

جافا في ال أوامر ال سطر خيارات تحليل

وتحليل بتحديد لك تسمح تطبيقاتك. في ال أوامر ال سطر خيارات تحليل لتسهيل مصممة `Parser` مكتبة هو `Parser` دليل أدناه ومفيدة. مرنه بطريقة `--file input.txt` مثل حجج مع وخيارات الصريح `-v` مثل ال أعلام مثل الخيارات فعال. بشكل `Parser` استخدام لكي في على بخطة خطوة

مشرورك إلى `Parser` إضافة: 1 الخطوة

ملف إلى ال التالية التبعية أضف، `Parser` تستخدم كنت إذا `Parser` مشروعك في تضمينه أول أعليك، `Parser` لاستخدام بك: الخاص `pom.xml`

```
<dependency>
  <groupId>net.sf.jopt-simple</groupId>
  <artifactId>jopt-simple</artifactId>
  <version>5.0.4</version>
</dependency>
```

البناء لأدوات بالنسبة الإصدار. أحدث هو 5.0.4 يكون لا قد حيث، `Parser` `Parser` على الإصدار أحدث من التحقق من تأكد `implementation 'net.sf.jopt-simple:jopt-simple:5.0.4'` مثل من اسب بشكل ال تبعية تعديليمكنك، `Parser` مثل الأخرى

مشرورك إلى `Parser` إنشاء: 2 الخطوة

مثيل بإنشاء ابداً ال أوامر. ال سطر خيارات وتحليل لتحديد تستخدمها ال `OptionParser` فئة هو `Parser` من القلب بك: الخاص `main` طريقة في منه

```
import joptsimple.OptionParser;
import joptsimple.OptionSet;

public class MyApp {
    public static void main(String[] args) {
        OptionParser parser = new OptionParser();
        //          )      (3
    }
}
```

الأوامر السطرية لتحديد: 3 الخطوة

تتطلب خيارات أو حجج [] بدون أعلام الخيارات تكون أن يمكن. accepts أو acceptsAll طرق باستخدام الخيارات تحديدي يمكن إعدادها: كيفية إلك عدد. أو ملف اسم [] مثل حجج

[] --verbose و -v [] مثل بديلة أسماء لتحديد acceptsAll أو واحد خيار لاسم accepts استخدم: الأعلام []

```
parser.acceptsAll(Arrays.asList("v", "verbose"), " " " " ;(
```

[] باستخدام نوعها تحديدي واختياري قيمة، إلى يحتاج خيار أن إلى لإشارة withRequiredArg() استخدم: حجج مع خيارات ofType():

```
parser.acceptsAll(Arrays.asList("f", "file"), " " " " .(withRequiredArg());
parser.acceptsAll(Arrays.asList("c", "count"), " " " " .(withRequiredArg().ofType(Integer.class).defaultTo(0));
```

[] defaultsTo(0) مثل افتراضية قيمة يحدد

[] ofType(Integer.class) من يضمن أن يملك

[] --help أو -h [] مثل المساعدة علم أضف: المساعدة خيار

```
parser.acceptsAll(Arrays.asList("h", "help"), " " " " ;(
```

الأوامر السطرية حجج تحليل: 4 الخطوة

يحتوي OptionSet كائن يعيد هذا الأوامر. السطر المدخلات لتحليل المحلل إلى main طريقة من args مصفوفة بتحويل قم المحلل: الخيارات على

```
OptionSet options = parser.parse(args);
```

المفردة: الحجج أو الصالحة غير الخيارات [] مثل التحليل أخطاء لمعالجة try-catch كتلة في هذا بتغلي قم

```
try {
    OptionSet options = parser.parse(args);
    // (5
} catch (Exception e) {
    System.err.println(" " + e.getMessage());
    try {
        parser.printHelpOn(System.err);
    } catch (IOException ex) {
        ex.printStackTrace();
    }
    System.exit(1);
}
```

المحللة الخيارات إلى الوصول: 5 الخطوة

بالخيارات: المرتبطة غير الحجج واسترجاع الخيارات، قيم استرجاع الأعلام، من للتحقق OptionSet استخدم

