

```

1: =====
2:
3: Smalltalk.KSU defineClass: #Workbench
4:   superclass: #{UI.ApplicationModel}
5:   indexedType: #none
6:   private: false
7:   instanceVariableNames: 'isInheritance isMeta classTabList classlist methodTabList methodList codeText '
8:   classInstanceVariableNames: ''
9:   imports: ''
10:  category: 'KSU-Tool'
11:
12: =====
13:
14: KSU.Workbench method for 'menu messages'
15:
16: acceptText: aText from: aController
17:
18: | aCode aBlock |
19: aText isEmpty ifTrue: [^nil].
20: aCode := aText asString.
21: aBlock :=
22:     [| compilerClass |
23:      compilerClass := Kernel.Compiler.
24:      self selectedClass
25:      ifNotNil:
26:          [:aClass |
27:           self selectedSelector
28:           ifNil: [compilerClass evaluate: aCode]
29:           ifNotNil:
30:               [:aSelector |
31:                | aProtocol aSymbol |
32:                aProtocol := aClass organization categoryOfElement: aSelector.
33:                aSymbol := aClass compile: aCode classified: aProtocol notifying: aController.
34:                aSymbol = aSelector iffFalse: [self setSelector: aSymbol]].
35:                self codeText]].
36: aBlock on: Object errorSignal
37:   do: [:anException | ^Dialog warn: '登録エラー: ', anException messageText asString]
38:
39: -----
40:
41: KSU.Workbench method for 'private'
42:
43: adjustScrollAmount: selectionInList
44:
45: (selectionInList listHolder dependents detect: [:aView | aView isKindOfClass: SequenceView]
46:  ifNone: [^nil])
47:  ifNotNil:
48:      [:aView |
49:       (aView boundsOfElementIndex: aView selectionIndex inBounds: aView bounds) bottom
50:        > aView bounds bottom ifTrue: [aView scrollBy: 0 @ aView lineGrid negated]]
51:
52: -----
53:
54: KSU.Workbench method for 'aspects'
55:
56: classList
57:
58: | theClass theSelector alist aString |
59: self class
60:   assert:
61:     [theClass := self selectedClass.
62:      theSelector := self selectedSelector]
63:   do:
64:     [alist := List new.
65:      isInheritance
66:      ifTrue:
67:          [self class allClassesDo:
68:            [:aClass |
69:             aString := self class toolListDisplayString: aClass withIndent: true.
70:             alist add: aString]]
71:         iffFalse:
72:             [(self class allClasses asSortedCollection: self class defaultSortBlock) do:
73:              [:aClass |
74:               aString := self class toolListDisplayString: aClass.
75:               alist add: aString]].
76:             classList list: alist]
77:     ensure: [self setClass: theClass andSelector: theSelector].
78: ^classList
79:
80: -----
81:
82: KSU.Workbench method for 'aspects'
83:
84: classTabList
85:
86: ^classTabList
87:
88: -----
89:
90: KSU.Workbench method for 'aspects'
91:
92: codeText
93:
94: | aText |
95: self class
96:   assert: []
97:   do:
98:     [aText := Text new.
99:      self selectedClass
100:      ifNotNil:

```

```

101:         [:aClass |
102:         aText := self selectedSelector
103:             ifNil: [aClass definition asText]
104:             ifNotNil: [:aSelector | aClass sourceMethodAt: aSelector]].
105:         codeText value: aText]
106:     ensure: [].
107:     ^codeText
108:
109: -----
110:
111: KSU.Workbench method for 'menu messages'
112:
113: findClass
114:
115:     | aClass aString |
116:     aClass := self class findClass ifNil: [^nil].
117:     aString := self class toolListDisplayString: aClass withIndent: isInheritance.
118:     classList selection: aString.
119:     self adjustScrollAmount: classList.
120:     self methodList
121:
122: -----
123:
124: KSU.Workbench method for 'menu messages'
125:
126: findImplementors
127:
128:     | aClass aSelector |
129:     aClass := self selectedClass ifNil: [^nil].
130:     aSelector := self selectedSelector ifNil: [^nil].
131:     aSelector := self class selectSelectorIn: aClass withSelector: aSelector.
132:     aSelector ifNil: [^nil].
133:     self class browseImplementorsOf: aSelector
134:
135: -----
136:
137: KSU.Workbench method for 'menu messages'
138:
139: findImplementorsHierarchy
140:
141:     | aClass aSelector |
142:     aClass := self selectedClass ifNil: [^nil].
143:     aSelector := self selectedSelector ifNil: [^nil].
144:     aSelector := self class selectSelectorIn: aClass withSelector: aSelector.
145:     aSelector ifNil: [^nil].
146:     self class browseImplementorsOf: aSelector inHierarchy: aClass
147:
148: -----
149:
150: KSU.Workbench method for 'menu messages'
151:
152: findMethod
153:
154:     | aClass aMethod aSelector |
155:     aClass := self selectedClass ifNil: [^nil].
156:     aMethod := (self class findMethodIn: aClass) ifNil: [^nil].
157:     aSelector := aMethod selector.
158:     methodList selection: aSelector.
