

Я хочу робити все те саме, що й ти.
Ти — мій кумир.

Телесеріал «Альф»

Серед мов програмування мова Java — найпопулярніша й найбільш затребувана. Ця книга про те, як програмувати мовою Java.

Особливості мови Java

Погано на цій планеті з почуттям гумору.

Т/с «Альф»

Історія мови Java почалась в 1990-х роках, коли групі інженерів компанії Sun Microsystems у межах проекту під назвою «Green» було поставлено завдання з розробки універсальної, компактної і незалежної від конкретної платформи мови програмування Oak, призначеної для використання в побутових пристроях. У процесі реалізації проекту змінилися не тільки основні пріоритети, але й назва мови програмування. Як би там не було, в 1995 році світ познайомився з мовою програмування Java.



З моменту появи першої версії Java було декілька оновлень платформи. На момент написання книги актуальною є версія Java 10. Однак слід мати на увазі, що версії Java 9 та Java 10 підтримуються лише для 64-розрядних операційних систем. Тому актуальною залишається і версія Java 8, яка дозволяє працювати з 32-розрядними системами. Приклади для книги підбиралися відповідно до принципу оберненої сумісності, щоб програмні коди виконувались у всіх версіях, починаючи з Java 8. Методи та прийоми,

характерні для версій Java 9 або Java 10, виділені коментарями та супроводжуються поясненнями.

Мова Java, хоч і не без труднощів, але завоювала своє «місце під сонцем». Сьогодні Java міцно утримує позиції найбільш затребуваної мови програмування. Успіху мови сприяв бурхливий розвиток Internet-технологій. Справа в тім, що для Java-програм характерним є високий ступінь універсальності й незалежності від апаратного забезпечення. Це важливо у випадку створення програмного забезпечення, орієнтованого на роботу в мережі, оскільки кінцеві користувачі використовують різні операційні системи і різне апаратне забезпечення. До того ж не останню роль відіграла застосовність мови Java для програмування всіляких мобільних пристроїв. Тому немає нічого дивного в тому, що значна доля комерційних і вільно поширюваних програм написані на мові Java. Відповідно, попит на програмістів, які працюють з мовою Java, стабільно високий, а загальні тенденції такі, що він залишиться високим і найближчим часом.

Універсальність програм, написаних мовою Java, базується на використанні *віртуальної машини*. Це такий специфічний «посередник», під керівництвом якого виконується байт-код, який створюється під час компіляції програми. Тут необхідні пояснення.

Після того, як програма написана, вона компілюється. Зазвичай в результаті компіляції програми створюється виконавчий файл з машинним кодом, який і виконується, коли необхідно виконати програму. Простіше кажучи, під час компіляції команди, зрозумілі для програміста, переводяться на «мову», зрозумілу для комп'ютера. Якщо мова йде про програму, написану мовою Java, то все відбувається схожим чином, але з деякими особливостями. Найважливіше те, що в результаті компіляції Java-програми створюється не машинний код, а проміжний *байт-код*. Це дещо середнє між машинним кодом і кодом програми. Якщо машинний код, як правило, виконується під управлінням операційної системи, то байт-код виконується під управлінням спеціальної програми, яка називається віртуальною машиною (або віртуальною Java-машиною). Зрозуміло, що таку програму на комп'ютер треба заздалегідь установити.

Виникає запитання: а в чому ж виграв від використання віртуальної машини і як усе описане впливає на універсальність кодів? Виграш у тому, що під час написання коду можна абстрагуватися від особливостей

операційної системи і апаратного забезпечення, які використовуються кінцевим користувачем. Ці особливості враховуються — але враховуються на рівні віртуальної машини. Саме віртуальна машина під час виконання байт-коду «бере до уваги» особливості операційної системи й апаратного забезпечення комп'ютера, на якому виконується програма.



