Survey on Multi-language Design Smells

Survey on Multi-language Design Smells

Thank you for agreeing to participate, it will take around 30 minutes to complete.

Study Policy:

- Participation in this study is completely voluntary. If you decide not to participate there will not be any negative consequences. If you
 decide to participate, you may stop participating at any time and withdraw entirely your participation or you may decide not to answer
 any specific question.
- Your identity and the data collected thanks to your participation will remain anonymous and will never be released to the public. Only
 anonymous data (aggregated or not) will be published in scientific articles, ensuring that the data cannot be linked back to a particular
 participant. The data will be kept by the principal investigator for five years before being destroyed.
- By submitting this survey, you are indicating that you have read the description of the study, are over the age of 18, and that you agree
 to the terms and consent as described in https://drive.google.com/file/d/1aZfHRCr0bEX0i331_oQHIS9ui9h6rlC5/view?usp=sharing

If you have any questions, please contact us at mouna.abidi@polymtl.ca

<u>Study Design:</u> The purpose of this study is to investigate the prevalence of design smells related to multi-language systems. These systems are developed using more than one programming language. We aim to investigate the perceived prevalence and impact of the design smells detailed below. Our main goal is to improve the quality of those systems.

Definition of terminologies:

Not Handling Exceptions	s The exceptions are not handled, developers generally rely on the exceptions provided by the other language
Assuming Safe Return	A value is returned to the other language without being checked. Thus, the interaction between both languages may
Value	not be correctly performed
Excessive Inter-language	eA wrong partitioning in both languages leads to many calls in a way or the other. It adds complexity takes more time
Communication	to run and may indicate a bad separation of concerns
Too Much Clustering	The multi-language code is concentrated in a few classes, regardless of their concerns and responsibilities.
Too Much Scattering	Many classes are scarcely used in multi-language communication
	When different libraries are needed depending on the operating system, they are not loaded with conditions on the
Hard Coding Libraries	operating system, but for instance, with a try-catch mechanism, making it hard to know which library has really been
	loaded
Local References Abuse	The developer does not manage the memory in the native space properly and does not release local and global
Local Neterences Abuse	references
Memory Management	Reference types passed from one language to another are not released in a language that does not handle the
Mismatch	management of memory causing memory leaks
Not Caching Objects	A method is called to retrieve a field every time this field is needed, although the field's ID or value could have been cached.
Not Securing Libraries	The code loads a foreign library without any security check or restriction privilege
0	A library is loaded using only the name not the path. It cannot be accessed in the same way from everywhere
_	A whole object is passed as an argument, although only some of the fields were needed, and it would have been
Excessive Objects	better for the system performance to pass only these fields
Unused Method	
Declaration	A method is declared in the host language but not implemented in the foreign language
Unused Method	A method is declared in the host language and implemented in the foreign language, but never called from the host
Implementation	language
Unused Parameters	Some arguments of a function are used neither in its body nor in the other language.

(Khomh, F., & Gueheneuce, Y. G. (2008, April). Do design patterns impact software quality positively? In Software Maintenance and Reengineering, 2008. CSMR 2008. 12th European Conference on (pp. 274-278).

IEEE.)		
– Exp	pandability: The degree to which the design o	of a system can be extended.
- Siı	mplicity: The degree to which the design of a	system can be understood easily.
- Re	usability: The degree to which a piece of des	sign can be reused in another design.
– Lea	arnability: The degree to which the code soul	rce of a system is easy to learn.
– Un	derstandability: The degree to which the cod	le source can be understood easily.
	formance: The degree to which the code me	
	-	
- 1010	dulanty. The degree to which the implement	ation of the functions of a system is independent of one another.
Than	k you.	
Best	regards,	
* 1.	What is your role within your organi (Yamashita, A., & Moonen, L. (2013, October). Do o Conference on (pp. 242-251). IEEE.)	ization? developers care about code smells? an exploratory survey. In Reverse Engineering (WCRE), 2013 20th Working
	Software Engineer	Opeveloper
	○ Team Lead	○ Tester
	O Architect	QA Manager
	O Project Manager	○ Self-employed
	Other, please specify	
* 2.	How many years of experience do y	ou have in software engineering?
	Less than 1 year	1 year - 5 Years
	○ 5-10 Years	O More than 10 years
* 3.	What is the domain of activity of you	ur organization?
	Research and development	○ Networks
	Healthcare	Analytics (Business,IT services, BigData)
	Banking and insurance	Robotics and Embeeded systems

* 4. What is your level of skill in the following languages? Please specify which other languages if relevant:

Other, please specify

(https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages)

Games

Other, please specify

	1 Novice	Little Knowledge	3 Practical	4 Comfortable	5 Expert
Python	0	0			
C++					
Java					
C					
C#					
PHP					
R					
JavaScript					
Go					
Assembly					
Other, please specify					

* 5. How often do you encounter the following design smells in your project(s)?

Please check the definitions provided above before answering this questions

	1 Very Often	2 Often	3 Rarely	N/A
Not Handling Exceptions	0			0
Assuming Safe Return Value			0	
Excessive Inter-language Communication				
Too Much Clustering				
Too Much Scattering			0	
Hard Coding Libraries				
Local References Abuse			0	
Memory Management Mismatch			0	
Not Caching Objects			0	
Not Securing Libraries			0	
Not Using Relative Path			0	
Excessive Objects				
Unused Method Declaration			0	
Unused Method Implementation			0	

Unused Parameters		

* 6. How do you evaluate the impact of the following design smells in those software quality attributes?

Please carefully read the definition of the smells provided bellow and the reference provided.

