جافا - الوحدة الرابعة: الطرق وإعادة استخدام الشيفرة

الطريقة كتلة شيفرة لها اسم، نكتبها مرة واحدة ونستخدمها مرات كثيرة. وبلا طرق ينسخ الحساب نفسه إلى عدة أماكن، ويجب تكرار أي تعديل لاحق في كل مكان. وتجعل أربع خطوات الطرق تعمل: التعريف، وتمرير المدخلات، وإعادة النتيجة، وإبقاء الطريقة صغيرة. وفي هذه الوحدة تستخدمها رانيا، إدارية في مستشفى، لتحسب أجر العمل الإضافي في مكان واحد، وتجيب كل خطوة عن سؤال عملي واحد.

1. تعريف الطريقة

لتعريف الطريقة نكتب نوع القيمة المعادة، والاسم، وقائمة المعاملات بين قوسين، والجسم بين أقواس معقوفة. ويخبر نوع القيمة المعادة بما يعود إلى المستدعي، وتعني الكلمة void أن شيئا لا يعود. وينبغي أن يبدأ الاسم بفعل بنمط camel case، مثل calculateOvertimePay، ليقرأ الاستدعاء كأنه فعل. ونستخدم الطريقة باستدعائها باسمها، فتشغل جافا جسمها ثم تعود إلى السطر الذي يلي الاستدعاء. وتقع الطرق داخل صنف، وطريقة main التي عرفناها في الوحدة الأولى هي طريقة أيضا. وسؤال هذه الخطوة هو: ما المهمة الوحيدة لهذه الطريقة؟

مثال: تجد صاحبة متجر الأسطر الثلاثة نفسها لطباعة رأس الإيصال على شاشة الصندوق وفي الإيصال المرسل بالبريد وفي تقرير نهاية اليوم. فتنقلها إلى طريقة واحدة اسمها printReceiptHeader. ويستدعيها كل مكان الآن باسمها، ويظهر الرأس كما كان تماما. وعندما يتغير عنوان المتجر تعدل الطريقة مرة واحدة، فيظهر العنوان الجديد في كل إيصال.

في العمل: تجد رانيا حساب العمل الإضافي منسوخا في ثلاثة تقارير: التمريض والأشعة والمختبر. وتكتب طريقة واحدة هي `static BigDecimal overtimePay(BigDecimal hourlyRate, BigDecimal hours)` وتنقل الحساب إلى جسمها. وتستدعي التقارير الثلاثة الطريقة باسمها بدلا من أن يحمل كل واحد منها نسخته. وكانت قد كتبت مهمة الطريقة في جملة واحدة أولا: حساب أجر العمل الإضافي من سعر وساعات.

2. تمرير المدخلات

المعامل متغير في تعريف الطريقة، والوسيطة هي القيمة الفعلية التي نعطيها عند الاستدعاء. ويجب أن تطابق الوسائط المعاملات في العدد والترتيب والنوع. وتمرر جافا الوسائط بالقيمة، أي أن الطريقة تعمل بنسخة من كل قيمة. ومع الكائن تكون النسخة نسخة من المرجع، فتستطيع الطريقة تغيير محتوى الكائن، لكنها لا تستطيع جعل متغير المستدعي يشير إلى كائن آخر. ويسمح التحميل الزائد overloading لعدة طرق بالاسم نفسه إذا اختلفت قوائم معاملاتها، كنسخة للساعات ونسخة للساعات مع سعر إضافي. واختلاف نوع القيمة المعادة وحده لا يكفي للتمييز بين الطرق المحملة. وسؤال هذه الخطوة هو: ماذا تحتاج الطريقة أن تعرف، ولا شيء أكثر؟

مثال: تستدعي كاتبة رواتب طريقة اسمها formatName بالاسم الأول واسم العائلة، فتعيد الاسم الكامل مثل سارة ناصر. وتقبل طريقة ثانية بالاسم نفسه لقبا أيضا، فيعيد الاستدعاء بثلاث قيم الاسم مع اللقب. ويختار المترجم الطريقة الصحيحة من عدد الوسائط ونوعها في الاستدعاء. وهذا هو التحميل الزائد، فتتذكر الكاتبة اسما واحدا فقط.

في العمل: تحتاج طريقة العمل الإضافي إلى مدخلين: سعر الساعة وعدد ساعات العمل الإضافي. تمرر رانيا القسم أيضا في البداية، ثم تحذفه لأن الحساب لا يستخدمه أبدا. وتضيف نسخة ثانية بالاسم نفسه تأخذ سعرا إضافيا للمناوبات الليلية. والاستدعاء بوسيطتين يجد النسخة الأولى، والاستدعاء بثلاث وسائط يجد الثانية.