159:     self adjustScrollAmount: methodList.
160:     self codeText
161:
162: -----
163:
164: KSU.Workbench method for 'menu messages'
165:
166: findReferences
167:
168:     | aClass |
169:     (aClass := self selectedClass) ifNil: [^nil].
170:     self class findReferencesTo: aClass
171:
172: -----
173:
174: KSU.Workbench method for 'menu messages'
175:
176: findSenders
177:
178:     | aClass aSelector |
179:     aClass := self selectedClass ifNil: [^nil].
180:     aSelector := self selectedSelector ifNil: [^nil].
181:     aSelector := self class selectSelectorIn: aClass withSelector: aSelector.
182:     aSelector ifNil: [^nil].
183:     self class browseSendersOf: aSelector
184:
185: -----
186:
187: KSU.Workbench method for 'menu messages'
188:
189: findSendersHierarchy
190:
191:     | aClass aSelector |
192:     aClass := self selectedClass ifNil: [^nil].
193:     aSelector := self selectedSelector ifNil: [^nil].
194:     aSelector := self class selectSelectorIn: aClass withSelector: aSelector.
195:     aSelector ifNil: [^nil].
196:     self class browseSendersOf: aSelector inHierarchy: aClass
197:
198: -----
199:
200: KSU.Workbench method for 'menu messages'

```

```

201:
202: formatText: aText from: aController
203:
204: | aCode aBlock |
205: aText isEmpty ifTrue: [^nil].
206: aCode := aText asString.
207: aBlock :=
208:     [| parserClass |
209:     parserClass := Refactory.Browser.RBParser.
210:     self selectedClass
211:     ifNotNil:
212:         [:aClass |
213:         aCode := self selectedSelector
214:             ifNil: [(parserClass parseExpression: aCode) formattedCode asText]
215:             ifNotNil: [(parserClass parseMethod: aCode) formattedCode asText makeSelectorBoldIn: aClass].
216:         self codeText value: aCode]].
217: aBlock on: Object errorSignal
218:     do: [:anException | ^Dialog warn: '整形エラー: ', anException messageText asString]
219:
220: -----
221:
222: KSU.Workbench method for 'initialize-release'
223:
224: initialize
225:
226: super initialize.
227: isInheritance := false.
228: isMeta := false.
229: classTabList := SelectionInList new.
230: classList := SelectionInList new.
231: methodTabList := SelectionInList new.
232: methodList := SelectionInList new.
233: codeText := ValueHolder with: Text new.
234: ^self
235:
236: -----
237:
238: KSU.Workbench method for 'aspects'
239:
240: methodList
241:
242: | theSelector aList |
243: self class
244:     assert: [theSelector := self selectedSelector]
245:     do:
246:         [aList := List new.
247:         self selectedClass
248:             ifNotNil:
249:                 [:aClass |
250:                 aClass selectors asSortedCollection do: [:aSelector | aList add: aSelector]].
251:         methodList list: aList]
252:     ensure: [self setSelector: theSelector].
253: self codeText.
254: ^methodList
255:
256: -----
257:
258: KSU.Workbench method for 'aspects'
259:
260: methodTabList
261:
262: ^methodTabList
263:
264: -----
265:
266: KSU.Workbench method for 'menu messages'
267:
268: openSourceCode
269:
270: | aClass aString aModel aWindow |
271: aClass := self selectedClass ifNil: [^nil].
272: aString := self class codePrintStringOf: aClass.
273: aModel := ValueHolder with: aString.
274: aWindow := TextEditorView
275:     createOn: aModel
276:     label: aClass name
277:     icon: Icon default
278:     controller: TextEditorController new.
279: aWindow openIn: self class defaultDisplayBox
280:
281: -----
282:
283: KSU.Workbench method for 'interface opening'
284:
285: postOpenWith: aBuilder
286:
287: super postOpenWith: aBuilder.
288: classTabList selectionHolder compute:
289:     [:aValue |
290:     isInheritance := aValue = 'インヘリタンス'.
291:     self classList].
292: classList selectionHolder compute: [:aValue | self methodList].
293: methodTabList selectionHolder compute:
294:     [:aValue |
295:     isMeta := aValue = 'クラスメソッド'.
296:     self methodList].
297: methodList selectionHolder compute: [:aValue | self codeText]
298:
299: -----
300:

```

```

301: KSU.Workbench method for 'accessing'
302:
303: selectedClass
304:
305: | aString aCollection aClass |
306: (aString := classList selection) ifNil: [^nil].
307: aCollection := JunStringUtility separate: aString dividers: '| {}'.
308: aString := aCollection size > 1
309:     ifTrue: [aCollection last , '.' , aCollection first]
310:     ifFalse: [aCollection first].
311: (aClass := Compiler evaluate: aString) ifNil: [^nil].
312: isMeta ifTrue: [aClass := aClass class].
313: ^aClass
314:
315: -----
316:
317: KSU.Workbench method for 'accessing'
318:
319: selectedSelector
320:
321: ^methodList selection
322:
323: -----
324:
325: KSU.Workbench method for 'private'
326:
327: setClass: theClass andSelector: theSelector
328:
329: theClass
330:     ifNotNil:
331:         [| aClass aString |
332:          aClass := theClass isMeta ifTrue: [theClass soleInstance] ifFalse: [theClass].
