جافا - الوحدة الثالثة: الشروط والحلقات

تصبح البرامج مفيدة عندما تتخذ قرارات وتكرر العمل. يشغل القرار مقطعا من الشيفرة فقط عندما يتحقق شرط، وتعيد الحلقة تشغيل الشيفرة مرارا إلى أن يتغير شرط. وتغطي ذلك أربع خطوات: القرار، والدمج، والاختيار، والتكرار. وفي هذه الوحدة يستخدمها طارق، مدير عمليات في متجر إلكتروني، ليمنح خصومات الولاء للطلبات الصحيحة، وتجيب كل خطوة عن سؤال عملي واحد.

1. القرار

يستخدم القرار الكلمة if وشرطا بين قوسين وكتلة شيفرة بين أقواس معقوفة. ويجب أن يكون الشرط قيمة من النوع boolean، فلا تقبل جافا عددا بدلا من صحيح أو خطأ. ونبني الشروط بمشغلات المقارنة: يساوي، ولا يساوي، وأكبر من، وأصغر من، ونسختي الأخيرين اللتين تشملان المساواة. وتعطي الكلمة else الشيفرة مسارا ثانيا، وتسمح else if باختبار عدة شروط بالترتيب، فنتوقف عند أول شرط متحقق. ونقارن النصوص بالطريقة equals لمحتواها، وليس بعلامتي المساواة. وسؤال هذه الخطوة هو: ماذا يجب أن يتحقق بالضبط قبل أن تعمل هذه الشيفرة؟

مثال: يفحص نظام حجز في فندق سؤالا واحدا بنعم أو لا لكل طلب: هل الغرفة التي طلبها الضيف خالية في هذه التواريخ؟ فإذا كان الجواب نعم أكد النظام الإقامة، وإلا اقترح أقرب تاريخ متاح، تماما كمسارين في if مع else. ولا يعمل إلا مسار واحد من الاثنين لكل طلب، فلا يؤكد الحجز ويرفض في الوقت نفسه.

في العمل: يكتب طارق قاعدته الأولى: إذا كان إجمالي الطلب فوق حد الخصم، يحصل الطلب على خصم بنسبة عشرة في المئة. ويظهر أول اختبار أن طلبا إجماليه عند الحد تماما لا ينال خصما، لأن القاعدة تقول فوق وليس عند أو فوق. فيغير المقارنة إلى أكبر من أو يساوي، وينجح الاختبار. وكان قد أجاب عن سؤال الخطوة بجملة واحدة قبل أن يكتب أي شيفرة.

2. الدمج

نادرا ما يكون للقاعدة الحقيقية شرط واحد، لذلك ندمج الشروط. ونستخدم المشغلات المنطقية التي نقرؤها and و or و not: and يحتاج إلى تحقق الشرطين معا، و or يحتاج إلى شرط واحد على الأقل، و not يعكس الشرط. وتقيم جافا من اليسار إلى اليمين وتتوقف بمجرد معرفة النتيجة، ففي and لا يقيم الطرف الأيمن إذا كان الأيسر خطأ بالفعل. وهذا مفيد للفحوص الآمنة، مثل التأكد من وجود قيمة قبل استخدامها. ومشغل and أقوى من مشغل or، لذلك توضح الأقواس القاعدة لكل قارئ. وتسهل قراءة القواعد المعقدة عندما يأخذ كل مقطع منها متغيرا من النوع boolean باسم ذي معنى. وسؤال هذه الخطوة هو: أي الشروط يجب أن تتحقق معا، وأيها بدائل؟

مثال: تصرف صيدلية مستشفى دواء فقط إذا كانت الوصفة صحيحة والمريض مسجل، وهذه قاعدة and تجمع شرطين. وإذا فشل أحد الفحصين فلا يصرف شيء، لأن and يحتاج إلى تحقق الشرطين معا. ويحفظ متغير من النوع boolean اسمه prescriptionIsValid القاعدة مقروءة، فيرى الصيدلي الشرط الأول بنظرة واحدة والثاني بجانبه.

في العمل: ينطبق خصم طارق عندما يكون العميل وفيا ويبلغ الإجمالي الحد، لكن لا ينطبق أبدا على طلب إرجاع. يكتب القاعدة أولا في سطر واحد طويل ولا يستطيع قراءتها بعد أسبوع. فينشئ ثلاثة متغيرات منطقية بأسماء واضحة: isLoyal و reachesThreshold و isReturn. وتقرأ القاعدة الآن كجملة تقريبا، ويراجعها زميل في دقيقة.

