

التطبيقات الموزعة – جلسة 3 – مراجعة جافا 2

أهداف الجلسة

هذه الجلسة تنتم للجلسة السابقة لمراجعة المزيد من أساسيات لغة البرمجة جافا.

من المتوقع في نهاية الجلسة أن يكون الطالب قادر على:

- التعامل مع الصفوف التي تمثل الأنواع الأولية في جافا هي Number, Character, String.
- استخدام المصفوفات و الاستثناءات في جافا.
- استخدام الواجهات و الحزم Interfaces and Packages.

استخدام الصف Number

الصف Number هو صف مجرد ترث منه عدة صفوف تمثل أنواع البيانات الأساسية في لغة جافا التي تمثل الأعداد و هي Integer, Byte, Short, Long, Float and Double.

يمكن استخدام الكائنات من هذه الصفوف بنفس الطريقة التي نستخدم فيها المتحولات العادية و أيضاً توجد عدة طرائق معرفة في هذه الصفوف هي:

xxxvalue() تحويل العدد إلى النوع xxx مثلاً intValue() تحويل إلى int...

equlas() التحقق فيما إذا كان هناك كائنين من الصف Number لهما نفس القيمة.

parseXxx() طريقة static تستخدم لتحويل النمط String إلى نوع ما مثلاً Integer.parseInt لتحويل نص إلى عدد صحيح.

استخدام الصف Character

يستخدم هذا الصف لتمثيل محارف مفردة من النوع char, يمكن إسناد قيمة char إلى الكائنات من الصف Character, لاستخدام بعض المحارف الخاصة نستخدم \ مثلاً \n تعني سطر جديد, \t تعني حرف tab

أهم الطرائق الستاتيكية هي:

isLetter() ترد القيمة true إذا كان المحرف هو حرف.

isDigit() ترد القيمة true إذا كان المحرف هو رقم.

isUpperCase() و isLowerCase() إذا كان المحرف حرف كبير أو صغير.

toLowerCase() و toUpperCase() لتحويل محرف صغير إلى كبير أو محرف كبير إلى صغير.

استخدام الصف String

الصف String يمثل مصفوفة من المحارف chars يتم التعامل معها على أنها كائن واحد, يمكن تعريف سلسلة نصية عن طريق إشارات الإقتباس المزدوجة " أو عن طريق تعرف مصفوفة محارف.

أهم الطرائق المستخدمة:

length() تعيد عدد المحارف ضمن السلسلة النصية.

يمكن استخدام المعامل + لجمع سلسلتين مع بعضهما البعض, كما يمكن استخدام الطريقة concat أيضاً.

تستخدم الطريقة format لتوليد لسلية نصية جديدة من تنسيق معين كما يلي:

```
String s = String.format("My name is %s, I am %d years old", "mouhsen", 28); // now s is "My name is mouhsen, I am 28 years old"
```

الطريقة `charAt()` تستخدم للحصول على حرف ما عند دليل معين.

الطريقة `indexOf()` ترد ليل محرف أو سلسلة نصية ضمن السلسلة النصية.

الطريقتين `startsWith()` و `endsWith()` تستخدمان لتحديد فيما إذا كان النص يبدأ بسلسلة معينة أو ينتهي بها.

الطريقة `toCharArray()` تحول النص إلى مصفوفة محارف.

الطريقة `trim()` تحذف جميع الفراغات من بداية و نهاية النص.

الطريقة `substring()` تعيد نص جزئي من السلسلة النصية.

الطريقة `replace()` تستخدم لاستبدال نص ما بنص آخر.

الطريقة `getBytes()` تعيد مصفوفة بايتات من النص.

المصفوفات

تستخدم المصفوفات لتخزين مجموعة من المتحولات ضمن متحول واحد يساعد في الوصول إليها و الإضافة إليها و الحذف.

يتم تعريف مصفوفة كما يلي: `int[] ints = new int[10];`

نستطيع الحصول على حجم المصفوفة باستخدام الوصفة `length`.