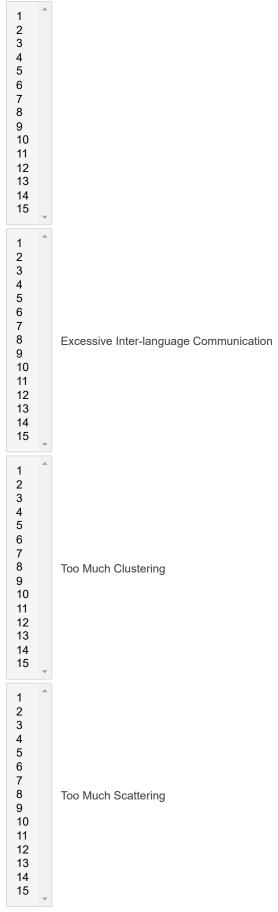
(VN: Very Negative, N: Negative, NS: Not significant/Neutral, P: Positive, and VP: Very Positive)

	Expandability	Simplicity	Reusability	Learnability	Understandabilit	y Performance	Modularity	N/A
Not Handling Exceptions								
Assuming Safe Return Value								
Excessive Inter-language Communication								
Too Much Clustering								
Too Much Scattering								
Hard Coding Libraries								
Local References Abuse								
Memory Management Mismatch								
Not Caching Objects								
Not Securing Libraries								
Not Using Relative Path								
Excessive Objects								
Unused Method Declaration								
Unused Method Implementation								
Unused Parameters								

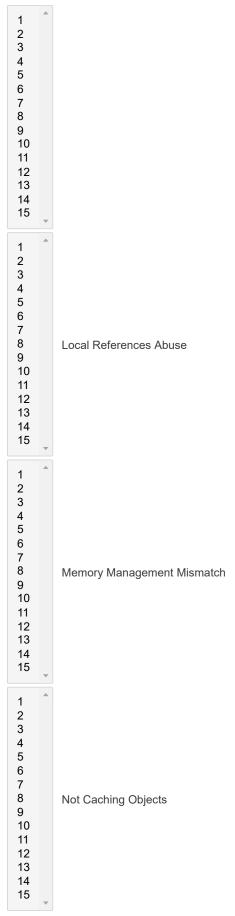
* 7. Please rank the following design smells from the most harmful to the less harmful

(Most harmful to the less harmful: 15 -> 1)

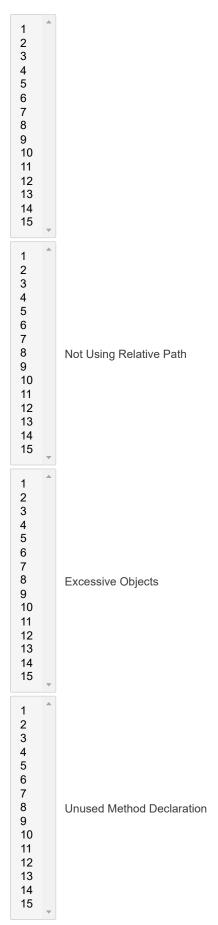
Assuming Safe Return Value



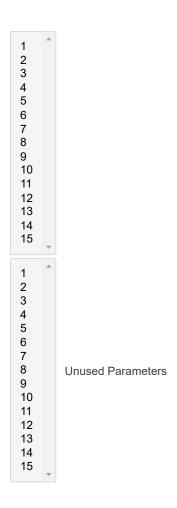
Hard Coding Libraries



Not Securing Libraries



Unused Method Implementation



* 8. <u>Task:</u>

a) In your opinion, does the following code(s) contain any occurrence of design smell(implementation and-or design problem)?

O No

```
void JNICALL Java_i2jrt_TAOObject__1release(JNIEnv *jni, jobject jThis)
{
    jclass clazz = findClass(jni, "i2jrt/TAOObject");
    jfieldID fid = jni->GetFieldID(clazz, "_jni_ptr", "J");
    jlong _jni_ptr = jni->GetLongField(jThis, fid);
    CORBA::Object_ptr o = reinterpret_cast(_jni_ptr);
    CORBA::release(o);
    jni->SetLongField(jThis, fid, reinterpret_cast(CORBA::Object::_nil()));
}
Yes
```

- 9. b) If YES, please provide an explanation or specify the design smell(s) involved?
- 10. c) If YES, (In your opinion,) What is the motivation behind using this specific way of implementation?
- * 11. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High)



* 12. e) If YES, would you apply this refactored solution?

```
void JNICALL Java_i2jrt_TAOObject__1release(JNIEnv *jni, jobject jThis)
{
jclass clazz = findClass(jni, "i2jrt/TAOObject");
if (clazz == NULL) {
jclass newExc = env->FindClass("java/lang/NullPointerException");
env->ThrowNew(newExc, "The native object does not exist.");
return 0;}
jfieldID fid = jni->GetFieldID(clazz, "_jni_ptr", "J");
jlong _jni_ptr = jni->GetLongField(jThis, fid);
CORBA::Object_ptr o = reinterpret_cast(_jni_ptr);
CORBA::release(o);
jni->SetLongField(jThis, fid, reinterpret_cast(CORBA::Object::_nil()));
}
Yes (Refactor with this solution)
                                                                      Yes (Refactor with an alternative solution)
No (No refactoring)
```

* 13. Task:

a) In your opinion, does the following code(s) contain any occurrence of design smell(implementation and-or design problem)?

```
public static char convertToChar(StreamItem item) throws MessageFormatException {
   if (item == null) throw new NullPointerException();
   final ItemKind itemKind = item.discriminator();
   if (compare(itemKind, ItemKind.CHAR_KIND)) {
     return item.charValue();
   } else {
     throw new MessageFormatException("Cannot convert stream item to char");
   }
}
Yes
No
```

14. b) If YES, please provide an explanation or specify the design smell(s) involved?

15. c) If YES, (In your opinion,) What is the motivation behind using this specific way of implementation?

* 16.	d) Please rate the	severity of the imple	ementation problem	(if any), from 1 (Ve	ry Low) to 5 (Very H	ligh)
	1 Very Low	2 Low	3 Medium	4 High	5 Very High	N/A
* 17.		ou apply this refacto		ion {		
	<pre>if (item == null) throw new final ItemKind itemKind = it if (ItemKind !== null)){</pre>	,				
	if (compare(itemKind, Item return item.charValue();	Kind.CHAR_KIND)) {				
	<pre>}} else { throw new MessageForma }</pre>	atException("Cannot convert	stream item to char");			
	}					
	 Yes (Refactor with 	this solution)	○ Yes	(Refactor with an alterr	native solution)	
	 No (No refactoring 	j)				
* 18.	Task:					
	design problem)?	does the following :Merge(WriteBatch* b, uint3; :sLength(key, value);				ntation and-or
	○ Yes			○ No		
	○ res			○ NO		
19. l	b) If YES, please pro	ovide an explanation	n or specify the des	ign smell(s) involve	ed?	
20 (c) If YES (In your or	pinion,) What is the	motivation behind	using this specific v	way of implementat	ion?
20.	5, 11 125, (111 your o	omion, what is the	motivation beinna	using this specific t	vay or implementat	
* 21.	d) Please rate the	severity of the imple	ementation problem	(if any), from 1 (Ve	ry Low) to 5 (Very H	ligh)
	1	2	3	4	5	
	Very Low	l ow	Medium	High	Very High	N/A

