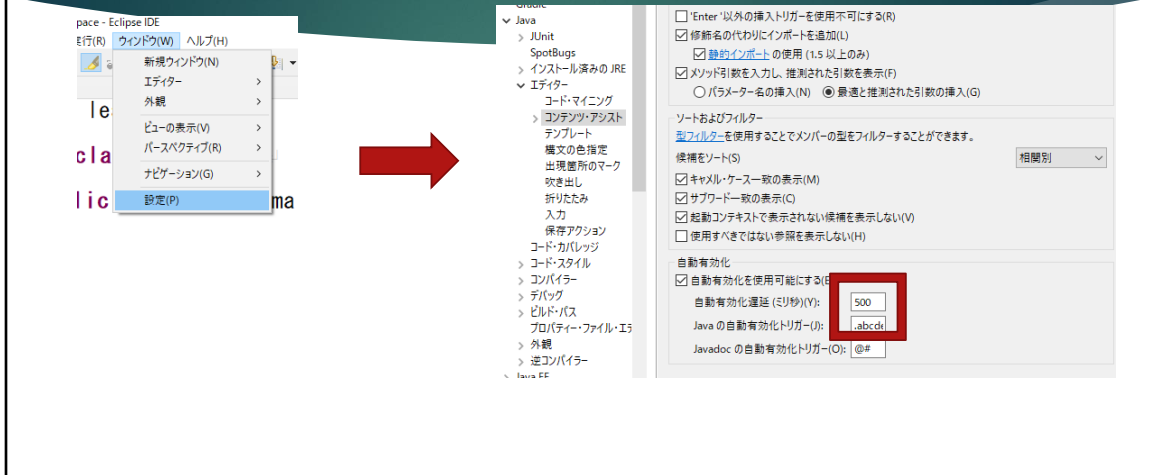




Eclipse設定集

～これであなたも開発効率2倍！～

自動補完機能



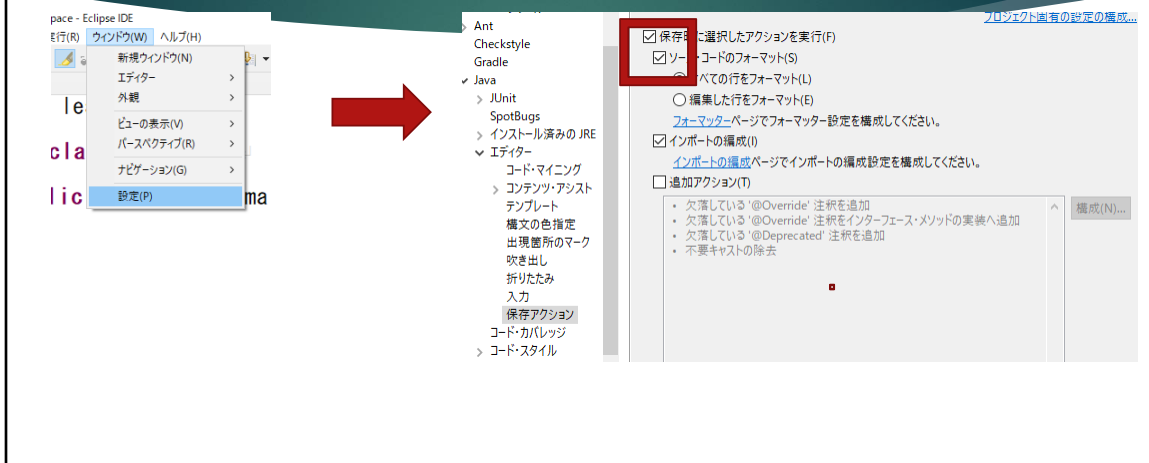
ソースコードを入力すると、コード補完機能が表示されます。
そのトリガーとなる文字と、表示するまでの時間を設定します。何も設定しなくても、コード入力中に「Ctrl + スペース」でも表示できます。

ウィンドウタブ→設定と進み
サイドバーから
> Java
 > エディター
 > コンテンツアシスト
を選択

画面下段の

- ・自動有効化遅延を500（0.5秒遅延）
 - ・自動有効化トリガーに「.@abcdefghijklmnopqrstuvwxyz」を加える
- 適用ボタンを押下

