

ОЛЕКСІЙ ВАСИЛЬЄВ

Програмування мовою РНР

Навчальний посібник

Київ
Видавництво Ліра-К
2022

УДК 004.774.6'41РНР
В19

Рецензенти:

Бичков Олексій Сергійович, доктор технічних наук, професор, завідувач кафедри програмних систем і технологій факультету інформаційних технологій Київського національного університету імені Тараса Шевченка.

Бойко Юрій Володимирович, кандидат фізико-математичних наук, доцент, завідувач кафедри комп'ютерної інженерії факультету радіофізики, електроніки та комп'ютерних систем, начальник Інформаційно-обчислювального центру Київського національного університету імені Тараса Шевченка.

Плескач Валентина Леонідівна, доктор економічних наук, кандидат технічних наук, професор, завідувач кафедри прикладних інформаційних систем факультету інформаційних технологій Київського національного університету імені Тараса Шевченка.

Васильєв Олексій

В19 Програмування мовою РНР: навч. посіб. Київ : Видавництво Ліра-К, 2022. 368 с.
ISBN 978-617-520-322-4

Книга, яку читач тримає в руках, присвячена мові програмування РНР. Ця мова особлива. Вона призначена не просто для програмування, а для веб-програмування. Тобто для створення проєктів, орієнтованих на використання в світовій мережі.

Книга містить найважливішу інформацію про основні синтаксичні конструкції РНР, принципи організації програм, особливості їх використання. У ній є багато прикладів, у тому числі й того, як програмні РНР-коди інтегруються у веб-документи. Діапазон тем, що розглядаються та обговорюються, досить широкий. Тому після опрацювання матеріалу книги читач зможе створювати ефективні програмні коди мовою РНР.

Книга буде корисною і цікавою програмістам-початківцям, студентам та викладачам відповідних спеціальностей, а також усім, хто захоплюється програмуванням та сучасними інформаційними технологіями.

УДК 004.774.6'41РНР

ISBN 978-617-520-322-4

© Васильєв О., 2022
© Видавництво Ліра-К, 2022

ЗМІСТ

Вступ. Книга про мову PHP

Що таке PHP.....	3
Клієнт і сервер.....	5
Як виконується код PHP.....	7
Програмне забезпечення.....	8
Про книгу.....	11
Про автора.....	13
Обернений зв'язок.....	13
Подяки.....	14

Розділ 1. Перша програма

Програмний код.....	15
Режим інтерпретатора.....	17
Режим сервера.....	25
Резюме.....	31

Розділ 2. Змінні та типи даних

Ознайомлення зі змінними.....	32
Основні типи даних.....	35
Арифметичні операції.....	39
Операції порівняння.....	41
Логічні операції.....	43
Побітові операції.....	45
Операції з текстом.....	49
Присвоєння значень.....	51
Тернарний оператор.....	52
Резюме.....	55

Розділ 3. Керуючі інструкції

Умовний оператор if.....	56
Оператор циклу while.....	64
Оператор циклу do-while.....	68
Оператор циклу for.....	71
Оператор вибору switch.....	75
Інструкція goto.....	83
Оператор вибору match.....	84
Резюме.....	89

Розділ 4. Масиви

Ознайомлення з масивами.....	90
Цикл за масивом.....	97
Багатомірні масиви.....	105
Присвоєння масивів.....	109
Об'єднання масивів.....	113

Порівняння масивів	115
Функції для роботи з масивами	117
Резюме	122

Розділ 5. Функції

Створення функцій	124
Результат функції	128
Тип аргументів та результату	133
Механізм передавання аргументів	136
Значення аргументів за замовчуванням	139
Довільна кількість аргументів	142
Рекурсія	150
Функція eval()	152
Анонімні функції	155
Іменовані аргументи	157
Резюме	161

