```
// $Id: jgrep.java,v 1.2 2014-04-02 21:30:12-07 - - $

//
// This program is a stub showing how to create a pattern from a
// regular expression.  It does not handle options or files, and
// has some other bugs which you must discover and fix.
//

import java.io.*;
import java.util.Scanner;
import java.util.regex.*;
import static java.lang.System.*;

class jgrep {

   static void scanfile (Scanner input, String filename,
                         Pattern pattern) {
      while (input.hasNextLine()) {
         String line = input.nextLine();
         boolean matches = pattern.matcher (line).find();
         if (matches) {
            out.printf ("%s:%s%n", filename, line);
         }
      }
   }

   public static void main (String[] args) {
      options opts = new options (args);
      try {
         Pattern pattern = Pattern.compile (opts.regex);
         if (opts.filenames.length == 0) {
            scanfile (new Scanner (in), "<stdin>", pattern);
         }else {
            for (int argi = 1; argi < opts.filenames.length; ++argi) {
               try {
                  String filename = opts.filenames[argi];
                  Scanner input = new Scanner (new File (filename));
                  scanfile (input, filename, pattern);
                  input.close();
               }catch (IOException error) {
                  messages.warn (error.getMessage());
               }
            }
         }
      }catch (PatternSyntaxException error) {
         messages.die (error.getMessage());
      }
      exit (messages.exit_status);
   }

}

```

```java
// $Id: messages.java,v 1.1 2014-03-24 18:45:16-07 - - $

import static java.lang.System.*;

class messages {
   public static final int EXIT_SUCCESS = 0;
   public static final int EXIT_FAILURE = 1;
   public static final String program_name =
                 basename (getProperty ("java.class.path"));
   public static int exit_status = EXIT_SUCCESS;

   //
   // constructor - prevents instantiation: only static fns allowed.
   //
   private messages() {
      throw new UnsupportedOperationException();
   }

   //
   // basename - strips the dirname and returns only the basename.
   //            See:  man -s 3c basename
   //
   public static String basename (String pathname) {
      if (pathname == null || pathname.length () == 0) return ".";
      String[] paths = pathname.split ("/");
      for (int index = paths.length - 1; index >= 0; --index) {
         if (paths[index].length () > 0) return paths[index];
      }
      return "/";
   }

   //
   // warn - print a warning and set exit status to failure.
   //
   public static void warn (Object... args) {
      exit_status = EXIT_FAILURE;
      err.printf ("%s", program_name);
      for (Object arg: args) err.printf (": %s", arg);
      err.printf ("%n");
   }

   //
   // die - print a warning and exit program.
   //
   public static void die (Object... args) {
      warn (args);
      exit (exit_status);
   }

}
```

```
// $Id: options.java,v 1.2 2014-04-02 21:30:12-07 - - $

import static java.lang.System.*;

class options {
   boolean insensitive;
   boolean filename_only;
   boolean number_lines;
   boolean reverse_match;
   String regex;
   String[] filenames;

   options (String[] args) {
      if (args.length == 0) {
         err.printf ("Usage: %s [-ilnv] regex [filename...]%n",
                     messages.program_name);
         exit (messages.EXIT_FAILURE);
      }
      regex = args[0];
      filenames = new String[args.length - 1];
      for (int argi = 1; argi < args.length; ++argi) {
         filenames[argi - 1] = args[argi];
      }
   }
}

```

```
# $Id: Makefile,v 1.1 2014-03-24 18:45:16-07 - - $

JAVASRC    = jgrep.java messages.java options.java
ALLSOURCE  = ${JAVASRC} Makefile README
MAINCLASS  = jgrep
CLASSES    = ${JAVASRC:.java=.class}
JARCLASSES = ${CLASSES}
JARFILE    = jgrep
LISTING    = Listing.ps
SUBMITDIR  = cmps012b-wm.s14 asg1

all : ${JARFILE}
	- checksource ${ALLSOURCE}

${JARFILE} : ${CLASSES}
	echo Main-class: ${MAINCLASS} >Manifest
	jar cvfm ${JARFILE} Manifest ${JARCLASSES}
	- rm Manifest
	chmod +x ${JARFILE}

%.class : %.java
	javac $<

clean :
	- rm ${JARCLASSES}

spotless : clean
	- rm ${JARFILE}

ci : ${ALLSOURCE}
	cid + ${ALLSOURCE}
	- checksource ${ALLSOURCE}

lis : ${ALLSOURCE}
	mkpspdf ${LISTING} ${ALLSOURCE}

submit : ${ALLSOURCE}
	submit ${SUBMITDIR} ${ALLSOURCE}

again : ${ALLSOURCE}
	make spotless ci all lis

```

```
# $Id: README,v 1.1 2014-03-24 18:45:16-07 - - $
```