Зображення, що містить метал, крок, камінь

Автоматично згенерований опис

**переклад С. Ярошко**

**С. Дюкас**

**Г. Ракіч**

**та**

**С. Каплар**

**К. Дюкас**

Pharo 9

на прикладах



Pharo 9 на прикладах

*Пориньте у світ живих об’єктів*

Pharo 9 by Example

Stéphane Ducasse and Gordana Rakic with Sebastijan Kaplar and Quentin Ducasse

September 1, 2022

BoD – Books on Demand

Pharo 9 на прикладах

Стефан Дюкас і Джордана Ракіч разом з Себастьян Каплар і Квентін Дюкас

Переклад українською з доповненнями

Сергій Ярошко

ЛНУ ім. Івана Франка

2022

УДК 004.432.2

Д 95

**Перекладено за оригіналом:**

Stéphane Ducasse and Gordana Rakic with Sebastijan Kaplar and Quentin Ducasse. Pharo 9 by Example

**Переклад:**

*Ярошко С. А.* – канд. фіз.-мат. наук, доцент   
(Львівський національний університет імені Івана Франка)

Перекладено з дотриманням умов ліцензії Creative Commons Attribution-  
ShareAlike 3.0 Unported.



**Рецензенти:**

*Іванчук Я. В.* – д-р техн. наук, професор   
(Вінницький національний технічний університет);

*Міца О. В.* – д-р техн. наук, доцент   
(Ужгородський національний університет);

*Яремко Г. В.* – канд. пед. наук, доцент   
(Національний університет «Львівська політехніка»)

*Рекомендовано до друку   
Вченою радою Львівського національного університету імені Івана Франка.   
Протокол № 37/10 від 26.10.2022 р.*

**Стефан Дюкас**

Pharo 9 на прикладах / С. Дюкас, Дж. Ракіч [та ін.] ; пер. з англ. С. Ярошка. – Львів : ЛНУ ім. Івана Франка, 2022. – 270 с.

**Д 95**

**ISBN 978-617-10-0757-4**

Pharo – це сучасна динамічно типізована об’єктно-орієнтована мова програму­вання з відкритим кодом і середовище інтерактивного програмування, сучасне втілення класичної системи Smalltalk. У книзі розглянуто синтаксис мови та типові способи використання середовища програмування, наведено велику кількість прикладів використання наявних класів і оголошення нових. Описано об’єктну модель Pharo, базові класи, колекції, взаємодію з потоками, систему модульного тестування й основи застосування технології розробки, керованої тестами, засоби рефлексії, інструменти та способи взаємодії з системами контролю версій.

Для студентів факультету прикладної математики та інформатики і всіх, хто цікавиться сучасними технологіями програмування.

**УДК 004.432.2**

ISBN 978-617-10-0757-4

© Дюкас С., Ракіч Дж., [та ін.], 2022

© Ярошко Сергій, переклад, 2022

© Львівський національний університет імені Івана Франка, 2022

Зміст

1. [Про цю книгу 1](#_Toc121067148)

[1.1. Що таке Pharo 1](#_Toc121067149)

[1.2. Для кого ця книга 2](#_Toc121067150)

[1.3. Порада 3](#_Toc121067151)

[1.4. Відкрита книга 4](#_Toc121067152)

[1.5. Спільнота Pharo 4](#_Toc121067153)

[1.6. Приклади і вправи 5](#_Toc121067154)

[1.7. Друкарські домовленості 5](#_Toc121067155)

[1.8. Подяки 6](#_Toc121067156)

[1.9. Спеціальні подяки 6](#_Toc121067157)

1. [Початок роботи з Pharo 7](#_Toc121067158)

[2.1. Встановлення Pharo 7](#_Toc121067159)

[2.2. Файли компонентів Pharo 7](#_Toc121067160)

[2.3. Запуск Pharo за допомогою Pharo Launcher 9](#_Toc121067161)

[2.4. Запуск Pharo з командного рядка 10](#_Toc121067162)

[2.5. Зберігання, завершення та повторний запуск сесії Pharo 12](#_Toc121067163)

[2.6. Для швидких і нетерплячих 13](#_Toc121067164)

[2.7. Підсумки розділу 14](#_Toc121067165)