333:          aString := self class toolListDisplayString: aClass withIndent: isInheritance.
334:          classList selection: aString.
335:          self adjustScrollAmount: classList].
336: self setSelector: theSelector
337:
338: -----
339:
340: KSU.Workbench method for 'private'
341:
342: setSelector: theSelector
343:
344: theSelector
345:     ifNotNil:
346:         [self selectedClass
347:          ifNotNil:
348:              [:aClass |
349:               (aClass selectors includes: theSelector)
350:               ifTrue:
351:                   [methodList selection: theSelector.
352:                    self adjustScrollAmount: methodList]]]
353:
354: =====
355:
356: KSU.Workbench class
357:     instanceVariableNames: ''
358:
359: =====
360:
361: KSU.Workbench class method for 'accessing'
362:
363: allClasses
364:     "KSU.Workbench allClasses."
365:
366: | aList |
367: aList := List new.
368: self allClassesDo: [:aClass | aList add: aClass].
369: ^aList
370:
371: -----
372:
373: KSU.Workbench class method for 'accessing'
374:
375: allClassesDo: aBlock
376:     "KSU.Workbench allClassesDo: [:aClass | Transcript cr; show: (KSU.Workbench toolListDisplayString: aClass)]."
377:
378: | aClosure aStack |
379: aClosure := self defaultSortBlock.
380: aStack := OrderedCollection new: 128.
381: aStack addAll: ((Class rootsOfTheWorld asSortedCollection: aClosure) reverse.
382: [aStack isEmpty] whileFalse:
383:     [| aClass |
384:      aClass := aStack removeLast.
385:      aBlock value: aClass.
386:      aStack addAll: ((aClass subclasses select: [:each | each isMeta not]) asSortedCollection: aClosure) reverse]
387:
388: -----
389:
390: KSU.Workbench class method for 'evaluating'
391:
392: assert: assertBlock do: doBlock ensure: ensureBlock
393:
394: | assertValue returnValue |
395: assertValue := assertBlock value.
396: returnValue := nil.
397: [returnValue := doBlock cull: assertValue] ensure: [ensureBlock cull: assertValue cull: returnValue].
398: ^returnValue
399:
400: -----

```

```

401:
402: KSU.Workbench class method for 'utilities'
403:
404: browseClass: aClass withSelector: aSelector
405: "KSU.Workbench browseClass: Integer withSelector: #abs."
406: "KSU.Workbench browseClass: Integer withSelector: nil."
407:
408: | aPackage aProtocol aBrowser aWindow aNavigator |
409: aPackage := aClass package.
410: aProtocol := aClass organization categoryOfElement: aSelector.
411: aBrowser := #{Refractory.Browser.RefactoringBrowser} value open.
412: aWindow := aBrowser builder window.
413: aWindow displayBox: self defaultDisplayBox.
414: aNavigator := aBrowser navigator.
415: aNavigator selectPundle: aPackage.
416: (aNavigator state)
417:   classesAndNameSpaces: (Array with: aClass);
418:   protocols: (Array with: aProtocol);
419:   selectors: (Array with: aSelector).
420: aNavigator
421:   setState: aNavigator state;
422:   changed.
423: ^aBrowser
424:
425: -----
426:
427: KSU.Workbench class method for 'utilities'
428:
429: browseImplementorsOf: aSelector
430: "KSU.Workbench browseImplementorsOf: #abs."
431:
432: ^self browseImplementorsOf: aSelector inHierarchy: nil
433:
434: -----
435:
436: KSU.Workbench class method for 'utilities'
437:
438: browseImplementorsOf: aSelector inHierarchy: aClass
439: "KSU.Workbench browseImplementorsOf: #abs inHierarchy: Integer."
440:
441: | aCollector aFilter aCollection aBrowser aWindow |
442: aCollector := MethodCollector new.
443: aFilter := aCollector implementorsOf: aSelector.
444: aClass ifNotNil: [aFilter := (aCollector searchClassHierarchy: aClass) & aFilter].
445: aCollection := aCollector select: aFilter.
446: aCollection isEmpty ifTrue: [^Dialog warn: '実装しているメソッドがありませんでした'].
447: aBrowser := aCollector
448:   openListBrowserOn: aCollection
449:   label: aFilter displayString
450:   initialSelection: aFilter initialSelection.
451: aWindow := aBrowser builder window.
452: aWindow displayBox: self defaultDisplayBox.
453: ^aBrowser
454:
455: -----
456:
457: KSU.Workbench class method for 'utilities'
458:
459: browseSendersOf: aSelector
460: "KSU.Workbench browseSendersOf: #abs."
461:
462: ^self browseSendersOf: aSelector inHierarchy: nil
463:
464: -----
465:
466: KSU.Workbench class method for 'utilities'
467:
468: browseSendersOf: aSelector inHierarchy: aClass
469: "KSU.Workbench browseSendersOf: #abs inHierarchy: Integer."
470:
471: | aCollector aFilter aCollection aBrowser aWindow |
472: aCollector := MethodCollector new.