Припустимо, існує програма, написана мовою C++. У процесі її компіляції створюється машинний код, який для різних операційних систем буде різним. Якщо компілюється програма, написана мовою Java, то створений в результаті байт-код не залежатиме від операційної системи, котра встановлена на комп'ютері — він буде одним і тим самим для різних операційних систем. Але віртуальна машина для кожної операційної системи своя. Відмінність в операційних системах «враховується», коли на комп'ютер установлюється віртуальна машина.

Описаний вище механізм, загалом, забезпечує високий ступінь універсальності програм, написаних на мові Java. Особливо це помітно у випадку створення програм із графічним інтерфейсом.



Забігаючи наперед, відмітимо, що у плані створення прикладних програм із графічним інтерфейсом мова Java особливо ефективна.

Існує ще один важливий аспект, який стосується мови Java, на який одразу звертаємо увагу. Мова Java — повністю *об'єктно-орієнтована* мова. Сказане означає, що для написання навіть найменшої і найпростішої програми доведеться описати, щонайменше, один *клас*. Це автоматично створює деякі труднощі в освоєнні премудростей мови Java. Особливо важко тим, хто не має досвіду програмування. Адже фактично відразу, з перших кроків, доводиться знайомитися з концепцією *об'єктно-орієнтованого програмування* (скорочено ООП), яка, слід визнати, не є тривіальною. Однак панікувати не варто — ми знайдемо спосіб донести необхідні відомості навіть до найбільш невідготовлених читачів. Головне, аби було бажання засвоїти мову Java.

Програмне забезпечення

— Кейт, мені потрібна твоя порада.
— Ось тобі моя порада: більше ніколи так не роби!

Т/с «Альф»

Якщо підійти до питання формально, то сама по собі мова Java — це набір правил, відповідно до яких формується програмний код. Однак програми пишуться задля того, щоб вони виконувались. А якщо так, то нам знадобиться спеціальне програмне забезпечення. Гарна новина в тому, що все необхідне програмне забезпечення можна отримати цілком вільно, просто і безкоштовно.



Зрозуміло, що існують і комерційні прикладні програми, призначені для написання програм на мові Java. Але для вирішення тих завдань, які ми ставимо перед собою, стандартного вільно поширюваного програмного забезпечення більш ніж достатньо.

Що ж нам знадобиться? У принципі, можемо обмежитись мінімальними засобами у вигляді пакета прикладних програм JDK (скорочення від *Java Development Kit* — засоби розробки Java). У склад пакета JDK, крім іншого, входить компілятор, усілякі бібліотеки, документація і виконавча система JRE (скорочення від *Java Runtime Environment* — середовище виконання Java) — фактично, віртуальна машина мови Java. Пакет прикладних програм JDK поширюється безкоштовно компанією Oracle (сайт компанії www.oracle.com).



Свого часу розробника мови Java, компанію Sun Microsystems, поглинула корпорація Oracle. Так що тепер підтримкою Java-технологій займається саме вона.

Ситуація така, що без пакета JDK нам не обійтися, але й обмежуватися тільки пакетом JDK не варто. Якщо обмежитися тільки пакетом JDK, то програмні коди доведеться набирати в текстовому редакторі, а компілювати програму доведеться «вручну» з командного рядка. Тому бажано використати *середовище розробки* (скорочено IDE від *Integrated Development Environment*).

Середовище розробки містить редактор кодів, наладник, який дозволяє в інтерактивному режимі відстежувати код на наявність синтаксичних помилок, набір інших утиліт, які дозволяють зробити процес написання, тестування і компіляції програм простим, зручним і дець навіть комфортним (наскільки це взагалі можливо). Простіше кажучи, середовище розробки має використовуватись — тим більше, якщо врахувати, що наявними є досить пристойні безкоштовно поширювані середовища розробки. Ми зупинимо свій вибір на середовищі розробки IntelliJ IDEA. Середовище поширюється безкоштовно, файли для його установки можна завантажити на сайті підтримки проекту www.jetbrains.com.

Далі коротко розглянемо, яке програмне забезпечення і звідки слід завантажити перед тим, як ми безпосередньо перейдемо до вивчення мови програмування Java.

Завдання наше просте:

- необхідно завантажити й установити пакет прикладних програм JDK;
- після встановлення JDK слід завантажити й установити середовище розробки IntelliJ IDEA.