الصف `Arrays` يمتلك بعض الطرائق للتعامل مع المصفوفات.

الطريقة `sort()` لترتيب مصفوفة.

الطريقة `binarySearch()` لتنفيذ خوارزمية البحث الثنائي على المصفوفة, يجب أن يتم ترتيبها أولاً.

الطريقة `fill()` لوضع قيمة ثابتة ضمن كل عنصر في المصفوفة.

الاستثناءات

الاستثناءات هي أخطاء تحدث أثناء التشغيل فقط, لا يمكن معالجتها قبل تنفيذ البرنامج لكن يمكن توقع الكود الذي سيسبب إستثناء و كتابته ضمن كتلة `try/catch` لمعالجة الإستثناء في حال حدوثه أثناء التشغيل.

يمكن استخدام العبارة `throws` بعد تعريف طريقة ما لكي نعبر أن هذه الطريقة يمكن أن تسبب إستثناء و بالتالي عند إستدعاء هذه الطريقة نكون مجبرين على وضع الإستثناء ضمن `try/catch`.

عند استخدام أكثر من عبارة `catch`, نضع الإستثناءات العامة في الأخير و نعالج أولاً الإستثناءات الخاصة.

يمكن أن نقوم بتسبب إستثناء بشكل يدوي عن طريق العبارة `throw`.

يمكن تعريف استثناءات جديدة عن طريق الوراثة من الصف `Exception`.

الطريقة `getMessage()` تطبع رسالة الإستثناء, و الطريقة `printStackTrace()` تطبع جميع التوابع التي تم إستدعاؤها للوصول إلى الإستثناء و لم يتم العودة منها.

الكتلة `finally` يتم استدعاؤها دوماً في حال حصل إستثناء أو لم يحصل.

الواجهات

تستخدم الواجهات لفصل تعريف الطريقة عن تحقيقها, حيث يتم وصف الطريقة و وسطائها و القيمة المعادة منها في الواجهة لكي يتم لاحقاً تحقيق الطريقة ضمن الصف الذي يحقق الواجهة.

ضمن الواجهة يمكن تعريف الواصلات `final` و `static` فقط, يستطيع الصف تحقيق أكثر من واجهة و تستطيع الواجهة أن تراث من أكثر من واجهة أخرى.

الحزم

تستخدم الحزم `packages` من أجل ترتيب الكود البرمجي ضمن ملفات و مجلدات منفصلة, حيث نجمع الصفوف, الواجهات الخ...

السطر الأول من الملف يوضح اسم الحزمة الموجود ضمنها الملف, تليها عمليات استيراد الحزم الأخرى التي نحتاجها ضمن الملف. يمكن أن نضع صفوف بنفس الاسم ضمن حزم مختلفة.

تمرين

قم بتعريف واجهة باسم `MathUtilsInt` تتضمن الطرائق التالية:

الطريقة `max` التي تأخذ عددين كوسيط لها و ترد العدد الأكبر, يجب أن تقبل أي نوع للوسطاء هي `byte, short, int, long, float, double`.

الطريقة `median` التي تقبل مصفوفة `int` كوسيط و ترد العنصر الأوسط `median` في المصفوفة و طريقة `average` ترد متوسط عناصر المصفوفة, يجب على هاتين الطريقتين أن تسبب إستثناء معرف من قبل المستخدم في حال كانت المصفوفة فارغة مع رسالة مناسبة.

الطريقة `sum` لجمع مصفوفتين أحاديتين `two vectors` ممرقتين كوسيط و ترد ناتج الجمع, يتم جمع العناصر عنصر عنصر.

تعريف صف `MathUtils` لتحقيق الواجهة السابقة مع كافة طرقها.

Contents

أهداف الجلسة	1
Number استخدام الصف	1
Character استخدام الصف	1
String استخدام الصف	1
المصفوفات	2
الإستثناءات	2
الواجهات	3
الحزم	3
تمرين	3