موجود: علم كان إذا لرؤية has() استخدم: الأعلام من تحقق

```
boolean verbose = options.has("v");
if (verbose) {
    System.out.println( "      " );(
}
```

الأم: لزم إذا المناسب النوع إلى بتحويلها قم خيار، حجة لاسترجاع valueOf() استخدم: الخيارات قيم استرجاع

```
String fileName = (String) options.valueOf("f"); // `null`
int count = (Integer) options.valueOf("c"); // `0` `defaultsTo(0)`
```

الافتراضية. القيمة أو نوعها المحدد القيمة valueOf() يعيد، defaultsTo() و ofType() بتحديد قيمت إذا

باستخدام الملفات من قائمة مثل بالخيارات مرتبطة ليست التي الحجج استرجاع: بالخيارات مرتبطة غير حجج nonOptionArguments():

```
List<String> files = options.nonOptionArguments();
System.out.println( " " + files);
```

موجود: المساعدة خيار كان إذا الاستخدام مع لومات طباعة: المساعدة مع الحجة

```
if (options.has("h")) {
    parser.printHelpOn(System.out);
    System.exit(0);
}
```

شيء لكل جمع مثال:

الملفات: من وقائمة عدد، خيار صريح، علم يقبل لبرنامج كامل مثال إليك

```
import joptsimple.OptionParser;
import joptsimple.OptionSet;
import java.io.IOException;
import java.util.Arrays;
import java.util.List;

public class MyApp {
    public static void main(String[] args) {
```

```

OptionParser parser = new OptionParser();
parser.acceptsAll(Arrays.asList("v", "verbose"), " " " ");
parser.acceptsAll(Arrays.asList("c", "count"), " " " " (
    .withRequiredArg()
    .ofType(Integer.class)
    .defaultsTo(0);
parser.acceptsAll(Arrays.asList("h", "help"), " " " " ");

try {
    OptionSet options = parser.parse(args);

    if (options.has("h")) {
        parser.printHelpOn(System.out);
        System.exit(0);
    }

    boolean verbose = options.has("v");
    int count = (Integer) options.valueOf("c");
    List<String> files = options.nonOptionArguments();

    if (verbose) {
        System.out.println(" " " ");
    }

    System.out.println(" " " " + count);
    System.out.println(" " " " + files);

} catch (Exception e) {
    System.err.println(" " " " + e.getMessage());
    try {
        parser.printHelpOn(System.err);
    } catch (IOException ex) {
        ex.printStackTrace();
    }
    System.exit(1);
}
}
}

```

وسيخرج: java MyApp -v -c 5 file1 file2، هذا بتشغيل قم

```
: 5
: [file1, file2]
```

المحددة. الخيارات جميع مع الاستخدام رسالة وسيعرض `-h` مع بتشغيله قم

إضافية ميزات

□ `-I dir1 -I dir2`: مثل متعددة مرات يظهر أن لخيار السماح: متعددة قيم

```
parser.accepts("I").withRequiredArg().ofType(String.class);
List<String> includes = options.valuesOf("I"); //
```

□ مطلوبة: الحجة تكن لم إذا `withRequiredArg()` من بدلاً `withOptionalArg()` استخدم: اختيارية حجج

```
parser.accepts("o").withOptionalArg().ofType(String.class);
```

□ ردود لتقديم عليه القبض يمكنك التحليل، لأخطاء `OptionException` استثناء `□□□□□□□□□□` يرمي: الأخطاء مع الحجة
مستخدمة. فعل

ملخص

□□□□□□□□□□ إنشاء 2. مشروعك. إلى المكتبة إضافة 1. يمكنك: □□□□. في الأوامر السطرية مع التعامل السهل من يجعل □□□□□□□□□□ استخدام 4. `OptionSet` إلى `args` مصفوفة تحليل 3. الافتراضية □□. القيم الحجج، الأعلام، الخيارات وتحدي `OptionParser` الحاجة. حسب المساعدة وتقديم الأخطاء مع الحجة 5. بالخيارات. المرتبطة غير والحجج القيم، الأعلام، إلى للوصول `OptionSet` تطبق! لحاجات مخصصة قوية الأوامر السطر واجهات بناء يمكنك الأساسيات، هذه مع