582

Ми можемо відсортувати список, але.....	583
Використання власного компаратора	584
Оновлюємо програму музичного автомата із використанням Comparator.....	585
Ой-ой! Сортування працює, але тепер у нас є дублікати... ..	588
Замість List нам потрібен Set	589
API для роботи з колекціями (частина).....	590
Використовування HashSet замість ArrayList	591
Що робить два об'єкти однаковими	592
Перевизначення hashCode() та equals()	593
Клас Song із перевизначеними методами hashCode() та equals()	594
І якщо ми хочемо, щоби множина залишалася відсортованою, у нас є TreeSet	596
Усе, що ви ПОВИННІ знати про TreeSet.....	597
Елементи TreeSet ПОВИННІ бути Comparable	598
Ми бачили List та Set, а тепер спробуємо використати Map.....	599
Нарешті повертаємося до шаблонів	600
Використання поліморфних аргументів і шаблонів	601
Чи спрацює ArrayList<Dog> ?.....	602
Що може статися, якби це було дозволено.....	603
Використання шаблонних символів йде на допомогу	606
Альтернативний синтаксис для виконання тих самих задач	607
Вправи і головоломки	608

17

Випуск коду

Настав час його відпустити. Ви написали код. Протестували і відкоригували його. Розповіли всім знайомим, що більше не бажаєте бачити жодного його рядка. Зрештою, це справжній витвір мистецтва. Це те, що дійсно працює! Але що далі? В останніх двох розділах ми розповімо, як організовувати, упаковувати і розгортати код на мові Java. Ми розглянемо локальний, напівлокальний і віддалений варіанти розгортання, включно з виконуваними архівами Java (jar), Java Web Start, RMI і сервлетами. Розслабтеся. Деякі з найкрутіших речей в Java простіші ніж ви думаєте.

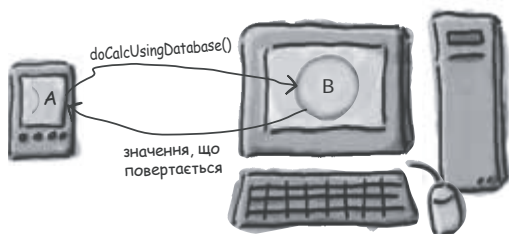


Варіанти розгортання вашого застосунку	614
Уявіть собі цей сценарій.....	615
Зберігайте початковий код окремо від файлів класів.....	616
Сховайте вашу Java в JAR	617
Запуск (виконання) JAR.....	618
Покладіть ваші класи до пакетів!.....	619
Запобігання конфліктам імен пакета.....	620
Компілюємо та запускаємо, використовуючи пакети.....	622
Прапорець -d навіть крутіший ніж ми вам казали	623
Створення виконуваних JAR (із пакетами всередині)	624
Тож куди йде файл маніфесту?	625
Java Web Start (JWS)	629
Як працює Java Web Start	630
Файл .jnlp	631
Вправи і головоломки	633

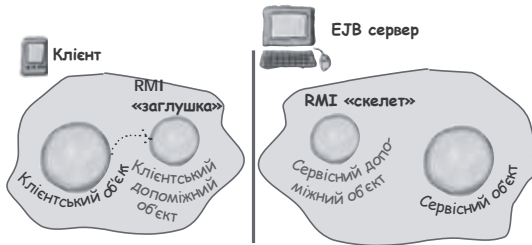
18

Розподілені обчислення

Бути віддаленим — це не завжди погано. Звісно, завдання спрощується, коли всі компоненти програми зібрані в одному місці і всім керує одна JVM. Але це не завжди можливо. Або не завжди доцільно. Що робити, якщо ваш застосунок виконує складні обчислення? А якщо вашому застосунку необхідна інформація із захищеної бази даних? У цьому розділі ми навчимося користуватися дивовижно простим викликом віддалених методів (Remote Method Invocation, RMI) у мові Java. Ми також нашвидкуруч ознайомимося із сервлетами (Servlets), Enterprise Java Beans (EJB) та Jini.



Виклики методів між двома об'єктами завжди відбуваються в одній купі.....	640
Що робити, якщо вам потрібно викликати метод з об'єкта, що працює на іншій машині?.....	641
Об'єкт А, запущений на Малюку, намагається викликати метод з об'єкта В, що працює на Велетні ..	642
Але ти не можеш цього зробити!	642
Роль «помічників»	644
Технологія RMI: дуже детально	646
Як клієнт отримує об'єкт «заглушка»?	652
Переконайтеся, що кожна машина має свої потрібні класи.....	654
Так, але хто насправді використовує RMI?	656
А як щодо сервлетів?	657



Дуже простий сервлет 659
 Сторінка HTML із посиланням на цей сервлет 659
 Давайте заради розваги розробимо генератор фраз, що працюватиме у вигляді сервлета 661
 Код генератора фраз, зручний для сервлетів 662
 Enterprise JavaBeans: RMI на стероїдах 663
 Наша остання хитрість — маленький Jini 664
 Адаптивна маршрутизація в дії 665
 Самовідтворювані мережі в дії 667
 Фінальний проект: універсальний оглядач сервісів (Universal Service browser)..... 668

Додаток А681

Заключний проєкт «Кухня Коду». Весь код для повноцінного клієнт-серверного бітбокс-чату. Маєш шанс стати рок-зіркою.



Додаток Б691

Десять важливих речей, що не потрапили до цієї книжки. Ми не наважуємося відпустити читача, бо маємо для нього ще кілька тем. Але це насправді кінець. І цього разу — цілком серйозно.

Перелік десяти найкращих речей692

Показчик708

Кінець безкоштовного уривку. Щоби читати далі,
придбайте, будь ласка, повну версію книги.