

```

package vmtranslator;

public class VMParser {

    public String commandType(String line) { // Returns to CodeWriter the type of command
        (A, B, C, D) see what they are in isA(), isB(), isC(), isD()
        String id = arg1(line); // Basically returns the first word of the command, which
        identifies the command type.
        if (isA(id)) return "A";
        if (isB(id)) return "B";
        if (isC(id)) return "C";
        if (isD(id)) return "D";
        return "ERROR";
    }

    private boolean isA(String id) {
        return id.equals("add") ||
            id.equals("sub") ||
            id.equals("neg") ||
            id.equals("eq") ||
            id.equals("gt") ||
            id.equals("lt") ||
            id.equals("and") ||
            id.equals("or") ||
            id.equals("not");
    }

    private boolean isB(String id) {
        return id.equals("push") ||
            id.equals("pop");
    }

    private boolean isC(String id) {
        return id.equals("label") ||
            id.equals("if-goto") ||
            id.equals("goto");
    }

    private boolean isD(String id) {
        return id.equals("function") ||
            id.equals("call") ||
            id.equals("return");
    }

    // [command] -> [arg1 arg2 arg3]

    public String arg1(String line) { // Returns the command identifier
        if (!line.contains(" ")) return line;
        else return line.substring(0, line.indexOf(" "));
    }

    public String arg2(String line) {
        int firstSpaceIndex = line.indexOf(" ");
        if (firstSpaceIndex == -1) return
null; // If there's
no space in 'line', there is no arg2.
        int secondSpaceIndex = line.indexOf(" ", firstSpaceIndex + 1);
        if (secondSpaceIndex == -1) return line.substring(firstSpaceIndex + 1).replaceAll("
", ""); // If there is no second space, arg2 is just the substring after the space.
        return line.substring(firstSpaceIndex + 1, secondSpaceIndex).replaceAll("
",
"); // If there is a second space, arg2 is the substring between the two
spaces.
    }

    public String arg3(String line) {
        int firstSpaceIndex = line.indexOf(" ");

```

```
    if (firstSpaceIndex == -1) return null;
    int secondSpaceIndex = line.indexOf(" ", firstSpaceIndex + 1);
    if (secondSpaceIndex == -1) return null;
    return line.substring(secondSpaceIndex + 1).replaceAll(" ", ""); // Only if there
    is a second space, it returns the substring after it.
}
}
```