3. إعادة النتيجة

تنهي عبارة return الطريقة وترسل قيمة إلى المستدعي. ويجب أن تطابق القيمة النوع المعلن، وعلى الطريقة التي نوعها ليس void أن تعيد قيمة في كل مسار ممكن. ويستطيع المستدعي أن يحفظ النتيجة في متغير، أو يستخدمها في حساب، أو

حقوق الطبع والنشر محمية © - أكاديمية أساسيات (شركة أساسيات للخدمات الاستشارية)

يمررها مباشرة إلى طريقة أخرى. وتستدعى طريقة void من أجل ما تفعله، كالطباعة أو الحفظ، وليس من أجل قيمة. ولا تعيد الطريقة الواحدة إلا قيمة واحدة، فإذا اجتمعت عدة قيم نعيد كائنا نبنيه في وحدة لاحقة. وسؤال هذه الخطوة هو: ماذا يعود إلى المستدعي؟

مثال: تستقبل طريقة شحن وزن الطرد وتعيد رسوما. ويحفظ المستدعي الرسوم في متغير ويضيفها إلى الفاتورة. وتعيد الطريقة رسوما قليلة للطرد الخفيف ورسوما أعلى للثقيل، وينتهي كل فرع بعبارة return خاصة به. ولو نسيت الطريقة أن تعيد قيمة في فرع الطرد الثقيل لتوقف المترجم برسالة missing return.

في العمل: تعيد طريقة العمل الإضافي الأجر بالنوع BigDecimal، فيستطيع كل تقرير حفظه أو جمعه. وتضيف رانيا فحصا في الأعلى: عندما تكون الساعات صفرا أو أقل تعيد الطريقة صفرا فورا. وفي أول محاولة يرفض المترجم الشيفرة برسالة missing return، لأن أحد الفروع لا يحتوي على return. فتضيف return الأخيرة فتختفي الرسالة.

4. إبقاء الطريقة صغيرة

المتغيرات التي تعلن داخل طريقة لا توجد إلا أثناء عملها، ولا تراها الطرق الأخرى. ويسمى هذا النطاق scope، وهو يبقي الطرق مستقلة. والطريقة الجيدة تؤدي مهمة واحدة، وتتسع لها شاشة واحدة، واسمها يقول ما تفعله. والطرق الثابتة static تنتمي إلى الصنف نفسه ويمكن استدعاؤها دون إنشاء كائن، ولهذا تكون main ثابتة، وكثيرا ما تكون طرق المساعدة الصغيرة في الصنف نفسه ثابتة أيضا. ويصف تعليق قصير فوق الطريقة، بأسلوب Javadoc، ما تفعله ومعنى كل معامل وما تعيده. ويمكن اختبار الطرق الصغيرة واحدة واحدة، وهذا موضوع الوحدة الأخيرة. وسؤال هذه الخطوة هو: هل يفهم زميل هذه الطريقة من اسمها وتعليقها فقط؟

مثال: تجد مطورة طريقة واحدة تقرأ ملفا وتحسب ضريبة وتطبع تقريرا. وأي تغيير في قاعدة الضريبة يعني قراءة المهام الثلاث كلها. فتقسمها إلى ثلاث طرق هي readFile ثم calculateTax ثم printReport. ويمكن الآن تعديل كل منها وفحصها على حدة، فيمس تغيير الضريبة calculateTax وحدها.

في العمل: نمت شيفرة تقارير رانيا إلى طريقة main واحدة طويلة. فتستخرج أجر العمل الإضافي وإجمالي القسم والطباعة في ثلاث طرق منفصلة. وتضيف تعليق Javadoc قصيرا إلى طريقة العمل الإضافي يصف السعر والساعات والأجر المعاد. وعندما يغير المستشفى قاعدة العمل الإضافي تعدل طريقة واحدة فتصح التقارير الثلاثة.

الخاتمة

في الخلاصة، الطريقة كتلة شيفرة لها اسم، نكتبها مرة ونستدعيها مرات كثيرة. نعرفها بمهمة واضحة واحدة، ونمرر إليها ما تحتاج إليه فقط، ونعيد نتيجة من النوع المعلن، ونبقيها صغيرة. وتمرر الوسائط بالقيمة، ويسمح التحميل الزائد لاسم واحد بخدمة عدة قوائم معاملات، وتبقى المتغيرات داخل طريقتها. وقاعدة العمل الإضافي عند رانيا تقع الآن في مكان واحد، وتتبعها التقارير الثلاثة. وفي المرة القادمة سنخزن قيما كثيرة معا في المصفوفات والقوائم والخرائط.

المصادر

- Oracle: دروس جافا، تعريف الطرق – https://docs.oracle.com/javase/tutorial/java/javaOO/methods.html
- Oracle: دروس جافا، تمرير المعلومات إلى طريقة أو بانٍ – https://docs.oracle.com/javase/tutorial/java/javaOO/arguments.html
- Oracle: دروس جافا، فهم أعضاء الصنف (static) – https://docs.oracle.com/javase/tutorial/java/javaOO/classvars.html
- Oracle: دروس جافا، إرجاع قيمة من طريقة – https://docs.oracle.com/javase/tutorial/java/javaOO/returnvalue.html
- Oracle: دروس جافا، نظرة أقرب إلى تطبيق Hello World وطريقة main – https://docs.oracle.com/javase/tutorial/getStarted/application/index.html
- dev.java: تعلّم جافا – https://dev.java/learn/

- Java علامة تجارية لشركة Oracle و/أو الشركات التابعة لها. هذا الفيديو مادة تدريبية مستقلة وغير تابعة لـ Oracle.

حقوق الطبع والنشر محمية © - أكاديمية أساسيات (شركة أساسيات للخدمات الاستشارية)