1. Kurze Beschreibung der einzelnen Klassen

- **BackendMIPS:**
  In BackendMIPS sind die Methoden laut Angabe CA1 implementiert.
- **GlobalPointerRegister:**
  Hier werden Bytes in der „Static Data Area“ allociert. Für eine passende Wortlänge befindet sich hier die Methode doWordAlignment.
- **Register:**
  So wird ein Register mit Name und Registernummer gespeichert. Abfrage und Setzen ob das Register in Verwendung ist.
- **Registers:**
  Hier befindet sich das Registermanagement, wie T und S Register anfordern und freigeben. Abrufen der restlichen Register, wie z.B: GlobalPointerRegister, FramePointerRegister, etc.
- **Segment:**
  Abfrage von Data- und Textsegment „String“.
- **StackPointerRegister:**
  Hier werden Bytes auf dem Stack allociert, freigegeben und WordAlignment kann vorgenommen werden. Offsets werden Prozeduren-übergreifend verwaltet.
- **SyscallCode:**
  Hier befinden sich Syscall Enums. Diese können nun auch mithilfe des Namens und nicht nur mit der Zahl im Backend verwendet werden.

2. Code Coverage

- Code Coverage wurde mit Hilfe von IntelliJ IDEA vorgenommen.

Klassen/Testfälle	1	2	3	4	5	6	Error	IfElse	Types
BackendMIPS									
Methodenabdeckung	49%	50%	59%	61%	70%	63%	3%	47%	64%
Zeilenabdeckung	42%	42%	51%	57%	66%	55%	2%	39%	57%
GlobalPointerRegister									
Methodenabdeckung	100%	33%	100%	100%	100%	100%	33%	33%	33%
Zeilenabdeckung	75%	23%	76%	70%	70%	70%	23%	23%	23%
Register									
Methodenabdeckung	100%	100%	100%	100%	100%	100%	33%	100%	100%
Zeilenabdeckung	100%	100%	100%	100%	100%	100%	58%	100%	100%
Registers									
Methodenabdeckung	64%	58%	64%	64%	76%	64%	5%	58%	64%
Zeilenabdeckung	63%	63%	64%	67%	71%	67%	35%	64%	67%
Segment									
Methodenabdeckung	100%	100%	100%	100%	100%	100%	0%	100%	100%
Zeilenabdeckung	100%	100%	100%	100%	100%	100%	0%	100%	100%
StackPointerRegister									
Methodenabdeckung	83%	100%	100%	100%	100%	100%	16%	83%	83%
Zeilenabdeckung	70%	91%	91%	91%	91%	91%	12%	70%	70%
SyscallCode									
Methodenabdeckung	100%	100%	100%	100%	100%	100%	0%	100%	100%
Zeilenabdeckung	100%	100%	100%	100%	100%	100%	0%	100%	100%

3. Code Coverage - Detailliert

❖ Von uns wurden zusätzlich zu den Testfällen 1-6 die Testfälle „Test_Error“, „Test_IfElse“, „Test_Types“ hinzugefügt, durch welche die Methodenabdeckung, als auch die Zeilenabdeckung der Klasse **BackendMIPS** zu 100% gegeben ist.

- **Test_Error:** Mit dem Testfall „Test_Error“ wurde der noch nicht abgedeckte Teil der Methode „BackendMIPS“ abgedeckt. Mit dem Testfall wird die If-Anweisung durch outputStream == 0 erfüllt und es wird eine Exception geworfen.

- **Test_Types**: Mit diesem Testfall wurden die noch nicht abgedeckten Methoden „arrayLength“, „neg“, „div“, „mod“, „isLess“, „isLessOrEqual“, „isEqual“, „not“, „and“, „or“ und „branchIf“ abgedeckt.