Розділ 6. Корисні прийоми та операції

Посилання	163
Константи	168
Глобальні змінні	171
Статичні змінні	173
Багаторядковий текст	175
Робота з файлами	178
Додавання файлу в програму	184
Резюме	185

Розділ 7. Класи та об'єкти

Принципи ООП	187
Створення класів та об'єктів	190
Методи	193
Конструктор і деструктор	197
Статичні поля і методи	200
Копіювання об'єктів	202
Закриті поля і методи	205
Спеціальні методи	208
Визначення полів у конструкторі	219
Резюме	223

Розділ 8. Спадкування

Створення дочірнього класу	225
Переозначення методів	230
Конструктори і спадкування	234
Спадкування і закриті члени	237
Захищені члени класу	240
Віртуальні методи	241
Функція як значення поля	243
Багаторівневе спадкування	245
Резюме	250

Розділ 9. Розширені механізми ООП

Абстрактні класи	251
Інтерфейси	256
Спадкування інтерфейсів	258
Спадкування класів і реалізація інтерфейсів	260
Трейти	262
Контроль типу об'єкта	264
Простір імен	268
Анонімні класи	274
Резюме	276

Розділ 10. Оброблення помилок

Принципи оброблення винятків	278
Класи винятків	282
Генерування винятків	284
Користувацькі винятки	287
Оброблення винятків різних класів	292
Вкладені конструкції try-catch і блок finally	297
Повторне генерування винятків	301
Функції для оброблення помилок	303
Резюме	305

Розділ 11. Генератори та ітератори

Ознайомлення з генераторами	307
Функція-генератор з аргументами	310
Масив на основі генератора	312
Результат генератора	314
Особливості використання генераторів	317
Передавання значення в генератор	322
Ітератори	324
Резюме	328

Розділ 12. Використання PHP

Сценарій у HTML-документі	330
Оброблення параметрів запиту	337
Використання кнопки	341
Використання декількох кнопок	347
Використання списків і опцій	351
Слайдер та перемикачі	357
Резюме	363

Післямова

Що було і що буде	364
-------------------------	-----

Вступ

Книга про мову PHP

Я би і свідомість втратив, та ви мені і так повірите.

т/с "Альф"

Книга присвячена мові програмування PHP. Мова PHP – проста, красива і елегантна. А ще вона, у деякому сенсі, особлива. Справа в тому, що мова PHP призначена для виконання кодів на боці сервера. Іншими словами, мало знати мову PHP, необхідно ще й розуміти, як і для чого вона використовується. Це важлива обставина, оскільки розуміння того, навіщо потрібна мова, часто є ключовим фактором для її ефективного застосування.



До уваги

Можна сказати й так, що PHP – мова, призначена не просто для програмування, а для веб-програмування.

Що таке PHP

Мову PHP використовують для створення сайтів і веб-застосунків. Мова має доволі довгу історію, популярна серед розробників і підтримується більшістю хост-серверів.

Деталі



Хостинг – це послуга з надання ресурсів і місця на сервері для розміщення даних та інформації (наприклад, у вигляді веб-сторінки). Отже, хост-сервером називається сервер, на якому розміщена відповідна інформація. Для простоти і більшої конкретики під хост-сервером розумітимемо комп'ютер (сервер), на якому розміщена веб-сторінка користувача (тобто та сторінка, при роботі з якою планується використовувати PHP).

У мови PHP є автор – данський програміст Расмус Лердорф. Починався проект з розроблення скриптів для підтримки персональної веб-сторінки і мав робочу назву Personal Homepages Tools, або скорочено PHP Tools. Поступово все це

трансформувалось в повноцінний і ефективний програмний продукт. Нині назва PHP зазвичай асоціюється з фразою Hypertext Preprocessor, що не далеко від істини.