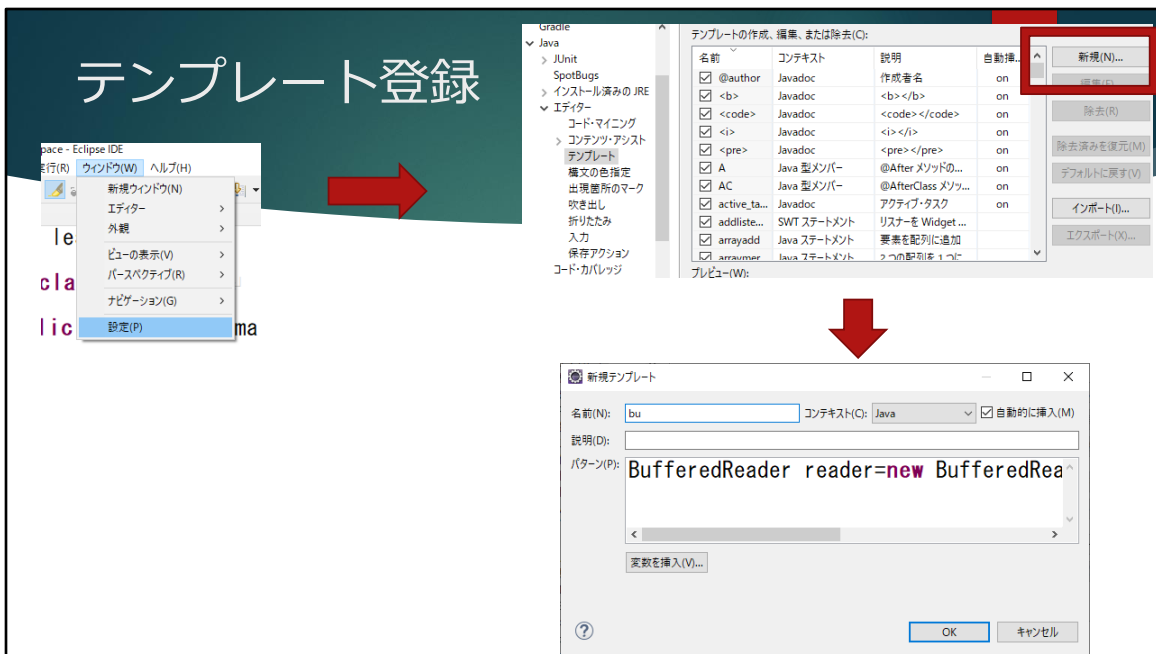
保存時の自動フォーマッター



ソースコードのインデントを自動的に調整する機能です。「Ctrl + Shift + F」でも実行できます。
これを、ソースの保存時に自動的に実行するように設定します。

ウィンドウタブ→設定と進み
サイドバーから
> Java
 > エディター
 > 保存アクション
を選択

ソースコードのフォーマット にチェックを入れる
適用ボタンを押下



自動補完機能で補完するテンプレートです。

ウィンドウタブ→設定と進み

サイドバーから

> Java

> エディター

> テンプレート

を選択

新規ボタンを押下

名前に入力時のキーワードを入力 (sysoutにあたるもの)

パターンに呼び出したいテンプレートを入力

OKを押下

適用ボタンを押下

任意の行移動

```
package lesson1;

public class Sample01 {

    public static void main(String[] args) {

        System.out.println("Hello");

        for (int i = 0; i < args.length; i++) {

        }

    }

}
```