473: aFilter := aCollector referencesTo: aSelector.
474: aClass ifNotNil: [aFilter := (aCollector searchClassHierarchy: aClass) & aFilter].
475: aCollection := aCollector select: aFilter.
476: aCollection isEmpty ifTrue: [^Dialog warn: '送信しているメソッドがありませんでした'].
477: aBrowser := aCollector
478:   openListBrowserOn: aCollection
479:   label: aFilter displayString
480:   initialSelection: aFilter initialSelection.
481: aWindow := aBrowser builder window.
482: aWindow displayBox: self defaultDisplayBox.
483: ^aBrowser
484:
485: -----
486:
487: KSU.Workbench class method for 'resources'
488:
489: classListMenu
490: "Tools.MenuEditor new openOnClass: self andSelector: #classListMenu"
491:
492: <resource: #menu>
493: ^#{UI.Menu} #{
494:   #{UI.MenuItem}
495:     #rawLabel: 'クラスの検索...'
496:     #value: #findClass )
497:   #{UI.MenuItem}
498:     #rawLabel: 'クラスへの参照...'
499:     #value: #findReferences )
500:   #{UI.MenuItem}

```

```

501:         #rawLabel: 'ソースコード'
502:         #value: #openSourceCode ) #(1 1 1) nil ) decodeAsLiteralArray
503:
504: -----
505:
506: KSU.Workbench class method for 'utilities'
507:
508: codePrintStringOf: theClass
509:     "KSU.Workbench codePrintStringOf: KSU.Workbench."
510:
511: | aString |
512: self
513:     assert: [(String new: 1024) writeStream]
514:     do:
515:         [:aStream |
516:             self codePrintStringOf: theClass on: aStream.
517:             aString := aStream contents]
518:     ensure: [:aStream | (aStream respondsTo: #close) ifTrue: [aStream close]].
519: ^aString
520:
521: -----
522:
523: KSU.Workbench class method for 'utilities'
524:
525: codePrintStringOf: theClass on: theStream
526:     "KSU.Workbench codePrintStringOf: KSU.Workbench on: (Transcript clear; yourself)."
527:
528: | aBlock aString |
529: aBlock :=
530:     [:aClass :aStream |
531:         | anOrganization aBoolean |
532:         80 timesRepeat: [aStream nextPutAll: '='].
533:         2 timesRepeat: [aStream cr].
534:         aStream nextPutAll: aClass definition.
535:         2 timesRepeat: [aStream cr].
536:         anOrganization := aClass organization.
537:         aBoolean := true.
538:         aClass selectors asSortedCollection do:
539:             [:aSelector |
540:                 | aCode |
541:                 aBoolean
542:                 ifTrue:
543:                     [80 timesRepeat: [aStream nextPutAll: '='].
544:                     2 timesRepeat: [aStream cr].
545:                     aBoolean := false]
546:                 ifFalse:
547:                     [60 timesRepeat: [aStream nextPutAll: '-'].
548:                     2 timesRepeat: [aStream cr]].
549:                 aStream
550:                     nextPutAll: aClass fullName;
551:                     nextPutAll: ' method for ';
552:                     nextPutAll: (anOrganization categoryOfElement: aSelector) asString printString.
553:                     2 timesRepeat: [aStream cr].
554:                     aCode := aClass sourceMethodAt: aSelector.
555:                     aStream nextPutAll: aCode.
556:                     2 timesRepeat: [aStream cr]]].
557:
558: self
559:     assert: [(String new: 1024) writeStream]
560:     do:
561:         [:aStream |
562:             | aClass |
563:             aClass := theClass isMeta ifTrue: [theClass soleInstance] ifFalse: [theClass yourself].
564:             aBlock value: aClass value: aStream.
565:             aBlock value: aClass class value: aStream.
566:             80 timesRepeat: [aStream nextPutAll: '='].
567:             aStream cr.
568:             aStream
569:                 nextPutAll: 'This document was generated by ', self fullName;
570:                 nextPutAll: ' on ', JunCalendarModel stringFromDate;
571:                 nextPutAll: ' at ', JunCalendarModel stringFromTime;
572:                 nextPutAll: ', '.
573:             aStream cr.
574:             aString := aStream contents]
575:     ensure: [:aStream | (aStream respondsTo: #close) ifTrue: [aStream close]].
576: theStream nextPutAll: aString.
577: ^aString
578:
579: -----
580:
581: KSU.Workbench class method for 'resources'
582:
583: codeTextMenu
584:     "Tools.MenuEditor new openOnClass: self andSelector: #codeTextMenu"
585:
586: <resource: #menu>
587: ^#{UI.Menu} #(
588:     #{UI.MenuItem}
589:     #rawLabel: '登録'
590:     #value:
591:     #acceptText:from: )
592:     #{UI.MenuItem}
593:     #rawLabel: '整形'
594:     #value:
595:     #formatText:from: )
596:     #{UI.MenuItem}
597:     #rawLabel: '実行'
598:     #value: #doIt )
599:     #{UI.MenuItem}
600:     #rawLabel: '検査'
601:     #value: #inspectIt )

```

```

601:         #({UI.MenuItem}
602:         #rawLabel: 'カット'
603:         #value: #cut )
604:         #({UI.MenuItem}
605:         #rawLabel: 'コピー'
606:         #value: #copySelection )
607:         #({UI.MenuItem}
608:         #rawLabel: 'ペースト'
609:         #value: #paste ) ) #(2 2 3 ) nil ) decodeAsLiteralArray
610:
611: -----
612:
613: KSU.Workbench class method for 'defaults'
614:
615: defaultDisplayBox
616: "KSU.Workbench defaultDisplayBox."
