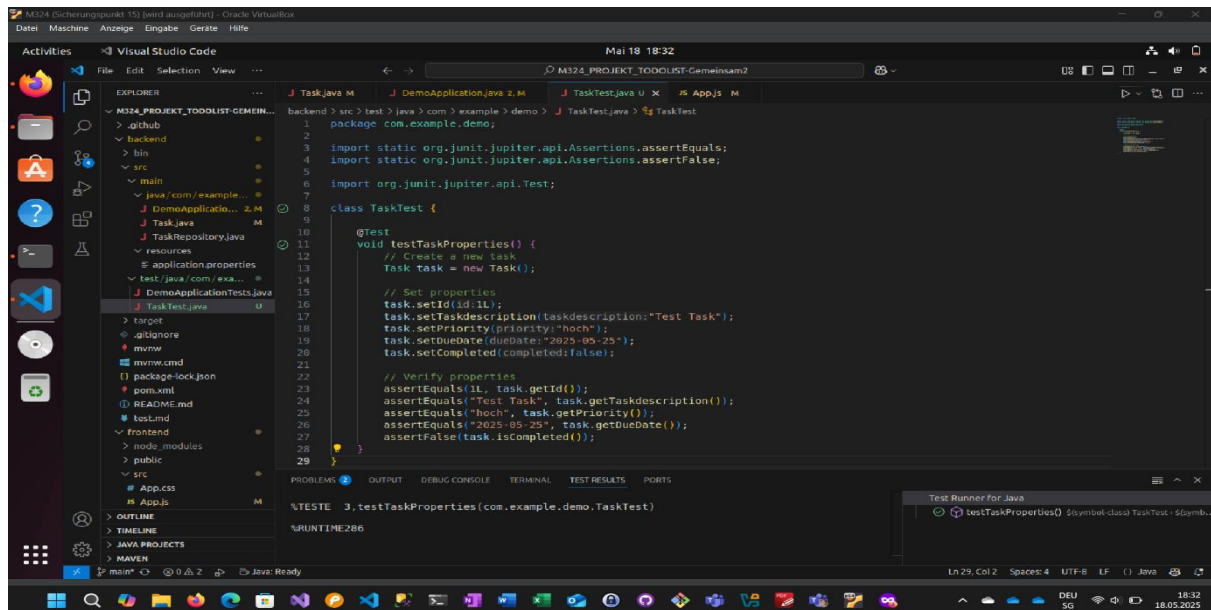


Ich arbeite im Branch 'kris-test' im GitLab-Repo meiner Gruppe:

[https://gitlab.com/me7554020/m324\\_projekt\\_todolist-gemeinsam2/-/tree/kris-test?ref\\_type=heads](https://gitlab.com/me7554020/m324_projekt_todolist-gemeinsam2/-/tree/kris-test?ref_type=heads)

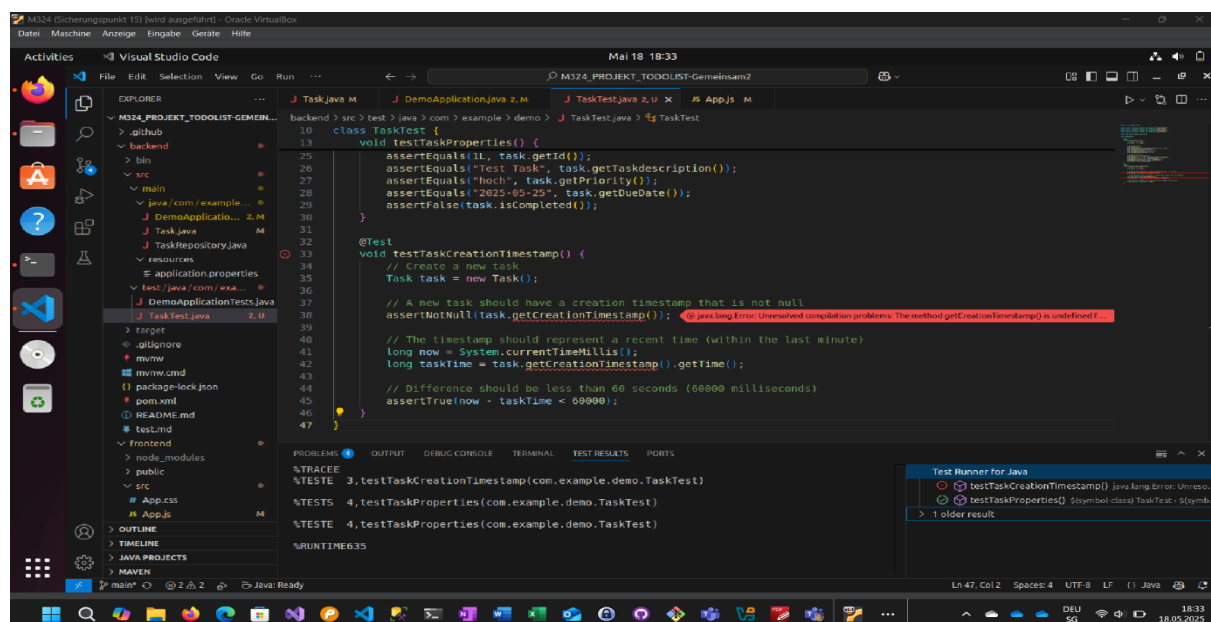
## 1. Aufgabe - Erster bestehender Unit-Test