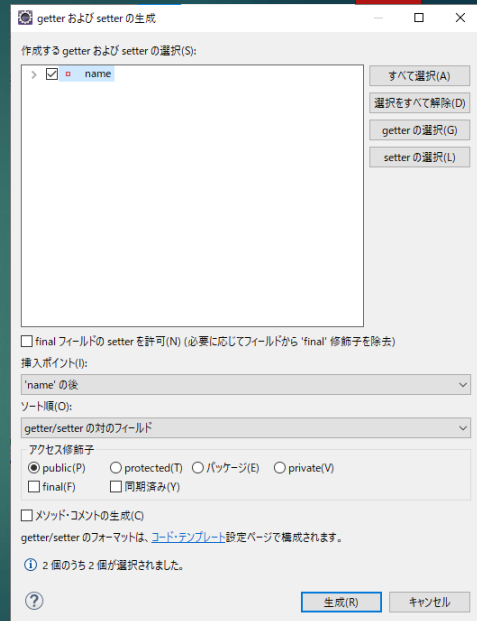


```
1 package lesson1;
2
3 public class Sample01 {
4
5     public static void main(String[] args) {
6
7         for (int i = 0; i < args.length; i++) {
8             System.out.println("Hello");
9         }
10    }
11
12 }
```

Altキー + ↑ または Altキー + ↓
でカーソルが当たっている行の移動が可能です。

また範囲を選択してから移動も可能です

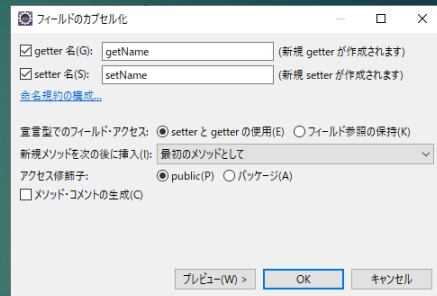
Setter,Getter自動生成



ソースタブ→Getter および Setterの生成

該当のフィールドにチェック
生成ボタンを押下

Setter,Getter自動生成 +カプセル化

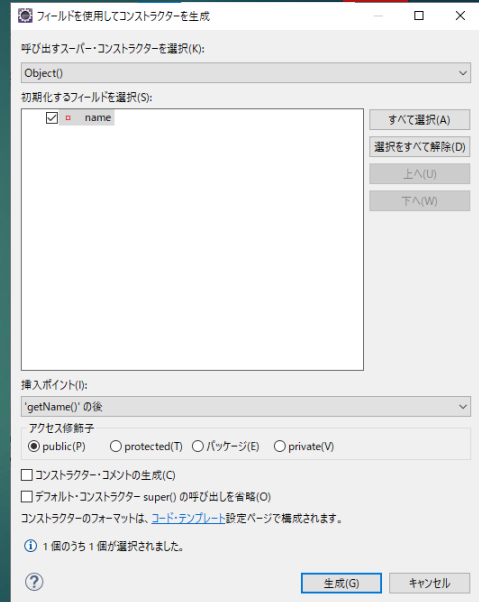


こちらを利用いただいても構いません
(※ただしフィールド一つずつしか作成することはできません。)

フィールド名にカーソルを合わせた後
リファクタリングタブ→フィールドのカプセル化

アクセス修飾子を「public」にチェック
OKボタンを押下

ソース(S)	リファクタリング(T)	ナビゲート(N)	検索(A)	プロジェクト(P)	その他(O)
コメントの切り替え(L)					Ctrl + 7
ブロック・コメントの追加(B)					Ctrl + Shift + *
ブロック・コメントの除去(K)					Ctrl + Shift + ¥
要素コメントの生成(E)					Alt + Shift + +
テキスト・ブロックの追加(T)					Ctrl + Shift + ^
右ヘシフト(S)					
左ヘシフト(H)					
インデントの訂正(I)					Ctrl + I
フォーマット(F)					Ctrl + Shift + F
要素のフォーマット(N)					
インポートの追加(D)					Ctrl + Shift + M
インポートの編集(G)					Ctrl + Shift + O
メンバーのソート(O)...					
クリーンアップ(U)...					
メソッドのオーバーライド/実装(V)...					
getter および setter の生成(R)...					
変数メソッドの生成(M)...					
hashCode() および equals() の生成(H)...					
toString() 生成(S)...					
フィールドを使用してコンストラクターを生成(A)...					
スーパークラスからコンストラクターを生成(C)...					
関数(W)					Alt + Shift + Z >
スプリングの外外部(X)...					
壊れた外外部にされたスプリングの検索(T)					



該当のフィールドにチェック
生成ボタンを押下