SRC：'if ( cut CaMeL Action != null ) cut CaMeL Action . add CaMeL Property CaMeL Change CaMeL Listener ( cut CaMeL Action CaMeL Listener ) ; <CTX> public void set CaMeL Cut CaMeL Action ( I CaMeL Action action ) { if ( cut CaMeL Action == action ) return ; if ( cut CaMeL Action != null ) cut CaMeL Action . remove CaMeL Property CaMeL Change CaMeL Listener ( cut CaMeL Action CaMeL Listener ) ; cut CaMeL Action = action ; if ( cut CaMeL Action != null ) cut CaMeL Action . add CaMeL Property CaMeL Change CaMeL Listener ( cut CaMeL Action CaMeL Listener ) ; cell CaMeL Cut CaMeL Action . update CaMeL Enabled CaMeL State ( ) ; }'

if ( model != null ) <CTX> private void add CaMeL Feature CaMeL Plugins ( I CaMeL Feature feature , Hash CaMeL Map map ) { Plugin CaMeL Model CaMeL Manager manager = PDE CaMeL Core . get CaMeL Default ( ) . get CaMeL Model CaMeL Manager ( ) ; I CaMeL Feature CaMeL Plugin [ ] plugins = feature . get CaMeL Plugins ( ) ; for ( int i = 0 ; i < plugins . length ; i ++ ) { String id = plugins [ i ] . get CaMeL Id ( ) ; if ( id == null || map . contains CaMeL Key ( id ) ) continue ; I CaMeL Plugin CaMeL Model CaMeL Base model = manager . find CaMeL Model ( id ) ; if ( model != null ) map . put ( id , model ) ; } } if ( model != null && Target CaMeL Platform . matches CaMeL Current CaMeL Environment ( model ) )

SRC：if ( cut CaMeL Action != null ) cut CaMeL Action . add CaMeL Property CaMeL Change CaMeL Listener ( cut CaMeL Action CaMeL Listener ) ;

TGT：if ( cut CaMeL Action != null ) { cut CaMeL Action . add CaMeL Property CaMeL Change CaMeL Listener ( cut CaMeL Action CaMeL Listener ) ; }

String CaMeL Buffer buff = new String CaMeL Buffer ( ) ; <CTX> private void check CaMeL Required CaMeL Options ( ) throws Missing CaMeL Option CaMeL Exception { if ( required CaMeL Options . size ( ) > 0 ) { Iterator iter = required CaMeL Options . iterator ( ) ; String CaMeL Buffer buff = new String CaMeL Buffer ( ) ; while ( iter . has CaMeL Next ( ) ) { buff . append ( iter . next ( ) ) ; } throw new Missing CaMeL Option CaMeL Exception ( buff . to CaMeL String ( ) ) ; } }

for ( Integer count : arr ) { <CTX> public class BUC@@ KET@@ SORT { public static Array CaMeL List < Integer > bu@@ ck@@ et@@ sort ( Array CaMeL List < Integer > arr , int k ) { Array CaMeL List < Integer > counts = new Array CaMeL List < Integer > ( Collections . n CaMeL Copies ( k , 0 ) ) ; for ( Integer x : arr ) { counts . set ( x , counts . get ( x ) + 1 ) ; } Array CaMeL List < Integer > sorted \_ arr = new Array CaMeL List < Integer > ( $NUMBER$ ) ; int i = 0 ; for ( Integer count : arr ) { / / arr is counts in fixed version sorted \_ arr . add CaMeL All ( Collections . n CaMeL Copies ( count , i ) ) ; i ++ ; } return sorted \_ arr ; } } for ( Integer count : counts ) {

Array CaMeL List others CaMeL List = new Array CaMeL List ( Arrays . as CaMeL List ( page . get CaMeL Editor CaMeL References ( ) ) ) ; if ( others CaMeL List . is CaMeL Empty ( ) ) { return null ; } I CaMeL Editor CaMeL Reference active = page . get CaMeL Active CaMeL Editor CaMeL Reference ( ) ; if ( active != null ) { others CaMeL List . remove ( active ) ; Array CaMeL List active CaMeL List = new Array CaMeL List ( 1 ) ; active CaMeL List . add ( active ) ; I CaMeL Editor CaMeL Part match = find CaMeL Editor ( input , active CaMeL List ) ; if ( match != null ) { return match ; } } return find CaMeL Editor ( input , others CaMeL List ) ; <CTX> public I CaMeL Editor CaMeL Part find CaMeL Editor ( I CaMeL Editor CaMeL Input input ) { Array CaMeL List others CaMeL List = new Array CaMeL List ( Arrays . as CaMeL List ( page . get CaMeL Editor CaMeL References ( ) ) ) ; if ( others CaMeL List . is CaMeL Empty ( ) ) { return null ; } I CaMeL Editor CaMeL Reference active = page . get CaMeL Active CaMeL Editor CaMeL Reference ( ) ; if ( active != null ) { others CaMeL List . remove ( active ) ; Array CaMeL List active CaMeL List = new Array CaMeL List ( 1 ) ; active CaMeL List . add ( active ) ; I CaMeL Editor CaMeL Part match = find CaMeL Editor ( input , active CaMeL List ) ; if ( match != null ) { return match ; } } return find CaMeL Editor ( input , others CaMeL List ) ; }

return find CaMeL Editor ( input , others CaMeL List ) ;

return find CaMeL Editor ( null , input , Workbench CaMeL Page . MATCH \_ INPUT ) ;