Дії з завантаження й установки програмного забезпечення виконуються саме в тій послідовності, в якій вони перераховані вище.



Середовище розробки IntelliJ IDEA у процесі роботи з програмними кодами звертається до системи JDK. Якщо систему JDK встановити до установки IntelliJ IDEA, то всі налаштування середовища розробки, пов'язані з JDK, скоріш за все, будуть виконані автоматично. Якщо систему JDK встановлювати після установки середовища розробки IntelliJ IDEA, то налаштування середовища розробки доведеться виконувати самостійно (хоча це й не так складно).

Отже, в першу чергу встановлюємо пакет JDK, для чого попередньо з сайту компанії Oracle завантажуюмо установочні файли. На рис. В.1 показано вікно браузера, відкрите на сторінці `www.oracle.com`.

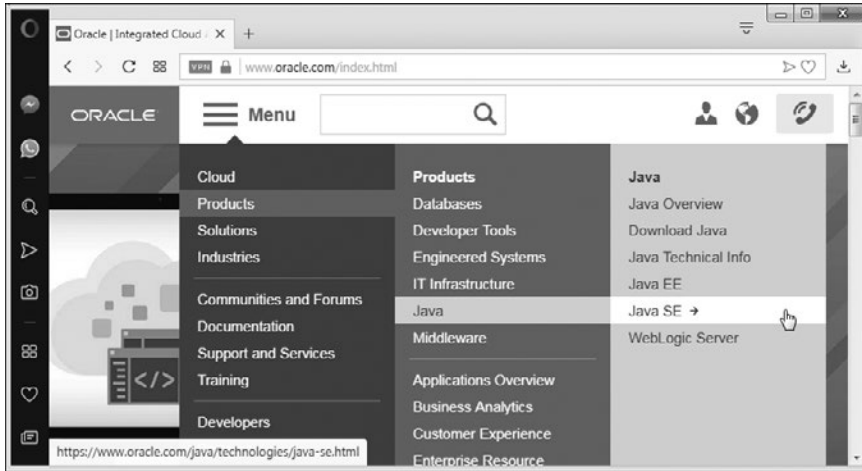


Рис. В.1. Вікно браузера, відкрите на сторінці `www.oracle.com` корпорації Oracle

На сайті потрібно знайти сторінку для завантаження програмного забезпечення для роботи з Java (як, наприклад, показано на рис. В.2).

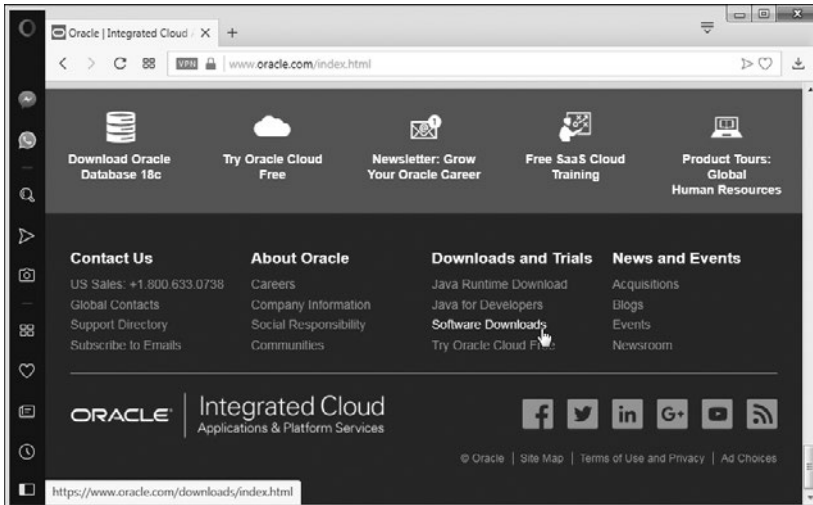


Рис. В.2. Гіперпосилання для завантаження програмного забезпечення для роботи з Java на сторінці `www.oracle.com`



Існує декілька редакцій для дистрибутивів Java. Наприклад, платформа Java для створення програмного забезпечення рівня великих корпорацій називається Java Enterprise Edition (скорочено Java EE). Стандартна редакція Java призначена для створення прикладних програм користувача і називається Java Standard Edition (скорочено Java SE). Також існує редакція Java Micro Edition (скорочено Java ME), яка використовується у випадку створення прикладних програм для усіляких мобільних пристроїв. Ми будемо використовувати стандартну редакцію Java Standard Edition (або Java SE).