e) If YE	ES, would you app	ly this refacto	red solution?			
			SliceParts& key, const Slice	eParts& value) {		
LocalSav	vePoint save(b);					
	= CheckSlicePartsLength((key, value);				
if (!s.ok())) {					
return s;						
}						
O Yes	(Refactor with this so	olution)	Yes	(Refactor with an alte	ernative solution)	
O No ((No refactoring)					
Task:						
		the following	code(s) contain any	occurrence of de	esign smell(impleme	ntation and-or
design static {	n problem)?					
try {						
	oadLibrary("JMPlayer");					
	Throwable err) {					
	tackTrace();					
}						
}						
}				O 11		
-				○ No		
}				○ No		
} • Yes						
} O Yes		an explanatior	n or specify the des		ved?	
} O Yes		an explanatior	n or specify the des		ved?	
} • Yes		an explanatior	n or specify the des		ved?	
} • Yes		an explanatior	n or specify the des		ved?	
Yes	S, please provide a			ign smell(s) involv		
Yes	S, please provide a			ign smell(s) involv	ved? way of implementat	tion?
Yes	S, please provide a			ign smell(s) involv		tion?
Yes	S, please provide a			ign smell(s) involv		tion?
Yes	S, please provide a			ign smell(s) involv		tion?
Yes b) If YES	S, please provide a	,) What is the	motivation behind (ign smell(s) involv	way of implementat	
Yes b) If YES	S, please provide a	,) What is the	motivation behind (ign smell(s) involv		
Yes b) If YES	S, please provide a	,) What is the	motivation behind (ign smell(s) involv	way of implementat	
Yes b) If YES c) If YES	S, please provide a S, (In your opinion ase rate the severi	,) What is the	motivation behind of the second secon	ign smell(s) involvusing this specific	e way of implementat	High)
Yes b) If YES c) If YES	S, please provide a	,) What is the	motivation behind (ign smell(s) involv	way of implementat	
) Yes b) If YES c) If YES	S, please provide a S, (In your opinion ase rate the severi	,) What is the	motivation behind of the second secon	ign smell(s) involvusing this specific	e way of implementat	High)
Yes b) If YES c) If YES	S, please provide a S, (In your opinion ase rate the severi	,) What is the	motivation behind of the second secon	ign smell(s) involvusing this specific	e way of implementat	High)
Yes b) If YES c) If YES	S, please provide a S, (In your opinion ase rate the severi	,) What is the	motivation behind of the second secon	ign smell(s) involvusing this specific	e way of implementat	High)

* 27. e) If YES, would you apply this refactored solution?

public static void loadLibrary

	<pre>static { AccessController.doPrivilege public Void run() { try { System.loadLibrary("JMPlay } catch (Throwable err) { err.printStackTrace(); }} }); }</pre>					
	Yes (Refactor with the No (No refactoring)	his solution)	O Yes	(Refactor with an alter	native solution)	
* 28.	Task: a) In your opinion, of design problem)? JNIEXPORT jint JNICALL Of { OSErr anErr = noErr; anErr = AERemoveEventHax (*env)->DeleteGlobalRef(enr) jvm = 0; ref = 0; mid = 0; return (jint)anErr; } Yes	S_NATIVE(RemoveEventH	dandler) (JNIEnv *env, jobj	ect this)		
29. I	o) If YES, please prov	vide an explanation	or specify the des	sign smell(s) involve	ed?	
30.	c) If YES, (In your opi	nion,) What is the r	motivation behind	using this specific	way of implementati	on?
* 31.	d) Please rate the se	everity of the imple	mentation problem	n (if any), from 1 (Ve	ery Low) to 5 (Very H	ligh)
	1 Very Low	2 Low	3 Medium	4 High	5 Very High	N/A
* 32.	e) If YES, would you JNIEXPORT jint JNICALL O			ect this)		

{

Page 12 of 28

	OSErr anErr = noErr;	and land land and Eq. (2)	LAFO-HIDLAN ASS	Aller ded IDD/AEE	diagnos and Market Communication)
	anErr = AERemoveEventH (*env)->DeleteGlobalRef(el	·	, kAEGetURL, NewAEEven	tHandlerUPP((AEEventHan	idlerProcPtr)NativeCallback), false);
	jvm = 0;	10, 161),				
	ref = 0;					
	mid = 0;					
	if (anErr not null){					
	return (jint)anErr; }}					
	Yes (Refactor with	this solution)	O Yes	(Refactor with an altern	native solution)	
	No (No refactoring))				
* 33.	Task:					
	a) In your opinion, design problem)?	does the following	code(s) contain any	occurrence of des	ign smell(implemer	ntation and-or
	std::vector certBufferRefs(r	numCerts);				
	std::vector certBuffers(num					
	for (size_t i = 0; i < numCer	, -	0.1011.14			
	certBufferRefs[i] = ByteArra		GetObjectArrayElement(end	codedCertificatesJava, i)));		
	if (!certBufferRefs[i]) {	ny tooryptoballer(eriv, cert-	irray.get(), numpti),			
	return;					
	}					
	certBuffers[i] = certBufferRe	efs[i].get();				
	}					
	}					
	○ Yes			O No		
34. I	o) If YES, please pro	vide an explanatior	n or specify the des	ign smell(s) involve	ed?	
35 (c) If YES, (In your op	ninion) What is the	motivation behind a	using this specific v	way of implementati	ion?
00.	5) ii 120, (iii your op	mion, windt is the	motivation benina	using this specific t	ray or implementati	
* 36.	d) Please rate the s	everity of the imple	ementation problem	(if any), from 1 (Ve	ry Low) to 5 (Very H	ligh)
	1 Very Low	2 Low	3 Medium	4 High	5 Very High	N/A

* 37. e) If YES, would you apply this refactored solution?

std::vector
bssl::uniqueptr> certBufferRefs(numCerts);
std::vector<crypto_buffer*> certBuffers(numCerts);

