

ЛАБОРАТОРНА РОБОТА 8

Тема: Створення власного архетипу Maven

Теоретична частина

Архетипи в Maven є важливим інструментом для автоматизації створення проєктів, особливо рівня підприємства. Вони слугують шаблонами, які дозволяють швидко генерувати нові проєкти за налаштованою структурою, що відповідає найкращим практикам. Основним призначенням архетипів є:

- стандартизація структури проєктів, особливо в межах однієї організації або команди розробки;
- забезпечення швидкого старту робіт для розробників, завдяки надання готових зразків для ключових функцій;
- забезпечення можливості додавати частини проєкту до існуючих, наприклад, створювати сайти для проєктів за допомогою архетипу сайту.

Архетипи можуть розглядатись як певна "абстракція шаблону проєкту", яку можна адаптувати під конкретні потреби, що і забезпечує певний підхід у стандартизації розробки та прискорення процесу створення проєктів. Maven надає низку стандартних архетипів, кожен з яких призначений для певного типу проєкту:

Артефакт ID архетипу	Опис
maven-archetype-archetype	Шаблон для створення зразка архетипу
maven-archetype-j2ee-simple	Шаблон для спрощеного J2EE-додатку
maven-archetype-plugin	Шаблон для створення плагіна Maven
maven-archetype-quickstart	Шаблон для базового Maven-проєкту
maven-archetype-webapp	Шаблон для веб-додатку Maven

Ці архетипи можна використовувати для різних сценаріїв: створення простих Java-проєктів, вебзастосунків чи навіть плагінів для Maven. Наприклад, архетип maven-archetype-quickstart генерує базовий проєкт зі структурою, до якої входять каталоги src/main/java та src/test/java.

До переваг використання архетипів, крім швидкого старту та певного рівня стандартизації, слід також віднести гнучкість. Архетипи можуть бути "додатковими", тобто їх можна використовувати для додавання частин до існуючих проєктів, наприклад, створення сайту для проєкту за допомогою архетипу сайту.

Крім названих переваг архетипи також мають певні обмеження. Наприклад, вибір правильного архетипу ускладнюється через їх значну кількість, різною активністю підтримки та адаптованості до нових версій Java SE та Jakarta EE.

Архетипи особливо корисні в таких сценаріях:

- створення проєктів для навчання, коли необхідно отримати зразок для експериментів;
- стандартизація розробки великими командами, де важливо, щоб усі проєкти мали однакову структуру;
- генерація специфічних типів проєктів, таких як вебзастосунки чи плагіни, за допомогою відповідних архетипів.

Приклад команди Maven для створення базового проєкту:

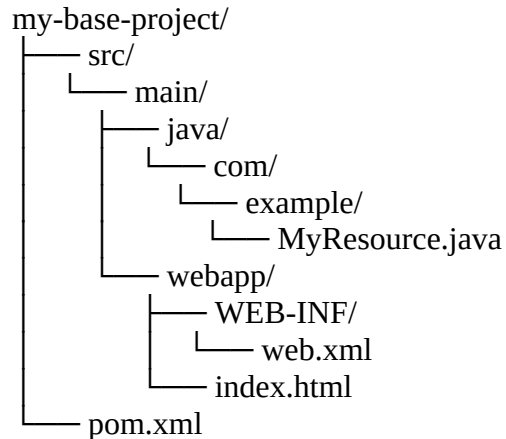
```
mvn archetype:generate -DarchetypeGroupId=org.apache.maven.archetypes \
-DarchetypeArtifactId=maven-archetype-quickstart
```

Практична частина

Розглянемо створення власного Maven archetype додатків Jakarta EE 10, як шаблону для швидкого старту проєктів.

1. Створення базового проєкту (шаблону)

Створіть новий Maven-проєкт зі структурою Jakarta EE 10. Приклад структури:



2. Налаштування pom.xml базового проєкту

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>jakartaee10-base</artifactId>
  <version>1.0.0</version>
  <packaging>war</packaging>

  <properties>
    <jakartaee.version>10.0.0</jakartaee.version>
    <maven.compiler.source>17</maven.compiler.source>
    <maven.compiler.target>17</maven.compiler.target>
  </properties>

  <dependencies>
    <!-- Jakarta EE Web Profile -->
    <dependency>
      <groupId>jakarta.platform</groupId>
      <artifactId>jakarta.jakartaee-web-api</artifactId>
      <version>${jakartaee.version}</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-war-plugin</artifactId>
        <version>3.3.2</version>
      </plugin>
    </plugins>
  </build>
</project>
```

3. Генерація archetype з базового проєкту

Виконайте в корені базового проєкту:

```
mvn archetype:create-from-project
```

Якщо при виконанні буде отримана помилка з причин відсутності файлу settings.xml, то його слід створити за вказаним шляхом і найпростішим вмістом:

```
<!-- settings.xml -->
<settings>
  <interactiveMode>false</interactiveMode>
</settings>
```

Після цього вищезгадувану команду Maven можна повторити.

4. Налаштування згенерованого archetype

Після виконання команди знайдіть згенерований archetype в:

target/generated-sources/archetype/

Тепер можна виконати редагування його файлів:

4.1. src/main/resources/META-INF/maven/archetype-metadata.xml додаємо фільтр файлів:

```
<fileSet filtered="true" packaged="true">
  <directory>src/main/java</directory>
  <includes>
    <include>**/*.java</include>
  </includes>
</fileSet>
```

4.2. src/main/resources/archetype-resources/pom.xml - замінюємо статичні значення на змінні:

```
<groupId>${groupId}</groupId>
<artifactId>${artifactId}</artifactId>
<version>${version}</version>
```

5. Встановлюємо archetype локально

Виконуємо в каталозі target/generated-sources/archetype/ наступну команду:

```
mvn clean install
```

У результаті в каталозі target/generated-sources/archetype/target/ повинен утворитись jar-файл, який буде вміщувати необхідний шаблон проєкту для використання при створенні нових проєктів:

```
./..
./classes
./maven-archiver
./test-classes
jakartaee10-web-base-archetype-1.0.0.jar
```

Крім того при виконанні команди `mvn install`, Maven розгортає артефакти — у даному випадку jar-файл, у локальний репозиторій. Локальний репозиторій зазвичай знаходиться у каталозі „.m2“ домашньої директорії користувача. Архетип — це особливий тип Maven-проєкту, який також зберігається у цьому репозиторії. Точний шлях до каталогу, де зберігається архетип, залежить від „groupId“, „artifactId“ та версії архетипу. Наприклад, якщо архетип має „groupId=com.example“, „artifactId=jakartaee10-web-base-archetype“ та версію „1.0.0“, то шлях буде таким: „~/m2/repository/com/example/jakartaee10-web-base-archetype/1.0.0/“.

```
~/m2/repository/com/example/jakartaee10-web-base-archetype/1.0.0
.n За назвою
/..
_remote.repositories
jakartaee10-web-base-archetype-1.0.0.jar
jakartaee10-web-base-archetype-1.0.0.pom
```

Тому після інсталяції архетип стає доступним для використання в інших проєктах через команду „mvn archetype:generate“, для чого необхідно вказати відповідні параметри „archetypeGroupId“, „archetypeArtifactId“ та „archetypeVersion“.

Якщо в налаштуваннях Maven, наприклад, у „settings.xml“ вказано інший локальний репозиторій, то шлях до створеного архетипу буде відрізнятися. Для видалення архетипу достатньо видалити відповідний каталог з локального репозиторію.

6. Використання archetype

Відкрийте новий проєкт на основі створеного archetype:

```
mvn archetype:generate \
  -DarchetypeGroupId=com.example \
  -DarchetypeArtifactId=jakartaee10-web-base-archetype \
  -DarchetypeVersion=1.0.0 \
  -DgroupId=ua.edu.znu.lab \
  -DartifactId=myNewWebProject \
  -Dversion=1.0.0-SNAPSHOT \
  -Dpackage=ua.edu.znu.lab.myNewWebProject \
  -DarchetypeCatalog=local \
  -DinteractiveMode=false
```

Такі параметри, як: DgroupId, DartifactId, Dversion та Dpackage можуть мати інші значення.

7. Перевірка роботи

Виконайте деплой створеного проєкту на сервер GlassFish 7 або WildFly 27+.

Завдання

1. Повторіть генерацію архетипу за наведеним у практичній частині прикладом.
2. Створіть власний архетип на основі проєкту із завдання лабораторної роботи 7.
3. Продемонструйте, що на основі створеного власного архетипу можна відкрити новий проєкт із встановленою структурою та вмістом.
4. Підготуйте звіт та створений архетип (jar-файл) до лабораторної роботи 7 в якості відповіді в Moodle.

Література

1. Maven Users Centre [Електронний ресурс] — <https://maven.apache.org/users/index.html>.