

# Vimでコード印刷

---

小見 拓 mail@nanasi.jp

名無しのvim使い <http://nanasi.jp/>

twitter @taku\_o

# 用意する環境

# 用意する環境

- Windows

# 用意する環境

- Windows
- プリンタが必要。

# 入門編

# Vimでの印刷実行方法

# Vimでの印刷実行方法

- :hardcopyコマンドを使うと印刷できる。

# Vimでの印刷実行方法

- :hardcopyコマンドを使うと印刷できる。
- printheaderオプションとか、 printoptions オプションといった印刷時用のオプションが用意されている。

# Vimでの印刷実行方法

- :hardcopyコマンドを使うと印刷できる。
- printheaderオプションとか、 printoptions オプションといった印刷時用のオプションが用意されている。
- どのように印刷するかは、それらオプションを設定することで決められる。

続きは:help printingで

入門編  
完

# 実践編

Vimを使って何を  
印刷するか？

プログラムのコードとか？

# Vimの印刷の良い点①

- ・構文ハイライト付きで印刷できる。

```
}

/**  
 * {@link #isAnd()}の値に応じてANDもしくはOR条件で評価します。  
 */  
public boolean match(Object o) {  
    if(o == null){  
        return false;  
    }  
    if (childCriterions == null || childCriterions.size() == 0) {  
        // 条件なし  
        return true;  
    }  
    for (Iterator it = childCriterions.iterator(); it.hasNext();) {  
        HBICriterion c = (HBICriterion) it.next();  
        boolean isMatch = c.match(o);  
        //AND OR 条件  
        if(!isAnd && isMatch){  
            //ORで一つ一致  
            return true;  
        }  
    }  
}
```

プログラムコードの印刷は、  
印刷枚数が多くなってしまう

# Vimの印刷の良い点②

- :hardcopyコマンド実行時に、印刷範囲を指定できる。
- 印刷したい箇所だけ印刷できる。

```
:3,12hardcopy  
" 3行目から12行目を印刷
```

```
:'  
<, '  
>hardcopy  
" 選択範囲を印刷
```



- 文字サイズを小さくしたり、紙のサイズを大きくしたり、両面印刷すれば、紙の印刷量は減らせる。

- ・文字サイズを小さくしたり、紙のサイズを大きくしたり、両面印刷すれば、紙の印刷量は減らせる。
- ・プログラムのコードの印刷なら、普通、行番号ぐらいつけるでしょう。

続きはwebで

<http://nanasi.jp/howto/hardcopy.html>

実践編

完

ところで、普段  
どのようなプリンタを使用  
していますか？

- ・こんな感じのプリンタ使っています。



× N台

※画像はイメージです。

- ・白黒印刷用、カラー印刷用のプリンタがあって、通常は白黒、お客様納品用はカラーで印刷している。

白黒プリンタには  
気をつけるべし！！

# カラースキーマによっては 酷い目にあうぞ

```
18 public static final String PARAMETER = "test.imunit.TestCore.testClass";
19
20 /**
21 * テストを実行し、テスト結果を出力します。
22 * HTTPリクエストの"test.imunit.TestCore.testClass"キーに入れられた、テストクラスを対象にクラスを実行する。
23 * 利用方法の詳細については、imunit.TestRunnerを参照してください。
24 *
25 * @param context
26 * @param request
27 * @param response
28 */
29
30 public void action(HTTPActionEventContext context, HttpServletRequest request, HttpServletResponse response)
31 throws ServletException {
32
33 // HTTPリクエストから、テスト対象クラスを抜き出す。
34 String testClass = request.getParameter(PARAMETER);
35
36 PrintWriter out = null;
37 try {
38     // JUnitCoreでテスト実行
39     Result result = JUnitCore.runClasses(Class.forName(testClass));
40
41     // ブレーンテキストでテスト結果を返す
42     response.setContentType("text/plain");
43     out = response.getWriter();
44
45     // テスト成功の場合は、テスト成功メッセージを出力。
46     if (result.wasSuccessful()) {
47         out.println("All test were successful.");
48
49     // テスト失敗
50     } else {
51         // テスト失敗結果を取得して出力
```

印刷物をスキャンしたもの（例はevening）

# カラースキーマによっては 酷い目にあうぞ

見づらい

```
18 public static final String PARAMETER = "test.imunit.testCore.testClass";
19
20 /**
21 * テストを実行し、テスト結果を出力します。
22 * HTTPリクエストの"test.imunit.TestCore.testClass"キーに入れられた、テストクラスを対象にクラスを実行する。
23 * 利用方法の詳細については、imunit.TestRunnerを参照してください。
24 *
25 * @param context
26 * @param request
27 * @param response
28 */
29
30 public void action(HTTPActionEventContext context, HttpServletRequest request, HttpServletResponse response)
31 throws ServletException {
32
33 // HTTPリクエストから、テスト対象クラスを抜き出す。
34 String testClass = request.getParameter(PARAMETER);
35
36 PrintWriter out = null;
37 try {
38     // JUnitCoreでテスト実行
39     Result result = JUnitCore.runClasses(Class.forName(testClass));
40
41     // ブレーンテキストでテスト結果を返す
42     response.setContentType("text/plain");
43     out = response.getWriter();
44
45     // テスト成功の場合は、テスト成功メッセージを出力。
46     if (result.wasSuccessful()) {
47         out.println("All test were successful.");
48
49     // テスト失敗
50     } else {
51         // テスト失敗結果を取得して出力
```