38.	Yes (Refactor with th No (No refactoring) Task:			Refactor with an alte		ation and ar
	design problem)? public NativeLoader(File pare logger.debug("Using native di this.parent = Files.verifyDirect }	nt) throws IOException rectory: %s", parent.Get	{	occurrence of de	sign smell(implement	ation and-or
	○ Yes			O No		
89. k						
	c) If YES, (In your opir	nion,) What is the	motivation behind u	sing this specific	way of implementation	on?
10. c					way of implementation	
10. c						

ask:						
		oes the following	code(s) contain any	occurrence of de	sign smell(implement	ation and-or
esign pro	•					
	,	JNIEnv* e, jbyteArray ap	oplicationProtocolsJava) {			
	nProtocols();					
	rotocolsJava !=	= nullptr) {				
	onProtocols =	uliantiau Duntanala lava				
	rotocols == nu	plicationProtocolsJava,	nulipir),			
earCallbacks		ipii) {				
	.,	etApplicationCallbackSta	ate => applicationProtocols ==	= null" this):		
eturn false;	ppbata 70p ot	a tppnoation can backet	ato application retocolo	, tilo),		
,						
pplicationPro	tocolsLength =	static_cast(e->GetArray	/Length(applicationProtocolsJ	ava));		
		ew char[applicationProto				
nemcpy(appli	cationProtocols	Data, applicationProtoc	ols, applicationProtocolsLeng	th);		
eturn true;						
,						
,						
				O NI-		
○ Yes	ease provi	de an explanatio	on or specify the desi	○ No gn smell(s) involv	ed?	
○ Yes				gn smell(s) involv	ed? way of implementation	on?
Yes If YES, pl	n your opin	nion,) What is the	e motivation behind u	gn smell(s) involv		
Yes If YES, pl	n your opin	nion,) What is the	e motivation behind under the motivation problem	gn smell(s) involvusing this specific (if any), from 1 (Vo	way of implementation	gh)
Yes If YES, pl	n your opin	nion,) What is the	e motivation behind under the motivation problem	gn smell(s) involvusing this specific (if any), from 1 (Vo	way of implementation	gh)
Yes If YES, pl	n your oping	nion,) What is the	e motivation behind under the motivation problem	gn smell(s) involvusing this specific (if any), from 1 (Vo	way of implementation	gh)

jbyte* applicationProtocols =

clearCallbackState();

if (applicationProtocols == nullptr) {

 $e\hbox{-}>\hbox{GetByteArrayElements} (application Protocols Java, nullptr);$

	JNI_TRACE("appData=%p setApplicationCallbackState => applicationProtocols == null", this); return false; }									
48. 50. c	} applicationProtocolsI ength =									
		(applicationProtocols.la	ava)).							
	applicationProtocolsData = new char[applicationProtocolsLength];									
	• •			th);						
	e->ReleaseByteArrayElements	s(applicationProtocolsJa	ava, applicationProtocols, JN	I_ABORT);						
	}									
	return true;									
	}									
	Yes (Refactor with this	s solution)	○ Yes	(Refactor with an alterr	native solution)					
	No (No refactoring)	,		•	,					
* 48.	Task:									
		es the following	code(s) contain any	occurrence of des	ign smell(implement	ation and-or				
	public synchronized void loadLibrary(final String tmpDir) throws IOException {									
	try {									
	System.loadLibrary(sharedLibraryName);									
	System.loadLibrary(jniLibraryN	lame);								
	} catch(final UnsatisfiedLinkEr	emcpy(applicationProtocolsData, applicationProtocols, applicationProtocols, applicationProtocols, JNI_ABORT); >ReleaseByteArrayElements(applicationProtocolsJava, applicationProtocols, JNI_ABORT); turn true; > Yes (Refactor with this solution) No (No refactoring) ask: In your opinion, does the following code(s) contain any occurrence of design smell(implementation and-oresign problem)? abilic synchronized void loadLibrary(final String tmpDir) throws IOException {								
	loadLibraryFromJar(tmpDir);}}]									
	O Voc			O No						
	0 103			0 140						
49. l	b) If YES, please provid	de an explanatio	n or specify the desi	ign smell(s) involve	d?					
50. d	c) If YES, (In your opinion,) What is the motivation behind using this specific way of implementation?									
· E1	d) Places rate the say	varity of the impl	omontation problem	(if any) from 1 (Va	my Lovy) to 5 (Vony His	7h)				
51.	u) Flease rate the sev	enty of the impl		(ii aliy), ii oiii i (ve	ly Low) to 5 (very riit	J''')				
	1	2	3	4	5					
						N/A				
52.	e) If YES, would you a									
	public synchronized void loadL	.ibrary(final String tmpD	ir) throws IOException {							
	try {									
	System.loadLibrary(sharedLibrary)									
	} catch(final UnsatisfiedLinkEri	or ule1) {								