1. [Швидкий огляд Pharo 15](#_Toc121067166)

[3.1. Головне меню 15](#_Toc121067167)

[3.2. Взаємодія з Pharo 15](#_Toc121067168)

[3.3. Вікна Playground і Transcript 18](#_Toc121067169)

[3.4. Гарячі клавіші 19](#_Toc121067170)

[3.5. Виконання проти виведення 19](#_Toc121067171)

[3.6. Інспектування 19](#_Toc121067172)

[3.7. Інші дії 20](#_Toc121067173)

[3.8. Оглядач класів Calypso 21](#_Toc121067174)

[3.9. Підсумки розділу 22](#_Toc121067175)

1. [Пошук інформації у Pharo 23](#_Toc121067176)

[4.1. Мандрівка системою за допомогою Оглядача 23](#_Toc121067177)

[4.2. Відшукання класів 24](#_Toc121067178)

[4.3. Відшукання методів 27](#_Toc121067179)

[4.4. Пошук методів за зразками 28](#_Toc121067180)

[4.5. Підсумки розділу 29](#_Toc121067181)

1. [Перший практикум. Розробка простого лічильника 30](#_Toc121067182)

[5.1. Завдання 30](#_Toc121067183)

[5.2. Створення пакета і класу 31](#_Toc121067184)

[5.3. Визначення протоколів і методів 33](#_Toc121067185)

[5.4. Створення методу 35](#_Toc121067186)

[5.5. Додавання методу запису значення 35](#_Toc121067187)

[5.6. Визначення класу тестів 36](#_Toc121067188)

[5.7. Зберігання коду в git-репозиторії за допомогою Iceberg 37](#_Toc121067189)

[5.8. Додавання нових методів 40](#_Toc121067190)

[5.9. Метод ініціалізації екземпляра 41](#_Toc121067191)

[5.10. Визначення методу ініціалізації 41](#_Toc121067192)

[5.11. Визначення методу створення нового екземпляра 41](#_Toc121067193)

[5.12. Покращення опису об’єкта 42](#_Toc121067194)

[5.13. Зберігання коду на віддаленому сервері 43](#_Toc121067195)

[5.14. Підсумки розділу 45](#_Toc121067196)

1. [Створення невеликої гри 46](#_Toc121067197)

[6.1. Гра Lights Out 46](#_Toc121067198)

[6.2. Створення нового пакета класів 46](#_Toc121067199)

[6.3. Визначення класу *LOCell* 47](#_Toc121067200)

[6.4. Створення нового класу 47](#_Toc121067201)

[6.5. Про коментарі 48](#_Toc121067202)

[6.6. Додавання методів до класу 49](#_Toc121067203)

[6.7. Інспектування об’єкта 50](#_Toc121067204)

[6.8. Визначення класу *LOGame* 52](#_Toc121067205)

[6.9. Ініціалізація гри 53](#_Toc121067206)

[6.10. Використання переваг Налагоджувача 54](#_Toc121067207)

[6.11. Вивчаємо метод *initialize* 55](#_Toc121067208)

[6.12. Поділ методів на протоколи 57](#_Toc121067209)

[6.13. Завершення розробки гри 57](#_Toc121067210)

[6.14. Останні методи класу *LOCell* 59](#_Toc121067211)

[6.15. Випробуємо код 60](#_Toc121067212)

[6.16. Домовленості щодо найменування методів доступу 61](#_Toc121067213)

[6.17. Про налагоджувач 62](#_Toc121067214)

[6.18. Удосконалення гри 63](#_Toc121067215)

[6.19. Зберігання та поширення коду Pharo 67](#_Toc121067216)

[6.20. У Pharo ви не можете втратити код 69](#_Toc121067217)

[6.21. Підсумки розділу 69](#_Toc121067218)

1. [Публікація вашого першого проєкту Pharo 70](#_Toc121067219)

[7.1. Для нетерплячих 70](#_Toc121067220)

[7.2. Базова архітектура 71](#_Toc121067221)

[7.3. Про оглядач репозиторіїв Iceberg 73](#_Toc121067222)

[7.4. Додавання нового проєкту до Iceberg 73](#_Toc121067223)