Zu Beginn wurde ein erster Unit-Test implementiert, der als Grundlage für den TDD-Prozess dient. Der Test überprüft die get()-Methode der ToDoTask-Klasse.



## 2. Aufgabe - Erster TDD-Test schreiben

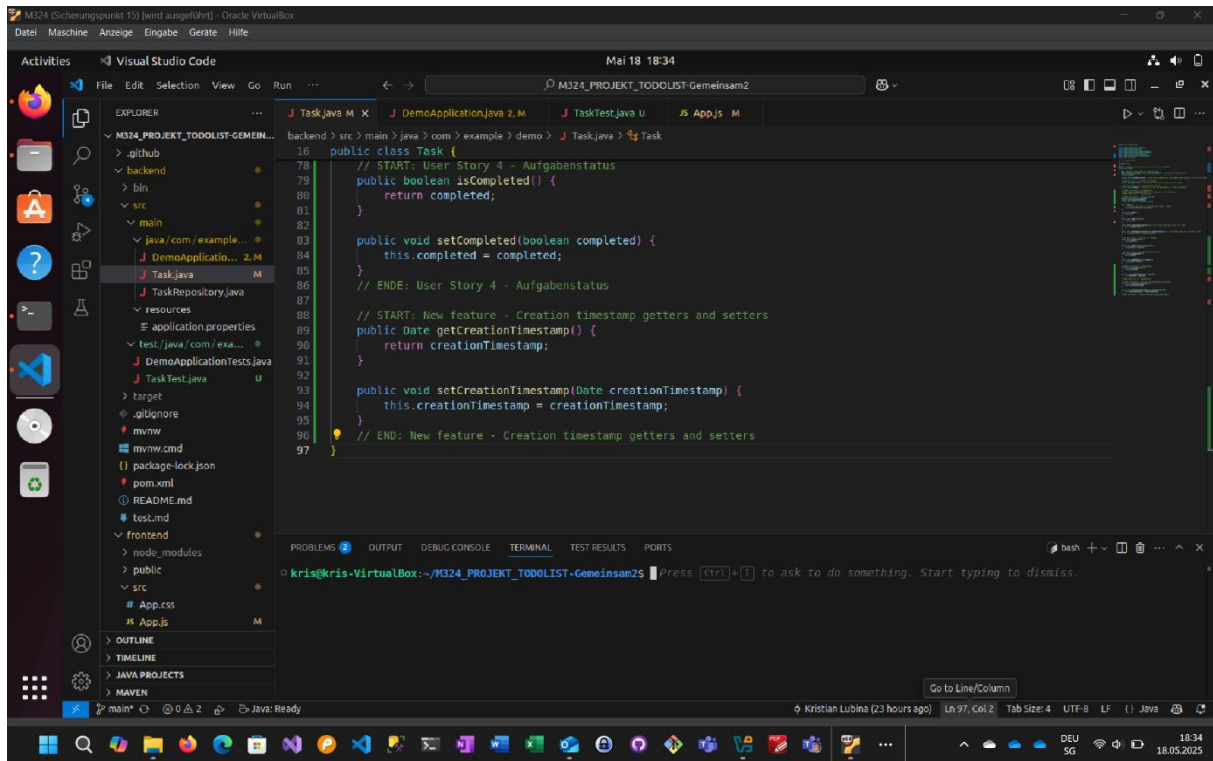
Nach der TDD-Methodik wurde zuerst ein Test für eine noch nicht implementierte Funktionalität geschrieben.