印刷物をスキャンしたもの（例はevening）

# カラースキーマによっては 酷い目にあうぞ

見づらい

```
18 public static final String PARAMETER = "test.imunit.TestCore.testClass";
19
20 /**
21 * テストを実行し、テスト結果を出力します。
22 * HTTPリクエストの"test.imunit.TestCore.testClass"キーに入れられた、テストクラスの象にクラスを実行する。
23 * 利用方法の詳細については、imunit.TestRunnerを参照してください。
24 *
25 * @param context
26 * @param request
27 * @param response
28 */
29
30 public void action(HTTPActionEventContext context, HttpServletRequest request, HttpServletResponse response)
31 throws ServletException {
32
33 // HTTPリクエストから、テスト対象クラスを抜き出す。
34 String testClass = request.getParameter(PARAMETER);
35
36 PrintWriter out = null;
37 try {
38     // JUnitCoreでテスト実行
39     Result result = JUnitCore.runClasses(Class.forName(testClass));
40
41     // プレーンテキストでテスト結果を返す
42     response.setContentType("text/plain");
43     out = response.getWriter();
44
45     // テスト成功の場合は、テスト成功メッセージを出力。
46     if (result.wasSuccessful()) {
47         out.println("All test were successful.");
48
49     // テスト失敗
50     } else {
51         // テスト失敗結果を取得して出力
```

読めない

印刷物をスキャンしたもの（例はevening）

# カラースキーマによっては 酷い目にあうぞ

見づらい

```
18 public static final String PARAMETER = "test.imunit.testCore.testClass";
19
20 /**
21 * テストを実行し、テスト結果を出力します。
22 * HTTPリクエストの"test.imunit.TestCore.testClass"キーに入れられた、テストクラスの象にクラスを実行する。
23 * 利用方法の詳細については、imunit.TestRunnerを参照してください。
24 *
25 * @param context
26 * @param request
27 * @param response
28 */
29
30 public void action(HTTPAction context, HttpServletRequest request, HttpServletResponse response)
31 throws ServletException {
32
33 // HTTPリクエストから、テスト対象クラスを抜き出す。
34 String testClass = request.getParameter(PARAMETER);
35
36 PrintWriter out = null;
37 try {
38     // JUnitCoreでテスト実行
39     Result result = JUnitCore.runClasses(Class.forName(testClass));
40
41     // ブラントキストでテスト結果を返す
42     response.setContentType("text/plain");
43     out = response.getWriter();
44
45     // テスト成功の場合は、テスト成功メッセージを出力。
46     if (result.wasSuccessful()) {
47         out.println("All test were successful.");
48
49     // テスト失敗
50     } else {
51         // テスト失敗結果を取得して出力
```

読みない

読みにくい

印刷物をスキャンしたもの（例はevening）

# カラースキーマによっては 酷い目にあうぞ

見づらい

```
18 public static final String PARAMETER = "test.imunit.TestCore.testClass";
19
20 /**
21 * テストを実行し、テスト結果を出力します。
22 * HTTPリクエストの"test.imunit.TestCore.testClass"キーに入れられた、テストクラスの象にクラスを実行する。
23 * 利用方法の詳細については、imunit.TestRunnerを参照してください。
24 *
25 * @param context
26 * @param request
27 * @param response
28 */
29
30 public void action(HTTPAction context, HttpServletRequest request, HttpServletResponse response)
31 throws ServletException {
32
33 // HTTPリクエストから、テスト対象クラスを抜き出す。
34 String testClass = request.getParameter(PARAMETER);
35
36 PrintWriter out = null;
37 try {
38     // JUnitCoreでテスト実行
39     Result result = JUnitCore.runClasses(Class.forName(testClass));
40
41     // ブレーンテキストでテスト結果を返す
42     response.setContentType("text/plain");
43     out = response.getWriter();
44
45     // テスト成功の場合は、テスト成功メッセージを出力。
46     if (result.wasSuccessful()) {
47         out.println("All test were successful.");
48
49     // テスト失敗
50     } else {
51         // テスト失敗結果を取得して出力
```

読みない

読みにくい

背景色と文字色で同系統  
使うのはやめてくれ

印刷物をスキャンしたもの（例はevening）

ご静聴ありがとうございました