[7.5. Додайте і збережіть пакет за допомогою оглядача робочої копії 76](#_Toc121067224)

[7.6. Що робити, якщо віддалений репозиторій не створювали? 78](#_Toc121067225)

[7.7. Конфігурування вашого проєкту 80](#_Toc121067226)

[7.8. Завантаження з наявного репозиторію 81](#_Toc121067227)

[7.9. Оглядаючись назад… 82](#_Toc121067228)

[7.10. Підсумок розділу 82](#_Toc121067229)

1. [Два слова про синтаксис 83](#_Toc121067230)

[8.1. Синтаксичні елементи 83](#_Toc121067231)

[8.2. Псевдозмінні 86](#_Toc121067232)

[8.3. Повідомлення і надсилання повідомлень 87](#_Toc121067233)

[8.4. Послідовності та каскади 88](#_Toc121067234)

[8.5. Синтаксис методу 88](#_Toc121067235)

[8.6. Синтаксис блока 89](#_Toc121067236)

[8.7. Галуження і повторення 90](#_Toc121067237)

[8.8. Анотації методів: примітиви і прагми 92](#_Toc121067238)

[8.9. Підсумки розділу 93](#_Toc121067239)

1. [Розуміння синтаксису повідомлень 95](#_Toc121067240)

[9.1. Розпізнавання повідомлень 95](#_Toc121067241)

[9.2. Три види повідомлень 97](#_Toc121067242)

[9.3. Композиція повідомлень 99](#_Toc121067243)

[9.4. Як розпізнати ключове повідомлення 104](#_Toc121067244)

[9.5. Послідовність повідомлень 105](#_Toc121067245)

[9.6. Каскад повідомлень 106](#_Toc121067246)

[9.7. Підсумки розділу 106](#_Toc121067247)

1. [Об’єктна модель Pharo 108](#_Toc121067248)

[10.1. Правила базової моделі 108](#_Toc121067249)

[10.2. Все є об’єктом 108](#_Toc121067250)

[10.3. Кожен об’єкт є екземпляром класу 109](#_Toc121067251)

[10.4. Структура та поведінка екземпляра 110](#_Toc121067252)

[10.5. Кожен клас має надклас 111](#_Toc121067253)

[10.6. Усе відбувається через надсилання повідомлень 112](#_Toc121067254)

[10.7. Надсилання повідомлення – двокроковий процес 114](#_Toc121067255)

[10.8. Алгоритм пошуку методу перебирає ланцюжок наслідування 114](#_Toc121067256)

[10.9. Виконання методу 115](#_Toc121067257)

[10.10. Об’єкт не зрозумів повідомлення 116](#_Toc121067258)

[10.11. Про повернення *self* 117](#_Toc121067259)

[10.12. Перевантаження та розширення 117](#_Toc121067260)

[10.13. Надсилання до *self* і *super* 118](#_Toc121067261)

[10.14. Крок назад 120](#_Toc121067262)

[10.15. Сторона екземпляра та сторона класу 120](#_Toc121067263)

[10.16. Методи класу 122](#_Toc121067264)

[10.17. Змінні екземпляра класу 123](#_Toc121067265)

[10.18. Приклад. Змінні екземпляра класу та підкласи 124](#_Toc121067266)

[10.19. Крок назад 124](#_Toc121067267)

[10.20. Приклад. Оголошення Одинака 125](#_Toc121067268)

[10.21. Зауваження щодо лінивої ініціалізації 126](#_Toc121067269)

[10.22. Спільні змінні 127](#_Toc121067270)

[10.23. Змінні класу 128](#_Toc121067271)

[10.24. Змінні пулу 129](#_Toc121067272)

[10.25. Абстрактні методи і абстрактні класи 130](#_Toc121067273)

[10.26. Приклад. Абстрактний клас *Magnitude* 131](#_Toc121067274)

[10.27. Підсумки розділу 131](#_Toc121067275)

1. [Ознаки – код для повторного використання 133](#_Toc121067276)

[11.1. Проста ознака 133](#_Toc121067277)

[11.2. Виклик необхідного методу 134](#_Toc121067278)

[11.3. У методі ознаки *self* вказує на отримувача 134](#_Toc121067279)