До уваги

На момент написання книги актуальною є версія PHP 8. Однак досвід свідчить про те, що остання версія мови далеко не відразу починає використовуватися на практиці. Тут простежується певна інертність, зумовлена як об'єктивними, так і суб'єктивними чинниками. Тому ми в основному розглядатимемо універсальні підходи, актуальні для декількох останніх версій мови. Цікавий також той факт, що шостої версії немає: після п'ятої відразу йде сьома. Причина в тому, що спроба випустити в світ шосту версію виявилася вкрай невдалою.

Мова PHP є скриптовою (сценарною) та інтерпретованою. Інтерпретованість означає, що програмний код виконується під керуванням спеціальної програми, яка називається інтерпретатором. Інтерпретатор зчитує порядково код програми і виконує відповідні інструкції.



Стандарт PHP 8

У версії PHP 8 анонсована поява JIT-компілятора (скорочення від Just in Time), призначеного для компілювання кодів PHP, що покликане підвищити швидкість виконання програм. При компіляції програмні інструкції переводяться в команди рівня процесора.

Сценарні мови зазвичай належать до високорівневих. Сценарії, на відміну від звичайних програм, як правило, містять інструкції для керування уже готовими програмними компонентами. Іншими словами, сценарна мова – це коли просто і зрозуміло. Хоча, безумовно, не завжди все так уже й райдужно.

Деталі



Синтаксис мови PHP відчув на собі вплив мови С. Тому читачі, які вивчали такі мови, як С, С++, С# та Java, знайдуть для себе багато знайомих синтаксичних конструкцій.

Безумовно, PHP можна розглядати як одну з мов програмування, яких на сьогодні доволі багато. Але все ж є особливість – хоча б у тому, як використовуються коди, написані мовою PHP.

Якщо маємо справу з будь-якою мовою програмування, то процес написання й використання програм виглядає приблизно так. Спочатку створюємо програмний код – простіше кажучи, пишемо програму. Потім цю програму необхідно виконати. Тут усе залежить від мови, але загальна схема пов'язана з тим, що програмний код компілюється або інтерпретується. При компіляції спеціальна програма-компілятор переводить програмний код у машинні інструкції (або дещо схоже на них), після чого ці інструкції виконуються. При інтерпретації спеціальна програма-інтерпретатор зчитує програмний код і виконує команди з цього коду. Але що б не відбувалося, важливо те, що все це ми в принципі робимо на одному й тому самому комп'ютері. На цьому комп'ютері і програму запускаємо, і результат отримуємо. З результатом виконання програми можемо робити все, що завгодно, але важливо те, що при цьому одного комп'ютера достатньо.

З мовою PHP справи дещо інакші. Щоб відчутти "родзинку", слід спочатку розглянути загальний механізм того, що відбувається у мережі у випадку звернення до веб-сторінки і яке місце в цьому процесі займає PHP.

Клієнт і сервер

Розглянемо у найбільш загальних рисах схему, відповідно до якої відбувається перегляд сайту в мережі. Головними діючими "персонажами" в цьому випадку виступають два комп'ютери: той комп'ютер, на якому хочемо переглянути веб-сторінку, і той комп'ютер, на якому ця сторінка фактично знаходиться. Перший комп'ютер (на якому намагаємося переглянути веб-сторінку) називається *клієнтом*, а комп'ютер, на якому веб-сторінка знаходиться, називається *сервером*. Між цими комп'ютерами існує зв'язок через глобальну мережу, і тому комп'ютери можуть обмінюватися інформацією. Ми працюємо на комп'ютері-клієнті і хочемо переглянути веб-сторінку. Для цього запускаємо спеціальну програму, призначену для перегляду веб-сторінок. Ця програма називається браузером. Ми відкриваємо браузер (наприклад, Chrome, Opera або Edge) і в адресному рядку вводимо адресу сайту, який хочемо переглянути. Браузер

робить запит на сервер. Сервер отримує запит, оброблює його, і повертає відповідь на комп'ютер клієнта. Фактично це код документа, котрий браузер має відобразити на екрані клієнта. Загальну схему такої "взаємодії" проілюстровано на рис. В.1.