Після кліку на гіперпосиланні для завантаження програмного забезпечення для роботи з Java переходимо до ще одної сторінки, з якої власне і виконується завантаження (рис. В.3).

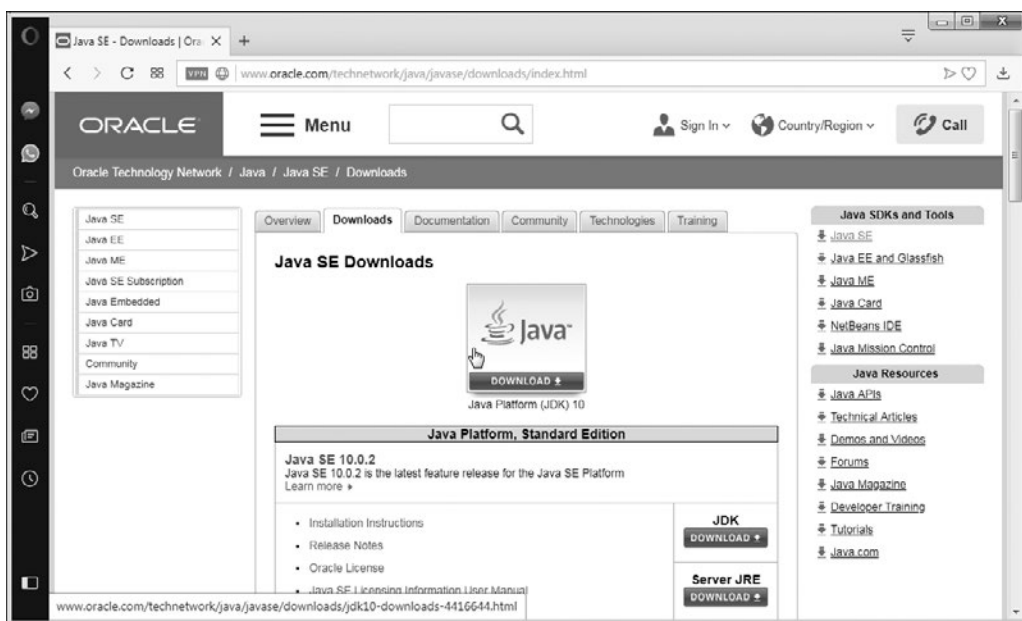


Рис. В.3. Вікно браузера, відкрите на сторінці завантаження установочного файлу пакета JDK

У процесі завантаження пропонується вибрати тип установочного файлу відповідно до того, яка операційна система використовується. Ситуація проілюстрована на рис. В.4.