Yes (Refactor with this solution) Yes (Refactor with an alternative solution) No (No refactoring) * 53. Task: a) In your opinion, does the following code(s) contain any occurrence of design smell(implementation and-or design problem)? bool composite_mapping:gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector:iterator ti(components_begin()); it is components_end(x+in){ if ("U*)=gen_struct(name, fields, repoid))} return false; } Yes No No 154. b) If YES, please provide an explanation or specify the design smell(s) involved? *55. c) If YES, (In your opinion,) What is the motivation behind using this specific way of implementation? *56. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High) VeryLow Low Medium High VeryHigh NIA		try { System.loadLibrary(jniLibrary) } catch(final UnsatisfiedLinkE	•				
S3. Task: a) In your opinion, does the following code(s) contain any occurrence of design smell(implementation and-or design problem)? bool composett, empting-gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector:illerator litcomponents_begin(t)); tit = components_empting_it = *th { if (if \text{\tex							
** 53. Task: a) In your opinion, does the following code(s) contain any occurrence of design smell(implementation and-or design problem)? boot composete_resping-gen_struct(UTL_Scopedhame *name, const vector &fields, const char *repoid) { for (vector::iterator it(components_begin(t)); it the components_end(t) +++10 { if (*nth-gen_struct(name, fields, repoid))		Yes (Refactor with the	nis solution)	○ Yes (Refactor with an alter	native solution)	
a) In your opinion, does the following code(s) contain any occurrence of design smell(implementation and-or design problem)? bool composite_mapping_amg_shucit(UTL_ScopedName "name, corest vector! &fields, corest char "repoid) { for (vector:libration (itcomponents_begin(i)): it the components_end():++it) { if (i('t))->gen_struct(name, fields, repoid)) return trate; } return true; } Yes		O No (No refactoring)					
a) In your opinion, does the following code(s) contain any occurrence of design smell(implementation and-or design problem)? bool composite_mapping-gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector:literator ((components_begin())): it != components_sen(): +th) { it ((*(1)->gen_struct(name, fields, repoid))): return failse: } return failse: } **Yes No No No No No No No No No N	. 50	Tools					
design problem? bool composite_mapping:gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector:iterator in(components_begin()); it != components_end(), ++it) { if (if(s)-gen_struct(name, fields, repoid)) return false; } return true; } Yes No 154. b) If YES, please provide an explanation or specify the design smell(s) involved? 155. c) If YES, (In your opinion,) What is the motivation behind using this specific way of implementation? 156. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High) Very Low 2 Medium 4 High Very High N/A 157. e) If YES, would you apply this refactored solution? bool composite_mapping:gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator It(components_begin()); it! te components_end(); ++it) {	53.						
{ for (vector:iterator it(components_begin()); it != components_end(); ++it) { if ((*it)-gen_struct(name, fields, repoid)) return false; } return false; } Yes No No 154. b) If YES, please provide an explanation or specify the design smell(s) involved? 155. c) If YES, (In your opinion,) What is the motivation behind using this specific way of implementation? 156. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High) 157. e) If YES, would you apply this refactored solution? bool composite_mapping:gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::tierator it(components_begin()); it != components_end(); ++it) {		design problem)?				ign smell(implemer	ntation and-or
# 57. e) If YES, would you apply this refactored solution? * 57. e) If YES, would you apply this refactored solution? bool composite, mapping:gen_struct(UTL_ScopedName *rame, const vector &fields, const char *repoid) { for (vector::terator if(components_end(); ++it) { if if(it)-yen_sinut(name, fields, repoid)) return false; } Pyes No No No No 154. b) If YES, please provide an explanation or specify the design smell(s) involved? *55. c) If YES, (In your opinion,) What is the motivation behind using this specific way of implementation? *56. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High) *57. e) If YES, would you apply this refactored solution? bool composite, mapping:gen_struct(UTL_ScopedName *rame, const vector &fields, const char *repoid) { for (vector::terator if(components_begin()); it = components_end(); ++it) {		{			,		
return false; } return true; } Yes							
Yes No			elds, repoid))				
Yes No		}					
54. b) If YES, please provide an explanation or specify the design smell(s) involved? 55. c) If YES, (In your opinion,) What is the motivation behind using this specific way of implementation? 56. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High) 1		return true;					
54. b) If YES, please provide an explanation or specify the design smell(s) involved? 55. c) If YES, (In your opinion,) What is the motivation behind using this specific way of implementation? 56. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High) 1		}					
* 56. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High) Very Low Low Medium High Very High N/A * 57. e) If YES, would you apply this refactored solution? bool composite_mapping::gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator it(componentsbegin()); it != componentsend(); ++it) {		O Yes			O No		
55. c) If YES, (In your opinion,) What is the motivation behind using this specific way of implementation? * 56. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High) Very Low A High Very High N/A * 57. e) If YES, would you apply this refactored solution? bool composite_mapping:gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator it(componentsbegin()): it != componentsend(): ++it) {							
* 56. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High) Very Low 2 Low Medium 4 High Very High N/A * 57. e) If YES, would you apply this refactored solution? bool composite_mapping::gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator it(componentsbegin()); it != componentsend(); ++it) {	54. k	o) If YES, please prov	ide an explanatio	n or specify the desi	gn smell(s) involve	ed?	
* 56. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High) 1							
* 56. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High) 1							
* 56. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High) Very Low Low Medium High Very High N/A * 57. e) If YES, would you apply this refactored solution? bool composite_mapping::gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator it(componentsbegin()); it != components_end(); ++it) {							
Very Low 2 3 4 High Very High N/A * 57. e) If YES, would you apply this refactored solution? bool composite_mapping::gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator it(componentsbegin()); it != componentsend(); ++it) {	55. c	c) If YES, (In your opin	nion,) What is the	motivation behind u	sing this specific	way of implementat	ion?
Very Low 2 N/A * 57. e) If YES, would you apply this refactored solution? bool composite_mapping::gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator it(componentsbegin()); it != componentsend(); ++it) {							
Very Low 2 3 Medium 4 High Very High N/A * 57. e) If YES, would you apply this refactored solution? bool composite_mapping::gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator it(componentsbegin()); it != componentsend(); ++it) {							
Very Low 2 3 4 High Very High N/A * 57. e) If YES, would you apply this refactored solution? bool composite_mapping::gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator it(componentsbegin()); it != components_end(); ++it) {	* EG	d) Diagon water the co	warity of the impl	amantation problem	(if any) from 1 (Va	m, Low) to E (Vom, L	liah)
Very Low Low Medium High Very High N/A * 57. e) If YES, would you apply this refactored solution? bool composite_mapping::gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator it(componentsbegin()); it != componentsend(); ++it) {	30.	d) Please rate the se	verity of the impli	ementation problem	(ii aiiy), iroiii i (ve	y Low) to 5 (very F	ligii)
Very Low Low Medium High Very High N/A * 57. e) If YES, would you apply this refactored solution? bool composite_mapping::gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator it(componentsbegin()); it != componentsend(); ++it) {							
* 57. e) If YES, would you apply this refactored solution? bool composite_mapping::gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator it(componentsbegin()); it != componentsend(); ++it) {		1 Very Low		3 Medium		5 Very High	N/A
* 57. e) If YES, would you apply this refactored solution? bool composite_mapping::gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator it(componentsbegin()); it != componentsend(); ++it) {							
* 57. e) If YES, would you apply this refactored solution? bool composite_mapping::gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator it(componentsbegin()); it != componentsend(); ++it) {							
bool composite_mapping::gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator it(componentsbegin()); it != componentsend(); ++it) {							
bool composite_mapping::gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator it(componentsbegin()); it != componentsend(); ++it) {							
bool composite_mapping::gen_struct(UTL_ScopedName *name, const vector &fields, const char *repoid) { for (vector::iterator it(componentsbegin()); it != componentsend(); ++it) {							
{ for (vector::iterator it(componentsbegin()); it != componentsend(); ++it) {	* 57.				ds, const char *repoid)		
it != componentsend(); ++it) {		{					
		, , ,					
		if (!(*it)->gen_struct(name, fie					
return false; }		return false; }					