617:
618: | aBox aRectangle |
619: aBox := 0 @ 0 extent: 800 @ 600.
620: aRectangle := Screen default bounds.
621: aBox := aBox align: aBox center with: aRectangle center.
622: ^aBox
623:
624: -----
625:
626: KSU.Workbench class method for 'defaults'
627:
628: defaultSortBlock
629: "KSU.Workbench defaultSortBlock value: Text value: XML.Text."
630:
631: | aBlock |
632: aBlock :=
633:     [:classA :classB |
634:     | nameA nameB |
635:     (nameA := classA name) = (nameB := classB name)
636:     ifTrue: [classA toolListDisplayString < classB toolListDisplayString]
637:     ifFalse: [nameA < nameB]].
638: ^aBlock
639:
640: -----
641:
642: KSU.Workbench class method for 'examples'
643:
644: example
645: "KSU.Workbench example."
646:
647: | aWorkBench |
648: aWorkBench := KSU.Workbench new.
649: aWorkBench open.
650: ^aWorkBench
651:
652: -----
653:
654: KSU.Workbench class method for 'examples'
655:
656: example0
657: "KSU.Workbench example0."
658:
659: | aList aString aModel aWindow |
660: aList := List new.
661: (KSU.Workbench allClasses asSortedCollection: KSU.Workbench defaultSortBlock) do:
662:     [:aClass |
663:     aString := KSU.Workbench toolListDisplayString: aClass.
664:     aList add: aString].
665: KSU.Workbench
666:     assert: [(String new: 1024) writeStream]
667:     do:
668:         [:aStream |
669:         aList do:
670:             [:each |
671:             aStream
672:                 nextPutAll: each;
673:                 cr].
674:             aString := aStream contents]
675:     ensure: [:aStream | (aStream respondsTo: #close) ifTrue: [aStream close]].
676: aModel := ValueHolder with: aString.
677: aWindow := TextEditorView
678:     createOn: aModel
679:     label: 'クラスリスト'
680:     icon: Icon default
681:     controller: TextEditorController new.
682: aWindow openIn: Workbench defaultDisplayBox.
683: ^aString
684:
685: -----
686:
687: KSU.Workbench class method for 'examples'
688:
689: example1
690: "KSU.Workbench example1."
691:
692: | aList aString aModel aWindow |
693: aList := List new.
694: KSU.Workbench allClassesDo:
695:     [:aClass |
696:     aString := KSU.Workbench toolListDisplayString: aClass withIndent: true.
697:     aList add: aString].
698: KSU.Workbench
699:     assert: [(String new: 1024) writeStream]
700:     do:

```

```

701:         [:aStream |
702:         alist do:
703:             [:each |
704:             aStream
705:                 nextPutAll: each;
706:                 cr].
707:             aString := aStream contents]
708:         ensure: [:aStream | (aStream respondsTo: #close) ifTrue: [aStream close]].
709:         aModel := ValueHolder with: aString.
710:         aWindow := TextViewView
711:             createOn: aModel
712:             label: 'インヘリタンス'
713:             icon: Icon default
714:             controller: TextEditorController new.
715:         aWindow openIn: Workbench defaultDisplayBox.
716:         ^aString
717:
718: -----
719:
720: KSU.Workbench class method for 'examples'
721:
722: example2
723:     "KSU.Workbench example2."
724:
725:     | aClass aString aModel aWindow |
726:     aClass := KSU.Workbench.
727:     aString := KSU.Workbench codePrintStringOf: aClass.
728:     aModel := ValueHolder with: aString.
729:     aWindow := TextViewView
730:         createOn: aModel
731:         label: aClass name
732:         icon: Icon default
733:         controller: TextEditorController new.
734:     aWindow openIn: KSU.Workbench defaultDisplayBox
735:
736: -----
737:
738: KSU.Workbench class method for 'utilities'
739:
740: findClass
741:     "KSU.Workbench findClass."
742:
743:     | aCollection aSelector aBlock aDialog theClass |
744:     aCollection := self allClasses asSortedCollection: self defaultSortBlock.
745:     aSelector := #toolListDisplayString.
746:     aBlock :=
747:         [:aPattern |
748:         aPattern isEmpty
749:             ifTrue: [aCollection yourself]
750:             ifFalse:
751:                 [aCollection select: [:aClass | aPattern , '*' match: (aClass perform: aSelector)]]].
752:     aDialog := IncrementalSearchDialog forSelectionWithSearch: aBlock.
