# جافا - الوحدة الثامنة: الاستثناءات والعمل مع الملفات

تواجه البرامج الحقيقية مدخلات سيئة وملفات مفقودة وقيما غير متوقعة. والاستثناء هو طريقة جافا في الإبلاغ عن مثل هذه المشكلة أثناء عمل البرنامج، وحسن التعامل معه هو الفرق بين انهيار ورسالة مفيدة. وتغطي ذلك أربع خطوات: الالتقاط، والتصنيف، والتنظيف، وقراءة الملف. وفي هذه الوحدة تستخدمها دينا، محللة إيرادات في مجموعة فنادق، لتقرأ ملف حجوزات فيه بعض الصفوف السيئة، وتجيب كل خطوة عن سؤال عملي واحد.

## 1. الالتقاط

عندما يحدث خطأ تنشئ جافا كائن استثناء وترميه، فيتوقف التدفق العادي. ثم تبحث جافا صعودا في سلسلة الاستدعاءات عن كتلة catch مناسبة، وإذا لم تجد أي كتلة انتهى البرنامج وطبع stack trace. ونحرس الشيفرة الخطرة بالكلمة try، ونضع المعالجة في catch مع تسمية نوع الاستثناء الذي نستطيع حقا معالجته. وقد تتبع try الواحدة كتل catch عديدة، يأتي فيها النوع الأكثر تحديدا أولا. وكتلة catch الفارغة التي تبتلع المشكلة بصمت هي أسوأ عادة، لأن الفشل يختفي حينئذ: فنسجل الرسالة على الأقل أو نترك الاستثناء ينتقل. وسؤال هذه الخطوة هو: أي خطأ يمكن أن يحدث هنا، وما رد الفعل المعقول؟

**مثال:** يطلب متجر ويب من الزبون كتابة كمية ويحولها إلى عدد. ويكتب زبون حروفا بدلا منها، فيرمي التحويل استثناء. ويحرس المتجر التحويل بالكلمة try، وتعرض كتلة catch رسالة ودية وتسأل مرة أخرى. ولأن catch تعالج هذا الاستثناء فقط، يستمر المتجر في العمل.

**في العمل:** يقرأ برنامج دينا كل سطر في ملف الحجوزات ويحول حقل السعر إلى BigDecimal. وفي السطر الأربعمئة يكون السعر الكلمة pending، فيرمي التحويل NumberFormatException. فتضع التحويل داخل try وتلتقط هذا النوع بالتحديد. وتسجل كتلة catch رقم السطر والرسالة، وينتقل البرنامج إلى السطر التالي.

## 2. التصنيف

تأتي الاستثناءات في عائلتين. فالاستثناءات المفحوصة checked، مثل IOException، مشاكل خارجة عن سيطرتنا كملف مفقود، ويجبرنا المترجم إما على التقاطها وإما على إعلانها بالكلمة throws في توقيع الطريقة. أما الاستثناءات غير المفحوصة unchecked، التي ترث من RuntimeException، مثل IllegalArgumentException و NumberFormatException، فتشير عادة إلى خطأ في البرنامج أو في البيانات، ولا يجبرنا المترجم على معالجتها. والأخطاء Errors، مثل نفاد الذاكرة، ليست شيئا يحاول برنامج عادي التعافي منه. ونستطيع رفع استثناء خاص بنا بالكلمة throw مع رسالة واضحة، كأن تستقبل طريقة عدد ليال سالبا. والطريقة التي لا تستطيع أداء مهمتها ينبغي أن ترمي استثناء بدلا من أن تعيد قيمة خاطئة صامتة. وسؤال هذه الخطوة هو: هل تعالج هذه الطريقة المشكلة، أم تمررها إلى مستدع يعرف أكثر؟

**مثال:** تستقبل طريقة رواتب عدد ساعات سالبا. ولا تستطيع حساب راتب عادل من هذه القيمة، فترمي IllegalArgumentException بالرسالة hours must not be negative. وهذا استثناء غير مفحوص يشير إلى بيانات سيئة، فلا يجبر المترجم أحدا على معالجته. ويرى المستدعي المشكلة فورا بدلا من طباعة راتب خاطئ.

**في العمل:** تعلن طريقة دينا التي تفتح ملف الحجوزات throws IOException، لأن الملف المفقود ليس شيئا تستطيع إصلاحه. وتلتقط طريقة main الاستثناء وتطبع رسالة واضحة باسم الملف. وترمي طريقتها التي تنشئ حجزا من سطر IllegalArgumentException عندما يكون عدد الليالي أقل من واحد. وتتعامل كل طبقة من برنامجها الآن مع المشاكل التي تفهمها.