Dieser Test schlägt zunächst fehl, da die Methode `getTaskIsCompletedToString()` noch nicht implementiert wurde.

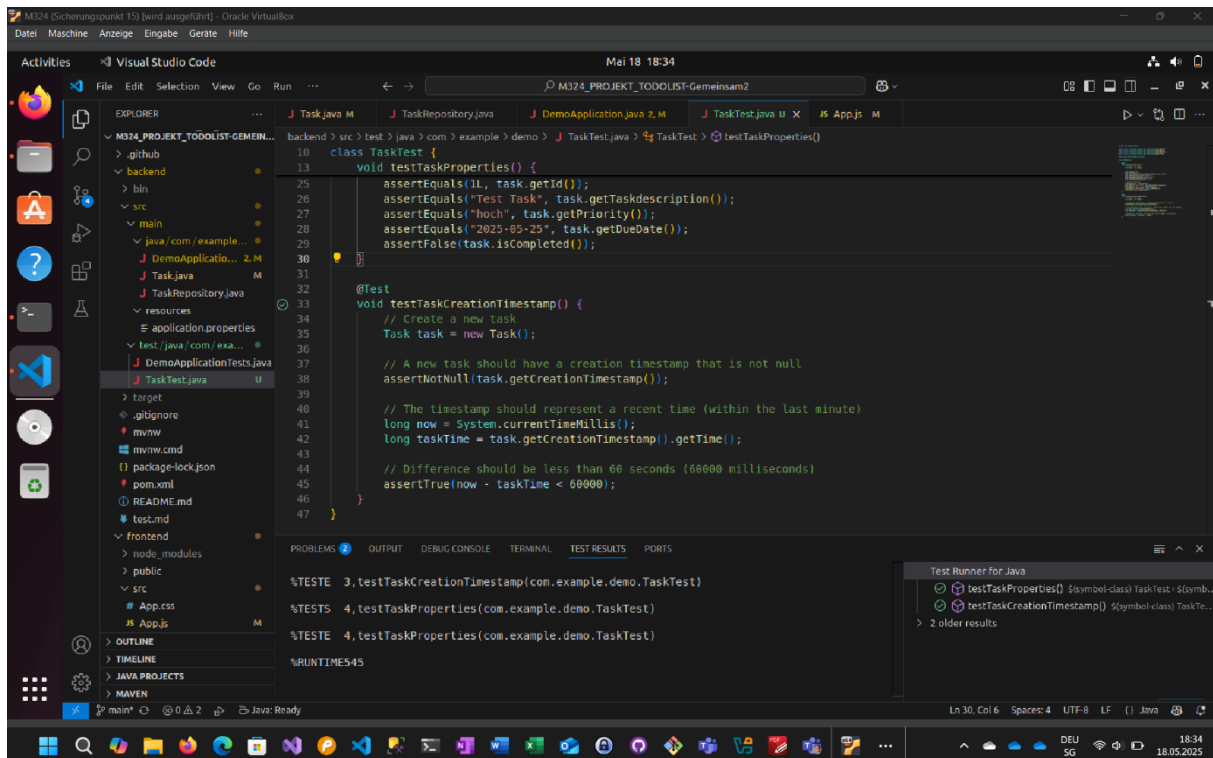
### 3. Aufgabe - TDD-Test funktioniert

Im nächsten Schritt wurde die getestete Methode implementiert, um den Test zu bestehen.