[11.4. Стан ознаки 135](#_Toc121067280)

[11.5. Клас може використати кілька ознак 136](#_Toc121067281)

[11.6. Перевизначений метод має вищий пріоритет, ніж метод ознаки 136](#_Toc121067282)

[11.7. Доступ до перевантаженого методу ознаки 137](#_Toc121067283)

[11.8. Опрацювання конфлікту 138](#_Toc121067284)

[11.9. Вирішення конфлікту – вилучити метод 138](#_Toc121067285)

[11.10. Вирішення конфлікту – перевизначити метод 139](#_Toc121067286)

[11.11. Ознаки та наслідування 139](#_Toc121067287)

[11.12. Про що не було сказано 140](#_Toc121067288)

[11.13. Підсумок розділу 142](#_Toc121067289)

1. [SUnit – модульне тестування у Pharo 143](#_Toc121067290)

[12.1. Вступ 143](#_Toc121067291)

[12.2. Чому тестування важливе 143](#_Toc121067292)

[12.3. Що робить тест хорошим? 144](#_Toc121067293)

[12.4. SUnit крок за кроком 145](#_Toc121067294)

[12.5. Крок 1. Створіть клас тестів 145](#_Toc121067295)

[12.6. Крок 2. Налаштуйте контекст виконання тестів 145](#_Toc121067296)

[12.7. Крок 3. Напишіть кілька методів тестування 146](#_Toc121067297)

[12.8. Крок 4. Запустіть тести 147](#_Toc121067298)

[12.9. Крок 5. Потрактуйте результати тестування 148](#_Toc121067299)

[12.10. Використання assert:equals: 148](#_Toc121067300)

[12.11. Як пропустити тест 148](#_Toc121067301)

[12.12. Перевірка виникнення винятків 149](#_Toc121067302)

[12.13. Програмний запуск тестів 149](#_Toc121067303)

[12.14. Підсумок розділу 150](#_Toc121067304)

1. [Базові класи 151](#_Toc121067305)

[13.1. Object 151](#_Toc121067306)

[13.2. Друк об’єкта 151](#_Toc121067307)

[13.3. Зображення і самовідтворення 153](#_Toc121067308)

[13.4. Ідентичність і рівність 154](#_Toc121067309)

[13.5. Належність до класу 154](#_Toc121067310)

[13.6. Про isKindOf: і respondsTo: 155](#_Toc121067311)

[13.7. Поверхневе копіювання об’єктів 155](#_Toc121067312)

[13.8. Глибоке копіювання об’єктів 156](#_Toc121067313)

[13.9. Налагодження 157](#_Toc121067314)

[13.10. Опрацювання винятків 158](#_Toc121067315)

[13.11. Тестування 159](#_Toc121067316)

[13.12. Ініціалізація 159](#_Toc121067317)

[13.13. Числа 159](#_Toc121067318)

[13.14. Magnitude 160](#_Toc121067319)

[13.15. Можливості чисел 160](#_Toc121067320)

[13.16. Дійсні 162](#_Toc121067321)

[13.17. Раціональні 162](#_Toc121067322)

[13.18. Цілі 162](#_Toc121067323)

[13.19. Літери 163](#_Toc121067324)

[13.20. Рядки 164](#_Toc121067325)

[13.21. Булеві величини 166](#_Toc121067326)

[13.22. Підсумки розділу 167](#_Toc121067327)

1. [Колекції 169](#_Toc121067328)

[14.1. Функції вищого порядку 170](#_Toc121067329)

[14.2. Різноманіття колекцій 171](#_Toc121067330)

[14.3. Реалізація колекцій 171](#_Toc121067331)

[14.4. Приклади головних класів 172](#_Toc121067332)

[14.5. Загальний протокол створення 172](#_Toc121067333)

[14.6. Array 174](#_Toc121067334)

[14.7. OrderedCollection 176](#_Toc121067335)

[14.8. Interval 176](#_Toc121067336)

[14.9. Dictionary 177](#_Toc121067337)

[14.10. IdentityDictionary 178](#_Toc121067338)

[14.11. Set 179](#_Toc121067339)

[14.12. SortedCollection 180](#_Toc121067340)

[14.13. Рядки 181](#_Toc121067341)