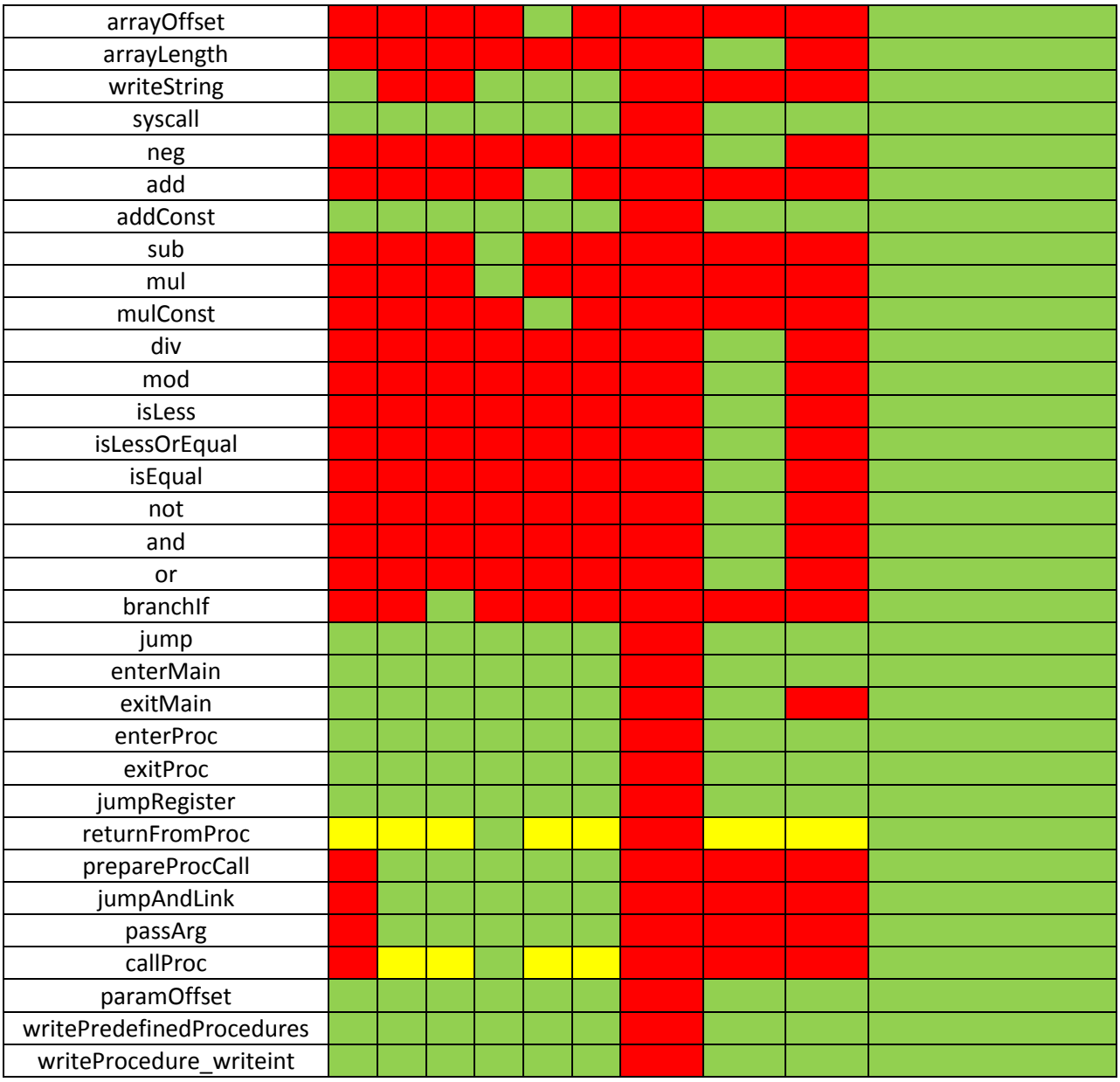
- **Test_IfElse:** Mit dem Testfall wurde in der Methode „storeArrayDim“ die If-Anweisung ausgeführt, sowie in der Methode „loadAdress“ der Else-Zweig benutzt.

❖ Ob eine ganze Methode, nur zeilenweise Abdeckung oder gar keine Abdeckung der Methode vorliegt, wird in der untenstehenden Tabelle dargestellt.

Agenda:

- Grün – Vollständige Abdeckung der gesamten Methode
- Gelb – Zeilenweise Abdeckung der Methode
- Rot – Keine Abdeckung der Methode

Testfälle	1	2	3	4	5	6	Error	Types	IfElse	In Summe
BackendMIPS										
BackendMIPS										
changeSegment										
wordSize										
boolValue										
allocReg										
freeReg										
zeroReg										
comment										
emitLabel										
allocStaticData										
allocStringConstant										
allocStack										
allocHeap										
move										
storeArrayDim										
allocArray										
loadConst										
loadAdress										
loadWord										
storeWord										
loadWordReg										
loadWordReg										
storeWordReg										


arrayOffset
arrayLength
writeString
syscall
neg
add
addConst
sub
mul
mulConst
div
mod
isLess
isLessOrEqual
isEqual
not
and
or
branchIf
jump
enterMain
exitMain
enterProc
exitProc
jumpRegister
returnFromProc
prepareProcCall
jumpAndLink
passArg
callProc
paramOffset
writePredefinedProcedures
writeProcedure_writeint