753:     aDialog
754:         windowLabel: 'クラスを検索する';
755:         firstLabel: '入力した名前からクラスを探します';
756:         secondLabel: '候補として見つかったクラス';
757:         displayStringSelector: aSelector.
758:     theClass := aDialog select ifNil: [^nil].
759:     ^theClass
760:
761: -----
762:
763: KSU.Workbench class method for 'utilities'
764:
765: findMethodIn: theClass
766:     "KSU.Workbench findMethodIn: Integer."
767:
768:     | aCollection aBlock aDialog theSelector |
769:     aCollection := theClass ifNil: [^nil] ifNotNil: [:aClass | aClass selectors asSortedCollection].
770:     aBlock :=
771:         [:aPattern |
772:         aPattern isEmpty
773:             ifTrue: [aCollection yourself]
774:             ifFalse: [aCollection select: [:aSelector | aPattern , '*' match: aSelector]]].
775:     aDialog := IncrementalSearchDialog forSelectionWithSearch: aBlock.
776:     aDialog
777:         windowLabel: 'メソッドを検索する';
778:         firstLabel: '入力した名前からメソッドを探します';
779:         secondLabel: '候補として見つかったメソッド'.
780:     theSelector := aDialog select ifNil: [^nil].
781:     ^theClass compiledMethodAt: theSelector
782:
783: -----
784:
785: KSU.Workbench class method for 'utilities'
786:
787: findReferencesTo: theClass
788:     "KSU.Workbench findReferencesTo: String."
789:
790:     | aClass aReference aCollector aFilter aCollection aBrowser aWindow |
791:     aClass := theClass isMeta ifTrue: [theClass soleInstance] ifFalse: [theClass].
792:     aReference := LiteralBindingReference pathString: aClass fullName.
793:     aCollector := MethodCollector new.
794:     aFilter := aCollector referencesTo: aReference binding.
795:     aCollection := aCollector select: aFilter.
796:     aCollection isEmpty ifTrue: [^Dialog warn: '参照しているメソッドがありませんでした'].
797:     aBrowser := aCollector
798:         openListBrowserOn: aCollection
799:         label: aFilter displayString
800:         initialSelection: aFilter initialSelection.

```



```

801: aWindow := aBrowser builder window.
802: aWindow displayBox: self defaultDisplayBox.
803: ^aClass
804:
805: -----
806:
807: KSU.Workbench class method for 'utilities'
808:
809: findSelector
810: "KSU.Workbench findSelector."
811:
812: | aModule aDialog aSelector |
813: aModule := IncrementalSearchModule new.
814: aModule
815:   firstLabel: '入力した名前からセレクトを探します';
816:   secondLabel: '候補として見つかったセレクト';
817:   searchBlock: [:aString | IncrementalSearchModule findMethodSelectorsMatching: aString];
818:   objectToStringBlock: [:aSymbol | aSymbol asString];
819:   useEagerSelection: true;
820:   pasteSelected: false.
821: aDialog := (IncrementalSearchDialog new)
822:   module: aModule;
823:   requireSelection: true;
824:   windowLabel: 'セレクトを検索します'.
825: aSelector := aDialog select ifNil: [^nil].
826: ^aSelector
827:
828: -----
829:
830: KSU.Workbench class method for 'utilities'
831:
832: findSelectorIn: theClass
833: "KSU.Workbench findSelectorIn: Integer."
834:
835: | aCollection aBlock aDialog theSelector |
836: aCollection := theClass ifNil: [^nil] ifNotNil: [:aClass | aClass selectors asSortedCollection].
837: aBlock :=
838:   [:aPattern |
839:     aPattern isEmpty
840:       ifTrue: [aCollection yourself]
841:       ifFalse: [aCollection select: [:aSelector | aPattern , '*' match: aSelector]].
842: aDialog := IncrementalSearchDialog forSelectionWithSearch: aBlock.
843: aDialog
844:   windowLabel: 'セレクトを検索する';
845:   firstLabel: '入力した名前からセレクトを探します';
846:   secondLabel: '候補として見つかったセレクト'.
847: theSelector := aDialog select ifNil: [^nil].
848: ^theSelector
849:
850: -----
851:
852: KSU.Workbench class method for 'resources'
853:
854: methodListMenu
855: "Tools.MenuEditor new openOnClass: self andSelector: #methodListMenu"
856:
857: <resource: #menu>
858: ^#{UI.Menu} #{
859:   #{UI.MenuItem}
860:     #rawLabel: 'メソッドの検索...'
861:     #value: #findMethod )
862:   #{UI.MenuItem}
863:     #rawLabel: 'センダ (呼出者) の検索...'
864:     #value: #findSenders )
865:   #{UI.MenuItem}
866:     #rawLabel: 'レシーバ (実装者) の検索...'
867:     #value: #findImplementors )
868:   #{UI.MenuItem}
869:     #rawLabel: 'センダ (呼出者) の階層検索...'
870:     #value: #findSendersHierarchy )
871:   #{UI.MenuItem}
872:     #rawLabel: 'レシーバ (実装者) の階層検索...'