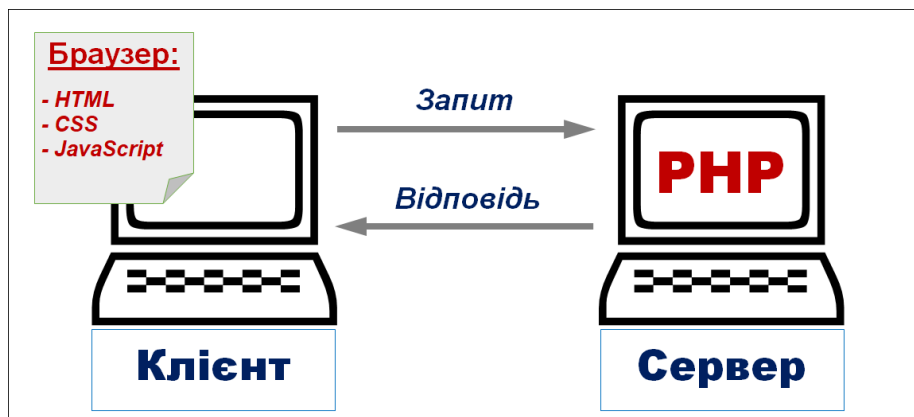


Рис. В.1. Схема взаємодії між клієнтом та сервером

Браузер оброблює документ, представлений у форматі HTML (від Hypertext Markup Language). Це текст, який містить спеціальну розмітку. Браузер цю розмітку "розуміє" і відображає документ відповідно до того, як передбачено вбудованими в документ інструкціями. Окрім власне розмітки HTML, документ, який відображає браузер, може містити й дещо більше. Наприклад, у документі може застосовуватися форматування, яке визначається каскадними таблицями стилів (аббревіатура CSS – від Cascade Style Sheets). Документ може містити сценарії мовою JavaScript. В останньому випадку браузер виконує ці сценарії. Тобто виходить, що сервер посилає набір команд, а браузер їх виконує. Важливо те, що виконуються команди на тому самому комп'ютері, який виконував запит.



До уваги

Простіше кажучи, сервер повідомляє, що потрібно робити, а браузер на комп'ютері-клієнті виконує необхідні операції. Зручно, але не завжди безпечно.

Де ж у цій схемі місце для PHP? Відповідь така: на етапі оброблення запиту сервером. Коли сервер отримує запит від клієнта, він оброблює запит, і в процесі оброблення можуть виконуватися сценарії – у цьому випадку написані мовою PHP.

Деталі



Часто результатом виконання сценарію є згенерований код HTML, який, власне, й передається клієнту.

Але й це ще не все. Багато програм намагаються обмінюватись інформацією через мережу. Процеси на клієнті посилають сигнали процесам на сервері, і навпаки, і потрібно знати, який сигнал призначений для якого саме процесу. Для цього використовують порти – спеціальні цілочислові ідентифікатори, за допомогою яких процеси "знаходять" призначені для них сигнали. Тому запити з браузера клієнта і відповіді сервера повинні бути синхронізовані за портом. Тобто для ефективної роботи з PHP потрібно вирішити багато технічних проблем. Усе це ми розглядатимемо поступово, за потреби.

Як виконується код PHP

Перейдемо до практичного боку вивчення PHP. А саме – спробуємо зрозуміти, як виконується програмний код PHP. Якщо йдеться про "природний" режим використання PHP-коду, то знадобився б сервер та клієнт. Тобто два комп'ютери. У цьому випадку сценарій (програма) розміщується на сервері, а переглядати результат виконання програми можна, звернувшись через браузер клієнта до веб-сторінки на сервері. Але навіть якщо всі ці ресурси є, то такий режим, м'яко кажучи, не дуже зручний, оскільки редагувати програмний код слід на одному комп'ютері (сервері), а перевіряти результат доведеться на іншому комп'ютері (клієнті). І навіть якщо в реальності не все так трагічно, зрозуміло, що потрібен надійніший підхід.