3. الاختيار

عندما تحدد قيمة واحدة الاختيار بين نتائج كثيرة، تصير سلسلة else if طويلة. وعبارة switch هي الأداة الأنظف. فهي تنظر في قيمة واحدة، مثل فئة العميل أو طريقة الشحن، وتقفز إلى الحالة المطابقة. وفي الصيغة الحديثة بالسهم تقف كل حالة وحدها، ويستطيع switch أن يعيد قيمة نحفظها في متغير. وتغطي الحالة default كل قيمة لم نذكرها، فلا يهمل

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

شيء بصمت. وفي الصيغة الأقدم بالنقطتين يجب أن تنتهي كل حالة بالكلمة break، وإلا انتقل التنفيذ إلى الحالة التالية، وهذا مصدر معروف للأخطاء. وسؤال هذه الخطوة هو: هل تحدد قيمة واحدة الاختيار بين نتائج كثيرة؟

مثال: تحدد شركة توصيل رسوم التسليم حسب مستوى الخدمة: عادية أو سريعة أو في اليوم نفسه، فقيمة واحدة هي المستوى تحسم الأمر بين ثلاث نتائج. ويربط switch واحد كل مستوى برسمه، وتقف كل حالة وحدها برسمها. ويضاف مستوى جديد بحالة واحدة أخرى، وتتعامل الحالة default مع أي مستوى غير معروف، فلا يبقى طلب بلا رسم.

في العمل: لمتجر طارق أربع فئات من العملاء: برونزية وفضية وذهبية وبلاتينية، ولكل منها نسبة خصم خاصة. ويستبدل بسلسلة else if الطويلة عبارة switch واحدة تعيد النسبة. ويضيف حالة default تعيد عدم الخصم، فلا تنال فئة غير معروفة شيئا ببساطة. ويتسع المنطق كله الآن لشاشة واحدة.

4. التكرار

تكرر الحلقة الشيفرة. ونستخدم الحلقة for عندما نعرف عدد الدورات، وفيها بداية وشرط وخطوة. وتفحص الحلقة while شرطها قبل كل دورة وتعمل ما دام صحيحا، أما do while فتفحص بعد الدورة، فتعمل مرة واحدة على الأقل. وتزور حلقة for المحسنة، وتسمى أيضا for each، كل عنصر في مجموعة أو مصفوفة دون عداد، فتزول أخطاء العد. وداخل الحلقة تخرجنا break منها فورا، وتنتقل continue إلى الدورة التالية. والحلقة التي لا يصير شرطها خطأ أبدا تعمل إلى الأبد، لذلك تحتاج كل حلقة إلى ما يغير شرطها. وسؤال هذه الخطوة هو: ماذا يتكرر، وما الذي يوقفه؟

مثال: يجب أن يعالج نظام رواتب كل موظف في قائمة، لذلك تتكرر الخطوات نفسها مرة لكل شخص. وتزور حلقة for each كل موظف بدوره، وتأخذ بياناته وتحسب راتبه دون أي عداد. وعندما تنتهي القائمة تتوقف الحلقة من تلقاء نفسها، وهكذا تتكرر لكل موظف وتقف عند نهاية القائمة.

في العمل: يتسلم طارق قائمة طلبات اليوم، وسنتعرف إليها جيدا في وحدة لاحقة، ويكتب حلقة for each عليها: `for (Order order : orders)`. ويضع داخلها القرار والقاعدة المدموجة وعبارة switch من الخطوات الثلاث الأولى. ويستخدم continue لتخطي طلبات الإرجاع، ويحتفظ بمجموع تراكمي للخصومات الممنوحة. ويظهر التقرير عند نهاية الحلقة كم طلبا نال خصما وكم كلف ذلك.

الخاتمة

في الخلاصة، يتخذ البرنامج قرارات بالكلمتين if و else، ويدمج الشروط بمشغلات and و or و not، ويختار بين نتائج كثيرة بعبارة switch، ويكرر العمل بالحلقات. ويجب أن يكون كل شرط من النوع boolean، وتقارن النصوص بالطريقة equals، وتحتاج كل حلقة إلى طريقة للتوقف. وتعمل قواعد طارق الآن على طلبات اليوم كلها، وتعطي كل عميل وفي الخصم الصحيح. وفي المرة القادمة سنضع هذا النوع من المنطق في طرق يمكن إعادة استخدامها واختبارها. ومن العادات الجيدة اختبار حواف القاعدة: القيمة التي تقل عن الحد قليلا، والتي تساويه تماما، والتي تزيد عليه قليلا.

المصادر

- أوراكل: دروس جافا، عبارات التحكم في التدفق – https://docs.oracle.com/javase/tutorial/java/nutsandbolts/flow.html
- أوراكل: دروس جافا، مشغلات المساواة والمقارنة والشروط – https://docs.oracle.com/javase/tutorial/java/nutsandbolts/op2.html
- أوراكل: دروس جافا، عبارة switch – https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html
- dev.java: تعلّم جافا – https://dev.java/learn/
- OpenJDK: تعبيرات switch (صيغة السهم) – https://openjdk.org/jeps/361
- 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)