## 3. التنظيف

تحتاج بعض الموارد إلى إغلاق بعد الاستخدام، مثل الملفات واتصالات الشبكة وجلسات قواعد البيانات. وتعمل كتلة finally سواء حدث استثناء أم لا، وهذا يجعلها مكانا للتنظيف. والطريقة الحديثة والأكثر أمانا هي try with resources: نعلن


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

المورد بين قوسين بعد try، فتغلقه جافا تلقائيا حتى لو رمي استثناء. والموارد التي يمكن استخدامها هكذا تنفذ الواجهة AutoCloseable، وهي فكرة العقد من الوحدة السابقة. وعندما يفتح عدة موارد معا فإنها تغلق بالترتيب العكسي. ونسيان إغلاق ملف قد يبقيه مقفولا أمام برامج أخرى. وسؤال هذه الخطوة هو: ماذا يجب أن يحرر عندما ننتهي، حتى لو فشل شيء؟

**مثال:** تفتح أداة تقارير اتصالا بقاعدة بيانات لقراءة المبيعات. وتعلن الأداة الاتصال بين قوسين بعد try، وهو ينفذ الواجهة AutoCloseable. وفي نهاية الكتلة تغلقه جافا تلقائيا. وحتى لو فشل الاستعلام في منتصفه يغلق الاتصال أيضا، فلا يبقى أي اتصال مفتوحا.

**في العمل:** تفتح النسخة الأولى عند دينا قارئا على ملف الحجوزات وتنسى إغلاقه عندما يفشل سطر. فيجد زميل الملف مقفولا ولا يستطيع الكتابة فوقه. فتعيد كتابة شيفرة القراءة بأسلوب try with resources. ويغلق الملف الآن في كل الحالات، سواء نجح التقرير أم فشل.

## 4. قراءة الملف

يقرأ الصنف Files في الحزمة java.nio.file الملفات كاملة ويكتبها باستدعاء واحد. وتبني Path.of مسارا من اسم ملف. وتعيد الطريقة readAllLines قائمة فيها نص لكل سطر، وترمي IOException، فهي تحتاج إلى معالجة. ومع الملفات الكبيرة جدا نقرأ سطرا بعد سطر، فلا يبقى الملف كله في الذاكرة. وتفصل الطريقة split عند الفاصلة حقول ملف CSV بسيط، لكن الحقول التي تحتوي فواصل داخل علامات اقتباس تحتاج إلى مكتبة CSV مناسبة. ولكتابة النتائج تحفظ الطريقتان write و writeString النص في ملف باستدعاء واحد. وسؤال هذه الخطوة هو: ما شكل كل سطر، وماذا نفعل بسطر لا يطابقه؟

**مثال:** يتلقى محلل لوجستيات قائمة شحنات مصدرة في ملف نص. ويعيد استدعاء واحد للطريقة readAllLines كل سطر، ثم نفصل كل سطر عند الفاصلة لنبني من حقوله كائن شحنة. والسطر الذي ينقصه حقل لا يطابق الشكل المتوقع، فنكتبه في قائمة مشاكل منفصلة مع رقمه.

**في العمل:** تقرأ دينا الملف باستدعاء واحد يعيد كل الأسطر: `List<String> lines = Files.readAllLines(Path.of("bookings.csv"));`. وتفصل كل سطر عند الفاصلة، وتبني حجزا من الحقول، وتلتقط استثناءات الخطوات السابقة. وتذهب الحجوزات الجيدة إلى التقرير، وتذهب السطور السيئة إلى قائمة مشاكل مع أرقامها. ويكتمل التقرير الشهري، ويسجل ملف قصير الصفوف الثلاثة التي تحتاج إلى تصحيح.

## الخاتمة

في الخلاصة، الاستثناء هو طريقة جافا في الإبلاغ عن مشكلة أثناء عمل البرنامج. نلتقط فقط ما نستطيع معالجته، ونصنف المشاكل إلى مفحوصة وغير مفحوصة، وننظف بأسلوب try with resources. ويقرأ Files الملفات ويكتبها كاملة، وينبغي تسجيل السطر السيئ بدلا من تركه يوقف كل شيء. وتقرير دينا الشهري ينجو الآن من صف سيئ ويخبرها بالضبط أي الصفوف تحتاج إلى تصحيح. وفي المرة القادمة سنجمع كل شيء في تطبيق أعمال صغير. ومن العادات الجيدة كتابة رسائل خطأ تذكر الملف والسطر والقيمة.

**المصادر**


- أوراكل: دروس جافا، الاستثناءات – https://docs.oracle.com/javase/tutorial/essential/exceptions/index.html
- أوراكل: دروس جافا، عبارة try-with-resources – https://docs.oracle.com/javase/tutorial/essential/exceptions/tryResourceClose.html
- أوراكل: دروس جافا، قراءة الملفات وكتابتها وإنشاؤها – https://docs.oracle.com/javase/tutorial/essential/io/file.html
- أوراكل: دروس جافا، شرط الالتقاط أو الإعلان – https://docs.oracle.com/javase/tutorial/essential/exceptions/catchOrDeclare.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)