873:     #value: #findImplementorsHierarchy ) #(1 2 2 ) nil ) decodeAsLiteralArray
874:
875: -----
876:
877: KSU.Workbench class method for 'instance creation'
878:
879: new
880:
881:   ^(super new)
882:     initialize;
883:     yourself
884:
885: -----
886:
887: KSU.Workbench class method for 'utilities'
888:
889: promptThenBrowseClass
890: "KSU.Workbench promptThenBrowseClass."
891:
892: | aClass aBrowser aWindow aNavigator |
893: aClass := self findClass ifNil: [^nil].
894: aBrowser := #{Refactory.Browser.RefactoringBrowser} value open.
895: aWindow := aBrowser builder window.
896: aWindow displayBox: self defaultDisplayBox.
897: aNavigator := aBrowser navigator.
898: aNavigator selectPundle: aClass package.
899: aNavigator state classesAndNameSpaces: (Array with: aClass).
900: aNavigator

```

```

901:         setState: aNavigator state;
902:         changed.
903:     ^aClass
904:
905: -----
906:
907: KSU.Workbench class method for 'utilities'
908:
909: promptThenBrowseClassHierarchy
910:     "KSU.Workbench promptThenBrowseClassHierarchy."
911:
912:     | aClass aBrowser aWindow |
913:     aClass := self findClass ifNil: [^nil].
914:     aBrowser := #{Refractory.Browser.RefractoringBrowser} value open.
915:     aWindow := aBrowser builder window.
916:     aWindow displayBox: self defaultDisplayBox.
917:     (aBrowser navigator)
918:         selectClass: aClass;
919:         changed;
920:         beHierarchy.
921:     ^aBrowser
922:
923: -----
924:
925: KSU.Workbench class method for 'utilities'
926:
927: promptThenBrowseImplementors
928:     "KSU.Workbench promptThenBrowseImplementors."
929:
930:     | aSelector aCollector aFilter aCollection aBrowser aWindow |
931:     aSelector := self findSelector ifNil: [^nil].
932:     aCollector := MethodCollector new.
933:     aFilter := aCollector implementorsOf: aSelector.
934:     aCollection := aCollector select: aFilter.
935:     aCollection isEmpty ifTrue: [^Dialog warn: '実装しているメソッドがありませんでした'].
936:     aBrowser := aCollector
937:         openListBrowserOn: aCollection
938:         label: aFilter displayString
939:         initialSelection: aFilter initialSelection.
940:     aWindow := aBrowser builder window.
941:     aWindow displayBox: self defaultDisplayBox.
942:     ^aBrowser
943:
944: -----
945:
946: KSU.Workbench class method for 'utilities'
947:
948: promptThenBrowseImplementorsInHierarchy: aClass
949:     "KSU.Workbench promptThenBrowseImplementorsInHierarchy: Integer."
950:
951:     | aSelector |
952:     aSelector := (self findSelectorIn: aClass) ifNil: [^nil].
953:     ^self browseImplementorsOf: aSelector inHierarchy: aClass
954:
955: -----
956:
957: KSU.Workbench class method for 'utilities'
958:
959: promptThenBrowseSenders
960:     "KSU.Workbench promptThenBrowseSenders."
961:
962:     | aSelector aCollector aFilter aCollection aBrowser aWindow |
963:     aSelector := self findSelector ifNil: [^nil].
964:     aCollector := MethodCollector new.
965:     aFilter := aCollector referencesTo: aSelector.
966:     aCollection := aCollector select: aFilter.
967:     aCollection isEmpty ifTrue: [^Dialog warn: '送信しているメソッドがありませんでした'].
968:     aBrowser := aCollector
969:         openListBrowserOn: aCollection
970:         label: aFilter displayString
971:         initialSelection: aFilter initialSelection.
972:     aWindow := aBrowser builder window.
973:     aWindow displayBox: self defaultDisplayBox.
974:     ^aBrowser
975:
976: -----
977:
978: KSU.Workbench class method for 'utilities'
979:
980: promptThenBrowseSendersInHierarchy: aClass
981:     "KSU.Workbench promptThenBrowseSendersInHierarchy: Integer."
982:
983:     | aSelector |
984:     aSelector := (self findSelectorIn: aClass) ifNil: [^nil].
985:     ^self browseSendersOf: aSelector inHierarchy: aClass
986:
987: -----
988:
989: KSU.Workbench class method for 'utilities'
990:
991: selectSelectorIn: aClass withSelector: aSelector
992:     "KSU.Workbench selectSelectorIn: KSU.Workbench class withSelector: #selectSelectorIn:withSelector:."
993:
994:     | aMethod aCollection aMenu aResult |
995:     aClass ifNil: [^nil].
996:     aSelector ifNil: [^nil].
997:     aMethod := aClass compiledMethodAt: aSelector.
998:     aCollection := Set new.
999:     aMethod allSymbolLiteralsDo: [:aSymbol | aCollection add: aSymbol].
1000:     aCollection := aCollection asSortedCollection asOrderedCollection.

```

```

1001: aCollection addFirst: aSelector.
1002: aCollection := aCollection asArray.
1003: aMenu := Menu
1004:     labelArray: aCollection
1005:     lines: #(1)
1006:     values: aCollection.
1007: aResult := aMenu startUp.