[14.14. Ітератори колекцій 185](#_Toc121067342)

[14.15. Збір результатів (*collect:*) 187](#_Toc121067343)

[14.16. Вибір і відхилення елементів 188](#_Toc121067344)

[14.17. Інші повідомлення вищого порядку 189](#_Toc121067345)

[14.18. Загальна помилка – використання результату *add:* 189](#_Toc121067346)

[14.19. Загальна помилка – вилучення елемента під час перебору 190](#_Toc121067347)

[14.20. Загальна помилка – перевизначення = без *hash* 190](#_Toc121067348)

[14.21. Підсумки розділу 191](#_Toc121067349)

1. [Потоки 192](#_Toc121067350)

[15.1. Дві послідовності елементів 192](#_Toc121067351)

[15.2. Потоки проти колекцій 193](#_Toc121067352)

[15.3. Віддавайте перевагу функціям інтерфейсу 193](#_Toc121067353)

[15.4. Читання колекцій 194](#_Toc121067354)

[15.5. Підглядання 195](#_Toc121067355)

[15.6. Керування вказівником потоку 196](#_Toc121067356)

[15.7. Пропуск елементів 196](#_Toc121067357)

[15.8. Предикати 197](#_Toc121067358)

[15.9. Запис у колекцію 197](#_Toc121067359)

[15.10. Про конкатенацію рядків 198](#_Toc121067360)

[15.11. Про *printString* 199](#_Toc121067361)

[15.12. Одночасні читання та запис 202](#_Toc121067362)

[15.13. Використання потоків для доступу до файлів 203](#_Toc121067363)

[15.14. Підсумки розділу 204](#_Toc121067364)

1. [Морфи 205](#_Toc121067365)

[16.1. Історія створення 205](#_Toc121067366)

[16.2. Морфи 206](#_Toc121067367)

[16.3. Маніпулювання морфами 207](#_Toc121067368)

[16.4. Композиція морфів 208](#_Toc121067369)

[16.5. Створення і малювання власних морф 209](#_Toc121067370)

[16.6. Взаємодія через події мишки 212](#_Toc121067371)

[16.7. Події клавіатури 213](#_Toc121067372)

[16.8. Анімація морф 214](#_Toc121067373)

[16.9. Діалоги 215](#_Toc121067374)

[16.10. Перетягування 215](#_Toc121067375)

[16.11. Завершений приклад 218](#_Toc121067376)

[16.12. Більше про полотно малювання 221](#_Toc121067377)

[16.13. Підсумки до розділу 222](#_Toc121067378)

1. [Класи і метакласи 223](#_Toc121067379)

[17.1. Правила для класів 223](#_Toc121067380)

[17.2. Метакласи 223](#_Toc121067381)

[17.3. Ще раз про об’єктну модель Pharo 224](#_Toc121067382)

[17.4. Кожен клас є екземпляром свого метакласу 225](#_Toc121067383)

[17.5. Запити до метакласів 226](#_Toc121067384)

[17.6. Ієрархія метакласів паралельна ієрархії класів 226](#_Toc121067385)

[17.7. Однорідність класів і об’єктів 228](#_Toc121067386)

[17.8. Інспектування об’єктів і класів 228](#_Toc121067387)

[17.9. Кожен метаклас наслідує класи *Class* і *Behavior* 229](#_Toc121067388)

[17.10. Призначення класів *Behavior*, *ClassDescription* i *Class* 231](#_Toc121067389)

[17.11. Кожен метаклас є екземпляром класу *Metaclass* 232](#_Toc121067390)

[17.12. Метаклас класу *Metaclass* є екземпляром класу *Metaclass* 233](#_Toc121067391)

[17.13. Підсумки розділу 234](#_Toc121067392)

1. [Рефлексія 235](#_Toc121067393)

[18.1. Суть рефлексії 235](#_Toc121067394)

[18.2. Інтроспекція 236](#_Toc121067395)

[18.3. Доступ до змінних екземпляра 237](#_Toc121067396)

[18.4. Про застосування рефлексії 238](#_Toc121067397)

[18.5. Про примітиви 238](#_Toc121067398)

[18.6. Опитування класів та інтерфейсів 238](#_Toc121067399)