	return true; }					
	Yes (Refactor with this No (No refactoring)	solution)	○ Yes (R	efactor with an alter	native solution)	
* 58.	Task:					
	a) In your opinion, do design problem)? int sumNative(JNIEnv* env.jobj { jint anative= (*env)->GetIntFiel jint bnative= (*env)->GetIntFiel jint cnative= (*env)->GetIntFiel return anative + bnative + cnative	ect obj.jobject allVal) eld(env,allVal,a); d(env,allVal,b); d(env,allVal,c);	code(s) contain any o	occurrence of des	ign smell(implemen	tation and-or
	○ Yes			○ No		
59. I	h) If YES, please provid	de an explanation	n or specify the desig	n smell(s) involve	2d?	
	o, 20, piodoo provid		n or opoony and doorg		, , ,	
60.	c) If YES, (In your opini	on,) What is the	motivation behind us	ing this specific	way of implementati	on?
* 61.	d) Please rate the sev	erity of the imple	ementation problem (f anv), from 1 (Ve	rv Low) to 5 (Verv H	iah)
						-97
	1 Very Low	2 Low	3 Medium	4 High	5 Very High	N/A
* 62.						
	Yes (Refactor with this	s solution)	○ Yes (R	efactor with an alter	native solution)	
	No (No refactoring)		(env,allVal,b); env,allVal,b); env,allVal,b); env,allVal,c); } No an explanation or specify the design smell(s) involved? n,) What is the motivation behind using this specific way of implementation? ity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High) 2 A High Very High N/A ply this refactored solution? ct obj, jint a, jint b, jint c){ return a + b + c};			
* 63.	Task:					
	a) In your opinion, do design problem)? int sumNative (JNIEnv* env,job jclass cls=(*env)->GetObjectCl jfieldID a=(*env)->GetFieldID(e	oject obj,jobject allVal){ ass(env,allVal);	code(s) contain any o	occurrence of des	ign smell(implemen	tation and-or

 $jfieldID \; b=(*env)->GetFieldID(env,cls,"b","I");\\$

jint anative=(*env)->GetIntFie jint bnative=(*env)->GetIntFie jint cnative=(*env)->GetIntFie	ld(env,allVal,a); ld(env,allVal,b); ld(env,allVal,c);				
O Yes			O No		
o) If YES, please provi	de an explanatio	n or specify the desi	gn smell(s) involv	ed?	
e) If YES, (In your opin	nion,) What is the	motivation behind u	sing this specific	way of implementatio	on?
d) Please rate the se	verity of the impl	ementation problem	(if any), from 1 (Ve	ery Low) to 5 (Very Hig	gh)
1 Very Low	2 Low	3 Medium	4 High	5 Very High	N/A
int sumNative (JNIEnv* env.jc jint anative=(*env)->GetIntFie jint bnative=(*env)->GetIntFie jint cnative=(*env)->GetIntFie return anative + bnative + cna	bject obj,jobject allVal){ ld(env,allVal,a); ld(env,allVal,b); ld(env,allVal,c); stive;}		Refactor with an alter	rnative solution)	
Task:					
a) In your opinion, do design problem)? public class GodotLib { public static GodotIO io; static { System.loadLibrary("g public static native void initiali	odot_android");}	code(s) contain any	occurrence of des	sign smell(implement	ation and-or
	jint anative=(*env)->GetIntFie jint bnative=(*env)->GetIntFie jint cnative=(*env)->GetIntFie return anative + bnative + cna Yes Task: a) In your opinion, do design problem)? public class GodotLib { public static GodotIO io;	e) If YES, (In your opinion,) What is the dependence of the implementation of the implem	jint anative=("env)->GetIntField(env,allVal,a); jint bnative=("env)->GetIntField(env,allVal,b); jint cnative=("env)->GetIntField(env,allVal,c); return anative + bnative + cnative;} Yes b) If YES, please provide an explanation or specify the desical provides and the severity of the implementation problem Very Low e) If YES, would you apply this refactored solution? int sumNative (JNIEnv* env,jobject obj,jobject allVal){ jint anative=("env)->GetIntField(env,allVal,a); jint bnative=("env)->GetIntField(env,allVal,a); jint cnative=("env)->GetIntField(env,allVal,a); return anative + bnative + cnative;} Yes (Refactor with this solution) No (No refactoring) Task: a) In your opinion, does the following code(s) contain any design problem)? public class GodotItio { public static GodotIto io;	jint anative=("env)>GetintField(env,alIVal.a); jint tonative=("env)>GetintField(env,alIVal.b); jint contive=("env)>GetintField(env,alIVal.b); jint contive=("env)>GetintField(env,alIVal.c); return anative + bnative + cnative) Yes	jint anative*("env)->Geltrif-leid(env,allVal.a); jint bnative*("env)->Geltrif-leid(env,allVal.a); jint bnative*("env)->Geltrif-leid(env,allVal.c); return anative + bnative + cnative.} Ves

	public static native void gravity(float x, float y, float z);	
	public static native void magnetometer(float x, float y, float z);	
	public static native void gyroscope(float x, float y, float z);	
	public static native void key(int p_scancode, int p_unicode_char, boolean p_pressed);	
	public static native void compress(int p_device, int p_but, boolean p_pressed);	
	public static native void decompress(int p_device, int p_axis, float p_value);	
	public static native void securehat(int p_device, int p_hat_x, int p_hat_y);	
	public static native void joyconnectionchanged(int p_device, boolean p_connected, String p_name);	
	public static native void image();	
	public static native void video();	
	public static native void audio();	
	public static native void setmedia(String p_name, Object p_object);	
	public static native void secure(String p_sname, String p_name, String p_ret, String[] p_params);	
	public static native String getGlobal(String p_key);	
	public static native void callobject(int p_ID, String p_method, Object[] p_params);	
	public static native void calldeferred(int p_ID, String p_method, Object[] p_params);	
	public static native void requestPermissionConenctionString p_permission, boolean p_result);	
	public static native void setVirtualKeyboardHeight(int p_height);	
	}	
	○ Yes ○ No	
20	b) If VEO along any ideas and anti-construction and artificial and	
9.	b) If YES, please provide an explanation or specify the design smell(s) involved?	
70.	c) If YES, (In your opinion,) What is the motivation behind using this specific way of implementation?	
71	1. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High)	
, 1.	i. a, ricaso rate the seventy of the implementation problem (if any), from rivery tow, to 5 (very riigh)	

1	2	3	4	5	N/A
Very Low	Low	Medium	High	Very High	

* 72. e) If YES, would you apply this refactored solution?