Альтернативна стратегія полягає в тому, щоб "обманути" браузер. Тобто створюється ілюзія, що сервером є сам клієнт. Йдеться про використання локального сервера. У такий режим перейти нескладно. Перевага підходу пов'язана з тим, що і програма, і її результат локалізовані в межах одного комп'ютера. При цьому програмний код мовою PHP можна виконувати за допомогою інтерпретатора – тобто приблизно так, як і у випадку з іншими інтерпретованими мовами. Мабуть, це не найпростіший спосіб побачити, який

результат виконання програми. Правда, не слід забувати, що пишуться програми мовою PHP зовсім не для того, аби виконуватись інтерпретатором на комп'ютері клієнта. Так що тут є свої тонкощі.

Програмне забезпечення

У книзі багато прикладів, і в процесі їх вивчення бажано не тільки аналізувати програмний код, а ще й досліджувати результат його виконання. Для цього знадобиться спеціальне програмне забезпечення.

По-перше, слід установити підтримку для PHP. З цією метою переходимо на сторінку `www.php.net`, як показано на рис. В.2.

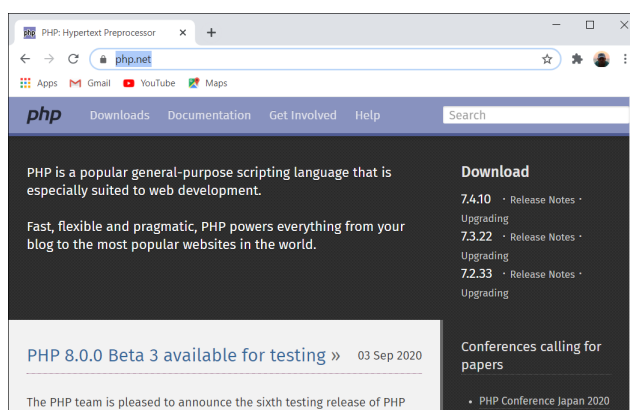


Рис. В.2. Вікно підтримки PHP за адресою `www.php.net`

У цьому вікні слід знайти розділ завантаження програмного забезпечення і скачати потрібні файли.



До уваги

У найпростішому варіанті встановлення зводиться до розпакування архіву, завантаженого із сайту `www.php.net`. Скоріш, цього буде достатньо для роботи з інтерпретатором PHP (файл `php.exe`) у режимі командного рядка. Для більш "комфортної" роботи, із залученням допоміжних програмних засобів, можливо, доведеться виконати додаткові налаштування. У цьому випадку слід скористатися довідниковою інформацією на сторінці `www.php.net` і довідкою з відповідного програмного продукту (наприклад, редактора кодів).

Деталі



Дізнатись версію PHP можна за допомогою інструкції `php -v` у командному рядку. Щоб отримати довідку з PHP, застосовують команду `php -h`. Детальнішу інформацію про PHP можна отримати за допомогою команди `php -i`.

Якщо використовується операційна система Windows, то для переходу в режим терміналу в адресному полі Провідника (Windows Explorer) можна увести інструкцію `cmd`. Потім у вікні терміналу слід перейти у каталог з PHP. Наприклад, якщо PHP знаходиться у папці `C:\PHP`, відповідна команда виглядатиме як `cd C:\PHP`. Альтернатива – спочатку перейти у каталог з PHP, а вже потім в адресному полі Провідника ввести інструкцію `cmd`.

У багатьох операційних системах сімейства Linux PHP встановлено за замовчуванням. Але якщо це не так, то для встановлення можна скористатися командою `sudo apt install php`.