[18.7. Прості метрики коду 239](#_Toc121067400)

[18.8. Дослідження екземплярів 240](#_Toc121067401)

[18.9. Від методів до змінних екземпляра 241](#_Toc121067402)

[18.10. Про *SystemNavigation* 241](#_Toc121067403)

[18.11. Класи, словники методів і методи 243](#_Toc121067404)

[18.12. Середовища перегляду 245](#_Toc121067405)

[18.13. Прагми – анотації методів 246](#_Toc121067406)

[18.14. Доступ до контексту етапу виконання 247](#_Toc121067407)

[18.15. Інтелектуальні контекстні точки переривання 249](#_Toc121067408)

[18.16. Перехоплення незрозумілих повідомлень 250](#_Toc121067409)

[18.17. Легкий проксі-сервер 251](#_Toc121067410)

[18.18. Генерування методів, яких бракує 253](#_Toc121067411)

[18.19. Об’єкти як обгортки методів 255](#_Toc121067412)

[18.20. Підсумки розділу 257](#_Toc121067413)

Глосарій

|  |  |  |
| --- | --- | --- |
| **Debugger** | *Налагоджувач* | Програма, інструмент для дослідження коду, що виконується. Можна використовувати для написання програм на льоту |
| **File Browser** | *Оглядач файлів* | Інструмент для доступу до файлової системи комп’ютера, має можливості завантаження, швидкого перегляду файлів |
| **Finder** | *Шукач* | Програма, інструмент для відшукання мето­дів, класів, програмного коду, анотацій, при­кладів. Уміє шукати за зразком перетворень |
| **Iceberg** | *Менеджер пакетів* | Система контролю версій для взаємодії з Git |
| **Image** | *Образ системи* | Колекція всіх об’єктів системи, завантажується у віртуальну машину |
| **Inspector** | *Інспектор об’єктів* | Програма, інструмент, який дає змогу загля­нути всередину об’єкта, вивчити та змінити його стан |
| **Morph** | *Морфа* | Видимий елемент інтерфейсу користувача |
| **Pharo Launcher** | *Пускова програма* | Спеціальна програма, написана на Pharo, для відстежування та запуску версій Pharo |
| **Playground** | *Пісочниця* | Вікно для безпечного виконання фрагментів коду, інтегроване з інспектором об’єктів |
| **Spotter** | *Навідник* | Вікно діалогу для швидкого відшукання будь-чого в системі |
| **System Browser** | *Оглядач класів, Системний оглядач* | Головний інструмент системи Pharo, який надає доступ до всіх пакетів класів, їхніх методів, дає змогу оголошувати нові класи та редагувати наявні |
| **Test Runner** | *Менеджер тестів* | Програма керування модульними тестами |
| **Transcript** | *Консоль системи* | Текстове вікно, яке можна використовувати для виведення тестових повідомлень від об’­єктів, публікує повідомлення про помилки під час інсталяції пакетів |
| **Virtual machine, VM** | *Віртуальна машина* | Частина Pharo, залежна від платформи, виконує байт-код програм |
| **Workspace** | *Робоче вікно* | Редактор текстів, можна використовувати для завантаження файлів, випробування фрагментів коду, зберігання до файлу |
| **World Menu** | *Головне меню* | Pharo – цілий світ на комп’ютері, тому його головне меню називається «Світ» |

Навчальне електронне видання

Стефан ДЮКАС, Джордана РАКІЧ,

Себастьян КАПЛАР, Квентін ДЮКАС

**PHARO 9 НА ПРИКЛАДАХ**

Навчальний посібник

Переклад з англійської   
з доповненнями

Сергій ЯРОШКО

Редактор *Н. Плиса*

Комп’ютерне макетування *С. Ярошко*

Фото на обкладинці *Джозефа Гегана*

Формат 60×84/8. Ум. друк. арк. 31,4.

Видавець та виготовлювач:

Львівський національний університет імені Івана Франка,

79000, Львів, вул. Університетська, 1

Свідоцтво про внесення суб’єкта   
видавничої справи до Державного реєстру видавців, виготівників   
і розповсюджувачів видавничої продукції.   
Серія ДК № 3059 від 12.12.2007 р.