public class GodotLib {

public static GodotlO io;

static { System.loadLibrary("godot_android");}

public static native void initialize(Godot p_instance, Object p_asset_manager, boolean use_apk_expansion);

public static native void setup(String[] p_cmdline);

public static native void newcontext(boolean p_32_bits);

public static native void resize(int width, int height);

public static native void back();

public static native void step();

public static native void touch(int what, int pointer, int howmany, int[] arr);

public static native void accelerometer(float x, float y, float z);

public static native void gravity(float x, float y, float z);

```
public static native void magnetometer(float x, float y, float z);
public static native void gyroscope(float x, float y, float z);
public static native String getGlobal(String p_key);}
public class media{
public static native void image();
public static native void video();
public static native void audio();
public static native void setmedia(String p_name, Object p_object);}
public class compression{
public static native void compress(int p_device, int p_but, boolean p_pressed);
public static native void decompress(int p_device, int p_axis, float p_value);}
public class connection{
public static native void key(int p_scancode, int p_unicode_char, boolean p_pressed);
public static native void securehat(int p_device, int p_hat_x, int p_hat_y);
public static native void joyconnectionchanged(int p_device, boolean p_connected, String p_name);
public static native void secure(String p_sname, String p_name, String p_ret, String[] p_params);
public static native void callobject(int p_ID, String p_method, Object[] p_params);
public static native void calldeferred(int p_ID, String p_method, Object[] p_params);
public static native void requestPermissionConenctionString p_permission, boolean p_result);
public static native void setVirtualKeyboardHeight(int p_height);}
Yes (Refactor with this solution)

    Yes (Refactor with an alternative solution)

No (No refactoring)
```

* 73. Task:

a) In your opinion, does the following code(s) contain any occurrence of design smell(implementation and-or design problem)?

```
package DDS;
public final class DATAREADER_QOS_DEFAULT {
private DATAREADER_QOS_DEFAULT() {}
public static native DataReaderQos get();
}
package DDS;
public final class DATAREADER_QOS_USE_TOPIC_QOS {
private DATAREADER_QOS_USE_TOPIC_QOS() {}
public static native DataReaderQos get();
package DDS;
public final class DATAWRITER_QOS_DEFAULT {
private DATAWRITER_QOS_DEFAULT() {}
public static native DataWriterQos get();
package DDS;
public final class DATAWRITER_QOS_USE_TOPIC_QOS {
private DATAWRITER_QOS_USE_TOPIC_QOS() {}
public static native DataWriterQos get();
O Yes
                                                                             O No
```

74. b) If YES, please provide an explanation or specify the design smell(s) involved?

75. c) If YES, (In your opinion,) What is the motivation behind using this specific way of implementation?

* 76. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High)



* 77. e) If YES, would you apply this refactored solution?

```
package DDS;
public final class DATAREADER_QOS{
private DATAREADER_QOS() {}
public static native DataReaderQos get();
public static native DataWriterQos get();
}

Yes (Refactor with this solution)

No (No refactoring)

Yes (Refactor with an alternative solution)
```

* 78. Task:

a) In your opinion, does the following code(s) contain any occurrence of design smell(implementation and-or design problem)?

```
public\ static\ String[]\ WinRegGetSubKeys (int\ hKey,\ String\ subKey,\ int\ maxKeyLength)\ \{
byte[] lpSubKey = stringToByteArray(subKey);
int[] openResult = RegOpenKey(hKey, lpSubKey, KEY READ);
if (openResult == null) {
return null;
if (openResult[ERROR_CODE] != ERROR_SUCCESS) {
return null;
} else {
int[] queryResult = RegQueryInfoKey(openResult[OPENED_KEY_HANDLE]);
int subKeysNum = queryResult[SUBKEYS_NUMBER];
if (subKeysNum == 0) {
RegCloseKey(openResult[OPENED_KEY_HANDLE]);
return null;
} else {
String[] keyStrings = new String[subKeysNum];
byte[] keyBytes;
for (int subKeyIndex = 0; subKeyIndex < subKeysNum; subKeyIndex++) {
keyBytes = RegEnumKeyEx(openResult[OPENED_KEY_HANDLE],
subKeyIndex, maxKeyLength);
keyStrings[subKeyIndex] = byteArrayToString(keyBytes);
```

	1		
	RegCloseKey(openResult[OPENED_KEY_HANDLE]);		
	return keyStrings;		
	}		
	}		
	}		
	,		
	private static native byte[] RegEnumKeyEx(int hKey, int subKeyIndex,		
	int maxKeyLength);		
	○ Yes	○ No	
70	b) If YES, please provide an explanation or specify the	ha dasian small(s) involvad?	
19.	b) if 123, please provide all explanation of specify the	ne design sinen(s) involved?	
80.	c) If YES, (In your opinion,) What is the motivation be	ehind using this specific way of implementation?	

* 81. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High)



* 82. e) If YES, would you apply this refactored solution?

```
public\ static\ String[]\ WinRegGetSubKeys(int\ hKey,\ String\ subKey,\ int\ maxKeyLength)\ \{
byte[] lpSubKey = stringToByteArray(subKey);
int[] openResult = RegOpenKey(hKey, IpSubKey, KEY_READ);
if (openResult == null) {
return null;
if (openResult[ERROR_CODE] != ERROR_SUCCESS) {
return null;
} else {
int[] \ queryResult = RegQueryInfoKey(openResult[OPENED\_KEY\_HANDLE]);
int subKeysNum = queryResult[SUBKEYS_NUMBER];
if (subKeysNum == 0) {
RegCloseKey(openResult[OPENED\_KEY\_HANDLE]);
return null;
} else {
String[] keyStrings = new String[subKeysNum];
byte[] keyBytes;
keyBytes = RegEnumKeyEx (openResult[OPENED\_KEY\_HANDLE], subKeysNum, maxKeyLength); \\
RegCloseKey(openResult[OPENED_KEY_HANDLE]);
return keyStrings;
}}}
```