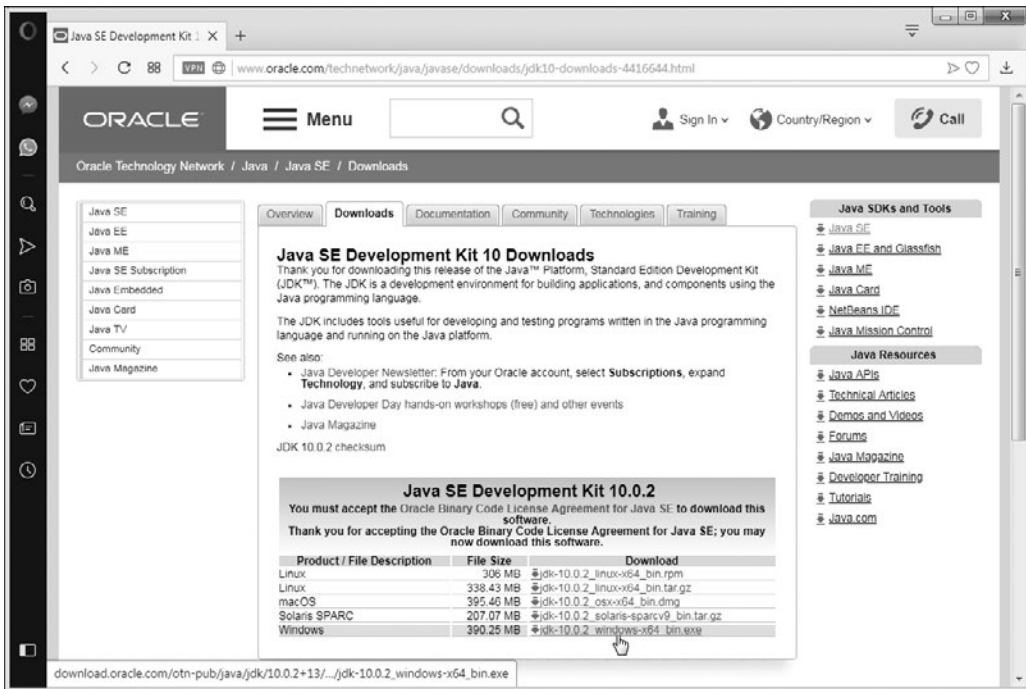


Рис. В.4. Вибір установочного файлу відповідно до того, яка операційна система використовується



Зовнішній вигляд сайтів, в тому числі й сайт корпорації Oracle, час від часу змінюється, тому не виключено, що для пошуку сторінки завантаження програмного забезпечення доведеться виявити деяку винахідливість.

Після завантаження й установки пакета JDK слід завантажити ще й установочний файл для середовища розробки IntelliJ IDEA. У цьому випадку переходимо на сторінку www.jetbrains.com проекту JetBrains, як показано на рис. В.5.

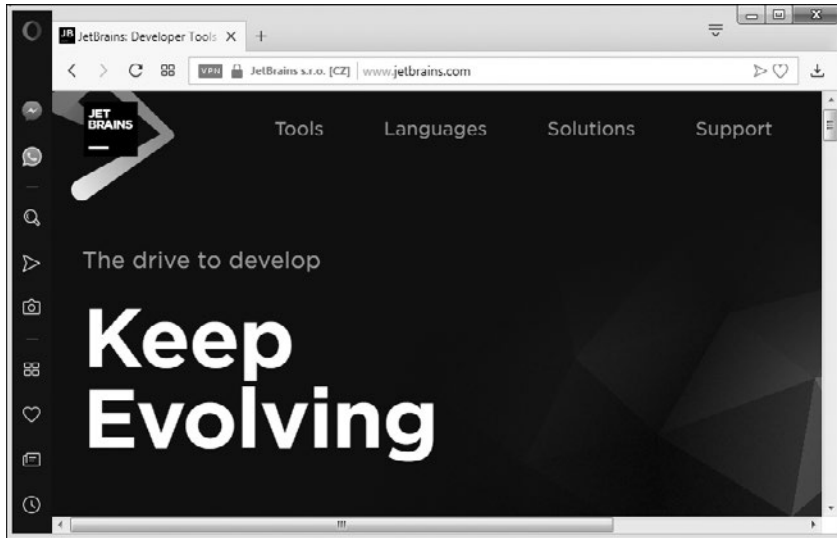


Рис. В.5. Сторінка www.jetbrains.com проекту Jet Brains

Потім переходимо на сторінку завантаження середовища IntelliJ IDEA, для чого вибираємо відповідний пункт у розділі **Tools**, як показано на рис. В.6.

