

ПРАКТИЧНЕ ЗАНЯТТЯ № 1

Тема: Компіляція простого програмного коду Java

Мета: Навчитися використовувати інструменти Java для створення найпростішого програмного коду, та його компіляції.

Обладнання: комп'ютери Pentium, Celeron.

Програмне забезпечення: MS Windows XP, пакет Java SDK.

1. Для підготовки Java-програми підійде будь-який текстовий редактор. Істотною є тільки наявність в ньому підтримка довгих імен файлів. У простому випадку Java-додаток складається з одного java -файлу. Наберемо просту Java -програму (файл Hello.java) :

```
public class Hello {  
  
    public static void main(String[] args){  
        System.out.println("Hello World!");  
    }  
}
```

2. Відкомпілюємо програму за допомогою команди

```
javac.exe Hello.java
```

3. Тепер запустимо її на виконання

```
java.exe Hello
```

4. На екран буде виведено одне слово Hello .

Зауваження!

Зверніть увагу, що при трансляції програми задається ім'я файлу (з розширенням), а при виконанні ім'я файлу без розширення.

Що демонструє цей приклад

Цей приклад примітивний, але проте на цьому прикладі можна познайомитися з дуже важливими поняттями. Отже, що демонструє цей приклад.

- Як ми розглянемо детальніше пізніше, увесь програмний код в Java міститься усередині класів. Не може бути ніякого програмного тексту (за винятком декількох спеціальних директив) поза класом (чи інтерфейсу).

- Кожен файл з ім'ям Name.java повинен містити клас з ім'ям Name (причому, враховується регістр). Кожен public-клас з ім'ям Name має бути у своєму файлі Name.java .

- Усередині вказаного файлу можуть бути і інші класи, але їх імена повинні відрізнятися від Name і вони не мають бути public .

- Усередині класу може бути конструкція

```
• public static void main(String[] args){  
• . . .  
• }
```

Це метод класу. Тут public, static, void - це описувачі, main - ім'я методу.

- Вказаний метод main є спеціальним випадком. При запуску Java-програми ми вказуємо ім'я класу. Java-машина шукає цей клас серед усіх доступних їй файлів *.class, і в цьому класі запускає на виконання метод main .

- Опис методу main має бути в точності таким, як наведено в прикладі (можна хіба що змінити ім'я args на якесь інше).

- У дужках після імені методу вказуються параметри методу. Для main -метода параметри мають бути такими як вказано. В даному випадку String[] вказує, що як параметри методу буде переданий масив рядків. При виклику програми на Java можна задати параметри виклику. Java-машина обробить їх і сформує масив рядків, який буде переданий в main-метод як параметр. Так, якщо викликати програму командою:

```
j Hello one two 3 4
```

то усередині програми args буде масивом з 4-х елементів

```
args[0] = "one"  
args[1] = "two"
```

```
args[2] = "3"  
args[3] = "4"
```

Інша справа, що в даному випадку параметри виклику ніяк не використовуються нашою програмою - вона їх просто ігнорує.

5. Створити архів .jar для файлу Hello.class використовуючи утиліту jar.

Синтаксис має наступний вигляд: jar cf jar-файл вихідний-файл(и)

*Опція c вказує, що необхідно створити (create) JAR-файл.

*Опція f вказує, що необхідно перенапривити інформацію у файл, а не на стандартний потік виводу.

6. Переглянути створений архів архіватором WinRar.

7. Знайти в архіві файл маніфесту та переглянути його вміст.

* **Маніфест** - це спеціальний файл, який може містити інформацію про файли, зібрані в JAR-архів. Підгонкою цієї "мета"-інформації, що міститься в маніфесті, ви можете зробити JAR-файл придатним для використання в різноманітних цілях.

8. Створити файл .java з назвою, що відповідає назві класу Example2.

9. Ввести в нього програмний код:

```
import java.io.IOException;  
public class Example2  
{  
    public static void main(String[] args)  
    {  
        char inChar;  
        System.out.println("Input symbol");  
        try {  
            inChar=(char)System.in.read();  
            System.out.println("Your symbol is "+inChar);  
        }  
        catch (IOException e)  
        {System.out.println("Input ERROR");  
        }  
    }  
}
```

10. Провести його компіляцію та запустити на виконання.

11. Розібрати елементи програмного коду.

12. Оформити звіти та здати викладачу.