Взагалі-то на цьому можна зупинитися. Серед завантажених утиліт має бути інтерпретатор PHP, що дозволить виконувати програмні коди, написані мовою PHP. Однак програмні коди слід десь набирати. В принципі, згодиться і звичайний текстовий редактор. Але краще встановити щось більш просунуте, з підтримкою синтаксису PHP (тобто редактор, який "розуміє" спеціальні інструкції PHP). Як варіант можна запропонувати редактор Atom (адреса www.atom.io). Вікно браузера, відкрите на сторінці підтримки проекту, представлене на рис. В.3.

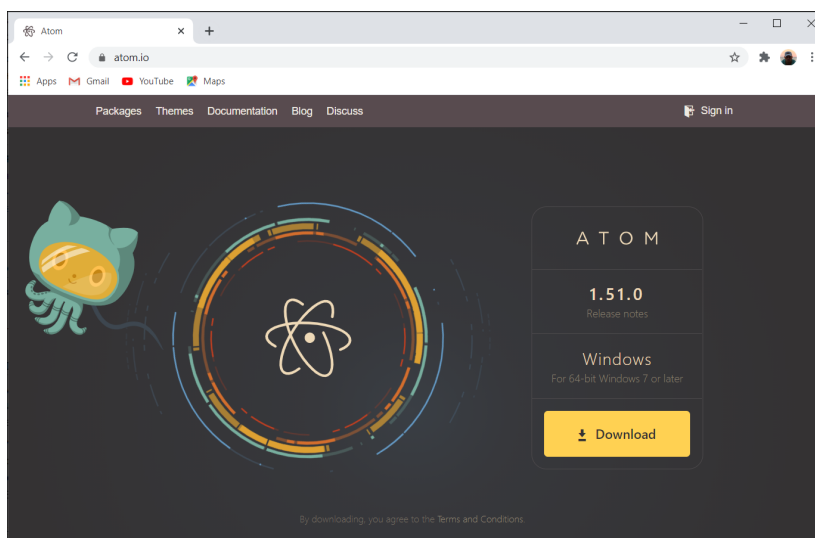


Рис. В.3. Сторінка www.atom.io завантаження редактора Atom

Але є й інші варіанти. Наприклад, досить зручне середовище розроблення Visual Studio Code (адреса <https://code.visualstudio.com>). Вікно браузера, відкрите на сторінці проекту, показано на рис. В.4.

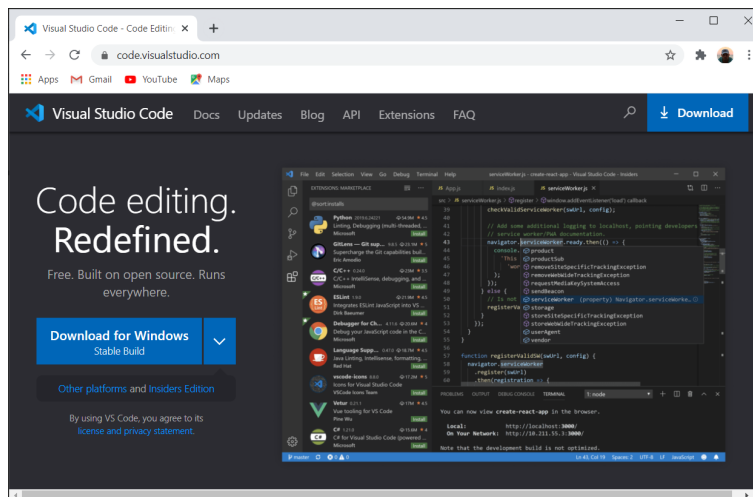


Рис. В.4. Сторінка підтримки проекту Visual Studio Code

Деталі



При першому запуску застосунку Visual Studio Code у розділі **Customize** слід підтвердити встановлення підтримки для PHP.

Ще один непоганий варіант для створення програм мовою PHP – середовище розроблення NetBeans. Установочні файли можна завантажити за адресою <http://netbeans.apache.org>. На рис. В.5 показано вікно браузера, відкрите на сторінці підтримки проекту NetBeans.