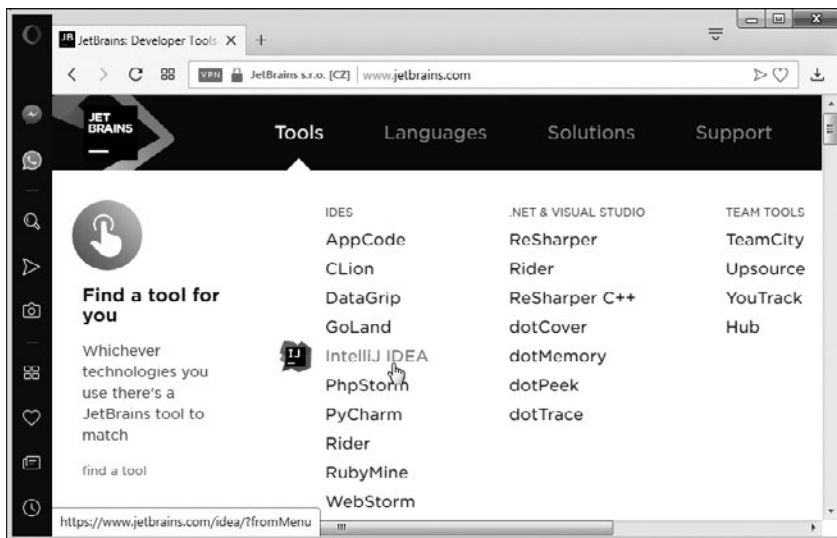


Рис. В.6. Перехід на сторінку для завантаження середовища розробки IntelliJ IDEA

Як виглядає сторінка, на якій можна завантажити установочний файл середовища розробки IntelliJ IDEA, показано на рис. В.7.

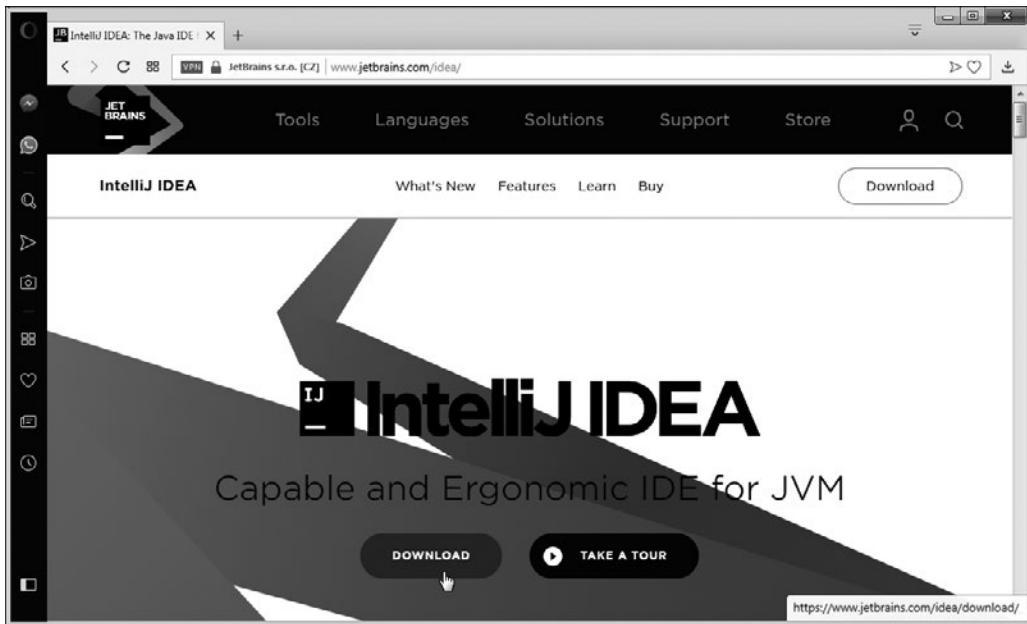


Рис. В.7. Сторінка www.jetbrains.com/idea для завантаження установочного файлу для середовища розробки IntelliJ IDEA

Завантажуємо установочний файл і встановлюємо середовище розробки IntelliJ IDEA. На цьому попередня підготовка до написання програм на Java завершується. Зазначимо лише, що ще одна корисна сторінка знаходиться за адресою www.java.com. На рис. В.8 показано вікно браузера, відкрите на цій сторінці.