1008: (aResult isKindOf: Symbol) ifFalse: [^nil].
1009: ^aResult
1010:
1011: -----
1012:
1013: KSU.Workbench class method for 'printing'
1014:
1015: toolListDisplayString: aClass
1016:
1017:     ^self toolListDisplayString: aClass withIndent: false
1018:
1019: -----
1020:
1021: KSU.Workbench class method for 'printing'
1022:
1023: toolListDisplayString: aClass withIndent: aBoolean
1024:
1025:     | aString |
1026:     self
1027:         assert: [(String new: 128) writeStream]
1028:         do:
1029:             [:aStream |
1030:                 aBoolean ifTrue: [aClass allSuperclasses size timesRepeat: [aStream nextPutAll: ' | ']].
1031:                 aStream nextPutAll: aClass toolListDisplayString.
1032:                 aString := aStream contents]
1033:             ensure: [:aStream | (aStream respondsTo: #close) ifTrue: [aStream close]].
1034:     ^aString
1035:
1036: -----
1037:
1038: KSU.Workbench class method for 'interface specs'
1039:
1040: windowSpec
1041:     "Tools.UIPainter new openOnClass: self andSelector: #windowSpec"
1042:
1043:     <resource: #canvas>
1044:     ^#{UI.FullSpec}
1045:     #window:
1046:         #{UI.WindowSpec}
1047:         #label: 'ワークベンチ'
1048:         #min: #{Core.Point} 400 300 )
1049:         #max: #{Core.Point} 0 0 )
1050:         #bounds: #{Graphics.Rectangle} 1280 720 2080 1320 )
1051:         #colors:
1052:         #{UI.LookPreferences}
1053:         #setForegroundColor: #{Graphics.ColorValue} #black )
1054:         #setBackgroundColors: #{Graphics.ColorValue} 7044 7044 7044 )
1055:         #setBorderColor: #{Graphics.ColorValue} #lightGray ) )
1056:     #component:
1057:     #{UI.SpecCollection}
1058:     #collection: #(
1059:         #{UI.TabControlSpec}
1060:         #layout: #{Graphics.LayoutFrame} 2 0 2 0 -1 0.5 -1 0.5 )
1061:         #name: #classTabList
1062:         #model: #classTabList
1063:         #labels: #('クラスリスト' 'インヘリタンス' ) )
1064:         #{UI.SequenceViewSpec}
1065:         #layout: #{Graphics.LayoutFrame} 2 0 26 0 -1 0.5 -1 0.5 )
1066:         #name: #classList
1067:         #colors:
1068:         #{UI.LookPreferences}
1069:         #setBackgroundColors: #{Graphics.ColorValue} #white ) )
1070:         #model: #classList
1071:         #menu: #classListMenu
1072:         #useModifierKeys: true
1073:         #selectionType: #highlight )
1074:         #{UI.TabControlSpec}
1075:         #layout: #{Graphics.LayoutFrame} 1 0.5 2 0 -2 1 -1 0.5 )
1076:         #name: #methodTabList
1077:         #model: #methodTabList
1078:         #tabable: true
1079:         #labels: #('インスタンスメソッド' 'クラスメソッド' ) )
1080:         #{UI.SequenceViewSpec}
1081:         #layout: #{Graphics.LayoutFrame} 1 0.5 26 0 -2 1 -1 0.5 )
1082:         #name: #methodList
1083:         #colors:
1084:         #{UI.LookPreferences}
1085:         #setBackgroundColors: #{Graphics.ColorValue} #white ) )
1086:         #model: #methodList
1087:         #menu: #methodListMenu
1088:         #useModifierKeys: true
1089:         #selectionType: #highlight )
1090:         #{UI.TextEditorSpec}
1091:         #layout: #{Graphics.LayoutFrame} 2 0 1 0.5 -2 1 -2 1 )
1092:         #name: #codeText
1093:         #colors:
1094:         #{UI.LookPreferences}
1095:         #setBackgroundColors: #{Graphics.ColorValue} #white ) )
1096:         #model: #codeText
1097:         #menu: #codeTextMenu
1098:         #tabRequiresControl: true )
1099:         #{UI.ResizingSplitterSpec}
1100:         #layout: #{Graphics.LayoutFrame} -1 0.5 2 0 1 0.5 -1 0.5 )

```

```
1101:         #name: #verticalResizingSplitter
1102:         #horizontal: false
1103:         #minAboveSize: 200
1104:         #minBelowSize: 200
1105:         #aboveWidgets: 'classTabList classList '
1106:         #belowWidgets: 'methodTabList methodList' )
1107:     #({UI.ResizingSplitterSpec}
1108:     #layout: #({Graphics.LayoutFrame} 2 0 -1 0.5 -2 1 1 0.5 )
1109:     #name: #horizontalResizingSplitter
1110:     #horizontal: true
1111:     #minAboveSize: 100
1112:     #minBelowSize: 100
1113:     #aboveWidgets: 'classTabList classList methodTabList methodList verticalResizingSplitter'
1114:     #belowWidgets: 'codeText' ) ) )
1115: =====
1117: This document was generated by KSU.Workbench on 2011/11/26 at 05:38:54.
```