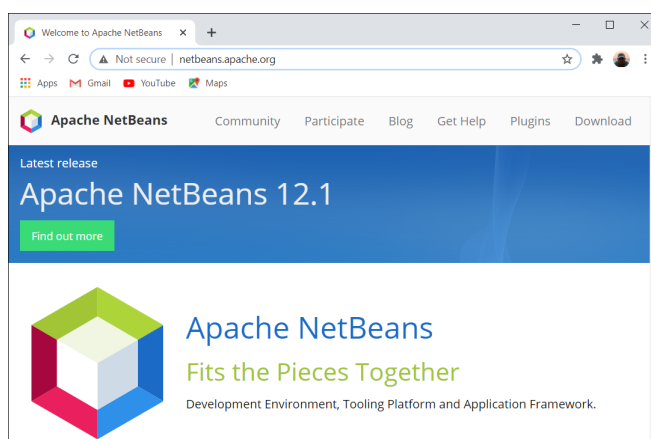


Рис. В.5. Сторінка підтримки проекту NetBeans

Існують й інші корисні програмні продукти, в тому числі й комерційні. Але ще раз підкреслимо, що в принципі досить установити PHP і підібрати пристойний редактор кодів.



До уваги

Методи роботи з програмним забезпеченням (у мінімальному обсязі) будемо вивчати під час розгляду прикладів.

Про книгу

Книга цілком присвячена мові PHP. Ми сфокусуємось на тому, що це за мова, навіщо вона потрібна і в чому її особливості. Розглянемо основні синтаксичні конструкції мови, а також підходи й механізми, які застосовуються в PHP. Значна увага приділяється прикладам. Діапазон обговорюваних питань достатній для того, аби після вивчення матеріалу книги писати ефективні програмні коди мовою PHP. Основна частина книги складається з дванадцяти розділів.

- У першому розділі наведено приклади простих програм, обговорено способи використання програмного забезпечення та проаналізовано загальні принципи створення програм мовою PHP.

- У другому розділі розглянуто змінні й типи даних. Там дізнаємось, як створюються змінні, які операції з ними виконуються і в чому особливості використання даних різних типів.

- Третій розділ присвячено керуючим інструкціям. Ми ознайомимося з умовним оператором, операторами вибору, операторами циклу, а також інструкцією безумовного переходу. Усі ці синтаксичні конструкції критично важливі для створення функціональних програм мовою PHP.

- Четвертий розділ містить інформацію про масиви. У мові PHP масиви мають низку унікальних властивостей, і саме такі особливості будуть предметом обговорення. У цьому розділі розглянуто одномірні й багатомірні масиви, описано оператор циклу за колекцією, а також проаналізовано основні операції, які виконуються з масивами.

• У п'ятому розділі описано функції. Ми дізнаємось, як вони створюються, обговоримо механізми передачі аргументів, способи повернення результату, навчимося задавати для аргументів значення за замовчуванням, створювати функції з довільною кількістю аргументів. Розглянуто спосіб передачі аргументів за іменем. Приділено увагу рекурсії, анонімним функціям і низці інших тем, що стосуються функцій.

• У шостому розділі розглянуто посилання, константи, глобальні й статичні змінні, робота з файлами і методи використання багаторядкових текстів.

• У сьомому розділі обговорено принципи об'єктно-орієнтованого програмування (ООП) і механізми реалізації цих принципів у PHP. Крім іншого, дізнаємось, як створюються класи та об'єкти, в чому особливість методів, ознайомимося з конструкторами й деструкторами, статичними членами класу. У розділі розглянуто задачу про копіювання об'єктів, проілюстровано особливості використання закритих полів та методів. Також описано деякі специфічні методи.