	No (No refactoring)					
* 83.	Task:						
	a) In your opinion, design problem)? final class LibDispatchNation	does the following	code(s) contain any	occurrence of des	ign smell(implemer	ntation and-or	
	static {						
	java.security.AccessContro	oller.doPrivileged(
	(PrivilegedAction) () -> {						
	System.loadLibrary("dispat	tch");					
	return null;						
	});						
	}						
	private LibDispatchNative()) {					
	}	•					
	static native boolean native	elsDispatchSupported():					
		ecuteAsync(long nativeQue	ie Runnahle task)				
	}	outer to yno (long hative daec	ao, ramasio taon,				
	public final class Dispatch	ſ					
	public static Dispatch getIn	istarice() {					
	checkSecurity();	15: (10 (10)					
		reIsDispatchSupported()) ret	turn null;				
	return instance;						
	}						
	NIEXPORT jboolean JNICALL Java_com_apple_concurrent_LibDispatchNative_nativelsDispatchSupported(JNIEnv *env, jclass clazz)						
	{						
	return JNI_TRUE;						
	}						
	Yes			O No			
04) ICVEO		26 41 1		10		
84. I	b) If YES, please pro	ovide an explanation	n or specify the des	ign smell(s) involve	d?		
25	c) If YES, (In your op	ninion) What is the	motivation behind a	using this enecific v	vay of implementati	ion?	
05. (c) ii 123, (iii your op	official, j with at 15 tile	motivation benind t	using this specific v	vay or implementat	1011 :	
* 86.	d) Please rate the s	severity of the imple	ementation problem	(if any), from 1 (Ve	ry Low) to 5 (Very H	ligh)	
	,				, ,		
	1 Very Low	2 Low	3 Medium	4 High	5 Very High	N/A	
	1017 2011	2011	modium	· · · · · · ·	vory riigii	Turk	

Yes (Refactor with an alternative solution)

Yes (Refactor with this solution)

* 87. e) If YES, would you apply this refactored solution?

```
final class LibDispatchNative {
static {
java.security.AccessController.doPrivileged(
(PrivilegedAction) () -> {
System.loadLibrary("dispatch");
return null;
});
}
private LibDispatchNative() {
static native boolean nativeIsDispatchSupported();
public final class Dispatch {
public static Dispatch getInstance() {
checkSecurity();
if (!LibDispatchNative.nativelsDispatchSupported()) return null;
return instance;
JNIEXPORT jboolean JNICALL Java_com_apple_concurrent_LibDispatchNative_nativelsDispatchSupported(JNIEnv *env, jclass clazz)
return JNI_TRUE;
Yes (Refactor with this solution)

    Yes (Refactor with an alternative solution)

No (No refactoring)
```

* 88. Task:

a) In your opinion, does the following code(s) contain any occurrence of design smell(implementation and-or design problem)?

```
static {
java.security.AccessController.doPrivileged(
(PrivilegedAction) () -> {
System.loadLibrary("dispatch");
return null;
});
}
private LibDispatchNative() {
static native boolean nativeIsDispatchSupported();
static native long nativeCreateConcurrentQueue(int priority);
public final class Dispatch {
public static Dispatch getInstance() {
checkSecurity();
if (!LibDispatchNative.nativeIsDispatchSupported()) return null;
return instance;
JNIEXPORT jboolean JNICALL Java_com_apple_concurrent_LibDispatchNative_nativelsDispatchSupported (JNIEnv *env, jclass clazz)
return JNI_TRUE;
}
JNIEXPORT jlong JNICALL Java com apple concurrent LibDispatchNative nativeCreateConcurrentQueue (JNIEnv *env, jclass clazz, jint priority)
dispatch_queue_t queue = dispatch_get_global_queue((long)priority, 0);
return pt
Yes
                                                                                       O No
```

89.	b) If YES, please provide an explanation or specify the design smell(s) involved?
90.	c) If YES, (In your opinion,) What is the motivation behind using this specific way of implementation?
* 91	. d) Please rate the severity of the implementation problem (if any), from 1 (Very Low) to 5 (Very High)

1	2	3	4	5	N/A
Very Low	Low	Medium	High	Very High	

* 92. e) If YES, would you apply this refactored solution?

```
static {
java.security.AccessController.doPrivileged(
(PrivilegedAction) () -> {
System.loadLibrary("dispatch");
return null;
});
}
private LibDispatchNative() {
static native boolean nativeIsDispatchSupported();
public final class Dispatch {
public static Dispatch getInstance() {
checkSecurity();
if (!LibDispatchNative.nativeIsDispatchSupported()) return null;
return instance;
{\tt JNIEXPORT\ jboolean\ JNICALL\ Java\_com\_apple\_concurrent\_LibDispatchNative\_nativelsDispatchSupported\ (JNIEnv\ ^env,\ jclass\ clazz)}
return JNI_TRUE;
Yes (Refactor with this solution)
                                                                       Yes (Refactor with an alternative solution)
No (No refactoring)
```

* 93. Task:

a) In your opinion, does the following code(s) contain any occurrence of design smell(implementation and-or design problem)?

```
public static void setSearchField(JTextField txt, boolean isSearchField) {
if (isSearchField == isSearchField(txt)) {
```

txt.putClientProperty(MAC_TEXT_FIELD_VARIANT_PROPERTY, "_triggerevent_"); } else if (isSearchField) { uiChangeHandler.install(txt); } else { uiChangeHandler.uninstall(txt); }									
txt.putClientProperty("Quaqu } else { txt.putClientProperty(MAC_1	txt.putClientProperty(MAC_TEXT_FIELD_VARIANT_PROPERTY, MAC_SEARCH_VARIANT); txt.putClientProperty("Quaqua.TextField.style", MAC_SEARCH_VARIANT);								
}}									
Yes			O No						
b) If VEC places pro-	iido an avalenetie:	n or anasifu the das:	an amell(a) involve	nd?					
b) If YES, please prov	ide an explanation	n or specify the desi	gn smell(s) involve	ed ?					
a) If VEC /In your ani	nion \ \Albatia tha	mativation behind .	roing this appoiling		-m2				
c) If YES, (In your opi	nion,) what is the	motivation bening u	ising this specific	way of implementation	on ?				
6. d) Please rate the se	everity of the imple	ementation problem 3 Medium	(if any), from 1 (Ve	ery Low) to 5 (Very Hi	gh) N/A				
7. e) If YES, would you public static void setSearchF if (isSearchField == isSearch txt.putClientProperty(MAC_T } else if (isSearchField) { uiChangeHandler.install(txt); } else { uiChangeHandler.uninstall(txt); } txt.putClientProperty(MAC_T txt.putClientProperty("Quaque")}	rield(JTextField txt, boolean Field(txt)) { EXT_FIELD_VARIANT_F ext); EXT_FIELD_VARIANT_F ia.TextField.style", "defaul	an isSearchField) { PROPERTY, "_triggerevent_" PROPERTY, "default"); It");							
 Yes (Refactor with the 	nis solution)	Voc.							
	,	U les (Refactor with an alter	native solution)					

Your responses have been registered!

Thank you for taking the time to complete the survey, your input is valuable to us.