interest_field（核心领域）：

即我们需要集中的目标区域，特点是：连续性强，较大面积

```
    at
org.eclipse.e4.ui.bindings.keys.KeyBindingDispatcher.executeCommand(KeyBindingDispatcher

    at
org.eclipse.e4.ui.bindings.keys.KeyBindingDispatcher.press(KeyBindingDispatcher.java:507

    at
org.eclipse.e4.ui.bindings.keys.KeyBindingDispatcher.processKeyEvent(KeyBindingDispatche

    at
org.eclipse.e4.ui.bindings.keys.KeyBindingDispatcher.filterKeySequenceBindings(KeyBindin

    at
org.eclipse.e4.ui.bindings.keys.KeyBindingDispatcher.access$0(KeyBindingDispatcher.java:

    at
org.eclipse.e4.ui.bindings.keys.KeyBindingDispatcher$KeyDownFilter.handleEvent(KeyBindin

    at org.eclipse.swt.widgets.EventTable.sendEvent(EventTable.java:84)
    at org.eclipse.swt.widgets.Display.filterEvent(Display.java:1262)
    at org.eclipse.swt.widgets.Widget.sendEvent(Widget.java:1060)
    at org.eclipse.swt.widgets.Widget.sendEvent(Widget.java:1085)
    at org.eclipse.swt.widgets.Widget.sendEvent(Widget.java:1070)
    at org.eclipse.swt.widgets.Widget.sendKeyEvent(Widget.java:1112)
    at org.eclipse.swt.widgets.Widget.sendKeyEvent(Widget.java:1108)
    at org.eclipse.swt.widgets.Widget.wmChar(Widget.java:1529)
    at org.eclipse.swt.widgets.Control.WM_CHAR(Control.java:4722)
    at org.eclipse.swt.widgets.Canvas.WM_CHAR(Canvas.java:343)
    at org.eclipse.swt.widgets.Control.windowProc(Control.java:4610)
    at org.eclipse.swt.widgets.Canvas.windowProc(Canvas.java:339)
    at org.eclipse.swt.widgets.Display.windowProc(Display.java:5023)
    at org.eclipse.swt.internal.win32.OS.DispatchMessageW(OS.java:-2)
    at org.eclipse.swt.internal.win32.OS.DispatchMessage(OS.java:2549)
    at org.eclipse.swt.widgets.Display.readAndDispatch(Display.java:3759)
    at
org.eclipse.e4.ui.internal.workbench.swt.PartRenderingEngine$9.run(PartRenderingEngine.j

    at org.eclipse.core.databinding.observable.Realm.runWithDefault(Realm.java:332)
    at
org.eclipse.e4.ui.internal.workbench.swt.PartRenderingEngine.run(PartRenderingEngine.jav

    at
org.eclipse.e4.ui.internal.workbench.E4Workbench.createAndRunUI(E4Workbench.java:148)

    at org.eclipse.ui.internal.Workbench$5.run(Workbench.java:636)
    at org.eclipse.core.databinding.observable.Realm.runWithDefault(Realm.java:332)
    at org.eclipse.ui.internal.Workbench.createAndRunWorkbench(Workbench.java:579)
    at org.eclipse.ui.PlatformUI.createAndRunWorkbench(PlatformUI.java:150)
    at
org.eclipse.ui.internal.ide.application.IDEApplication.start(IDEApplication.java:135)

    at
org.eclipse.equinox.internal.app.EclipseAppHandle.run(EclipseAppHandle.java:196)
    at
org.eclipse.core.runtime.internal.adaptor.EclipseAppLauncher.runApplication(EclipseAppLa

    at
org.eclipse.core.runtime.internal.adaptor.EclipseAppLauncher.start(EclipseAppLauncher.ja
```

调用链：

即 call — provide 的循环，从 stack 中提取

样例：

目前提取了四种调用链：

inner、outer1、outer2、all

即：核心区域（interest_field）、核心区域上方、核心区域下方、全部区域的调用链

Field（包领域）：

即：包含的 package 信息域

可这样理解：如果 field 完全不一致，则两个堆栈肯定没关系，若部分一致则计算重合率是多少；

获取方法：

把包变成一棵树，

root - org - eclipse - swt

- java - lang - xyzabc

- swing - xframe

找到较多的一层，切出来即可（目前“较多”的判断为：兄弟结点大于2）

最终两两比较结果数据为：

label	value	Value type	
exception	1	1/0	异常是否相同
field_isdup	1	1/0	领域是否为互相包含关系
field_dup_index	0.888	0~1	领域重合率
field_deep	0.6	0~1	领域的深度比例
field_length	0.6	0~1	领域的长度比例
field_interest_issame	1	0/1	核心领域是否相同
field_interest_length	1	0~1	核心领域的长度（call数量）比例
callstack_inner	0.9	0~1	核心领域调用链相似度
callstack_outer_1	0.9	0~1	核心领域上方调用链相似度
callstack_outer_2	0.9	0~1	核心领域下方调用链相似度
callstack_all	0.9	0~1	总体调用链相似度

最后再根据上述信息计算最终相似度（即两个堆栈的相似度）

目前未完成处：核心领域相似度未计算，最终相似度未计算，其余已完成。