• Восьмий розділ присвячено такому важливому механізму ООП, як спадкування. Ми дізнаємось, як створюється дочірній клас і як переозначаються методи. Розглянемо особливості використання конструкторів, закритих і захищених членів класу при спадкуванні. Ще в розділі обговорено віртуальність методів, описано багаторівневе спадкування, розглянуто деякі інші теми.

• У дев'ятому розділі описано абстрактні класи та інтерфейси (в тому числі їх реалізацію і спадкування). Ми дізнаємося, що таке трейти, як за допомогою інтерфейсу можна контролювати тип об'єкта, що таке простір імен і як він використовується.

• Принципи оброблення помилок і винятків подано в десятому розділі. Ми ознайомимося з основними класами винятків, дізнаємось, як винятки генеруються. Крім цього, в розділі описано методи створення класів для користувацьких винятків. Висвітлено й деякі інші питання, що стосуються оброблення помилок та винятків.

- Одинадцятий розділ присвячено генераторам та ітераторам. Ми навчимося використовувати функції-генератори, застосовувати їх у різних ситуаціях. Ще дізнаємося, що таке ітератори та як їх створювати.

- У дванадцятому розділі подано приклади використання PHP-програм на практиці. Розділ містить детальний розбір декількох задач, котрі дають уявлення про роль кодів PHP при створенні веб-документів.

У кінці кожного розділу для зручності наводиться коротке *резюме*, в якому перераховуються всі основні моменти розділу. Це сприяє кращому засвоєнню матеріалу книги.



До уваги

Книга розрахована насамперед на тих, хто має мінімальну підготовку в програмуванні. Відповідно, матеріал за можливості подано максимально просто.

Про автора



Васильєв Олексій Миколайович, доктор фізико-математичних наук, професор кафедри теоретичної фізики Київського національного університету імені Тараса Шевченка. Автор книг із програмування та математичного моделювання. Сфера наукових інтересів: фізика рідин та рідких кристалів, синергетика, біофізика, економіка та математична лінгвістика.

Обернений зв'язок

Свої побажання та зауваження з приводу цієї та інших книг автора можна направляти за адресою oleksii@vasyliev.kyiv.ua. Конструктивна критика вітається. Разом з тим автор заздалегідь вибачається за те, що через об'єктивні причини не на всі листи зможе дати відповідь.

Більше інформації про автора та його книги можна знайти на сайті www.vasyliev.kyiv.ua.

Подяки

Користуючись нагодою, висловлюю щиру подяку читачам. Їхня цікавість до уже виданих книг є найкращим стимулом для написання книг нових. Також найкращі слова вдячності – моїм студентам. Саме вони стали першою аудиторією, на якій "тестувались" методики і приклади. Їхній допитливий розум і нестандартне мислення багато в чому сприяли тому, щоб книги ставали кращими.

Розділ 1

Перша програма

– Я вимагаю відновлення озонового шару Землі.

– Альф, до суботи ми не встигнемо.

т/с "Альф"

У цьому розділі ми створимо першу програму мовою PHP. А якщо конкретніше, то розглянемо простий програмний код і дізнаємось, як його можна виконати. Власне програмування буде не дуже багато, проте багато інформації, надто важливої для подальшої роботи з PHP.

Програмний код

Програма в PHP – це набір інструкцій, які виконує інтерпретатор. Ці інструкції мають бути оформлені правильно, а потім передані інтерпретаторові для виконання. Відповідно, в наші найближчі плани входить вирішення двох завдань:

- скласти програмний код;
- запустити програму на виконання.

Причому друге завдання на порядок складніше за перше.

Отже, приступимо до створення нашої першої програми мовою PHP. Для початку визначимося з тим, що саме наша програма повинна робити. Традиційно першою розглядають програму, яка виводить повідомлення. Ми теж вчинимо так. Наша програма відобразить у вікні виведення (у терміналі) повідомлення з привітанням. Код програми представлено у лістингу 1.1.

Лістинг 1.1. Перша програма

```
<?php
    print("Hello, PHP!");
?>
```