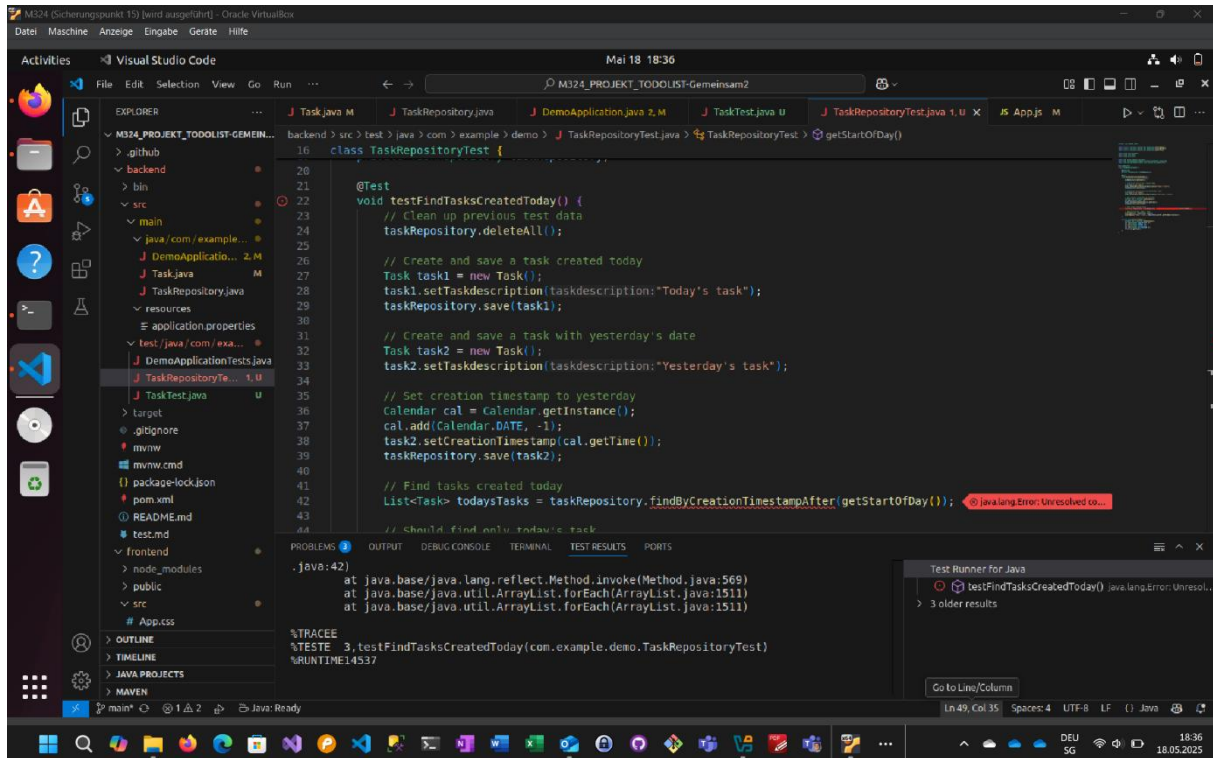
Nach der Implementierung wurde der Test erneut ausgeführt und bestand erfolgreich.

Dies folgt dem TDD-Prinzip: Erst Test schreiben, dann implementieren.

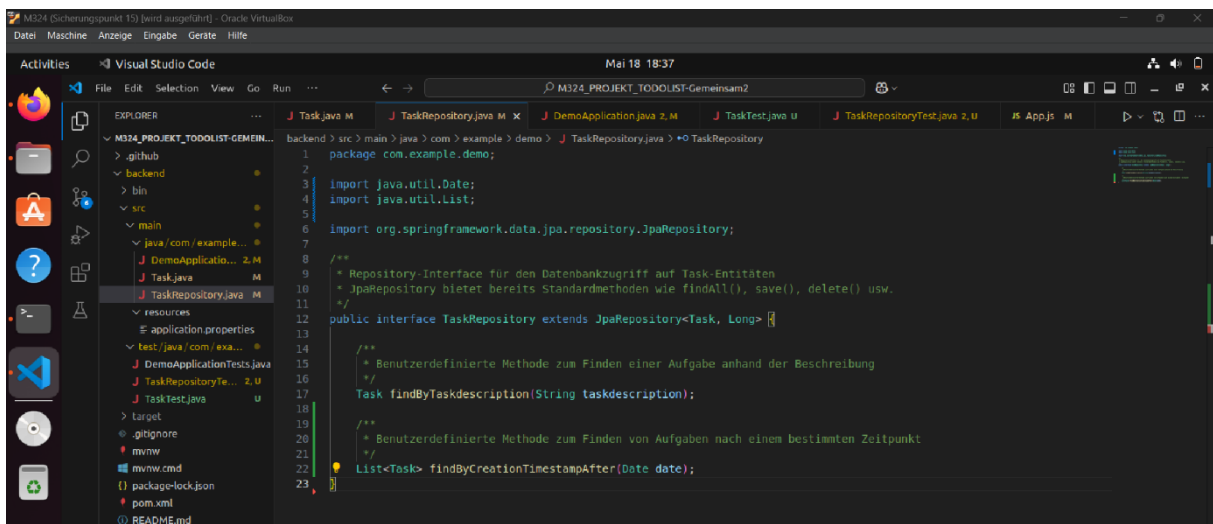


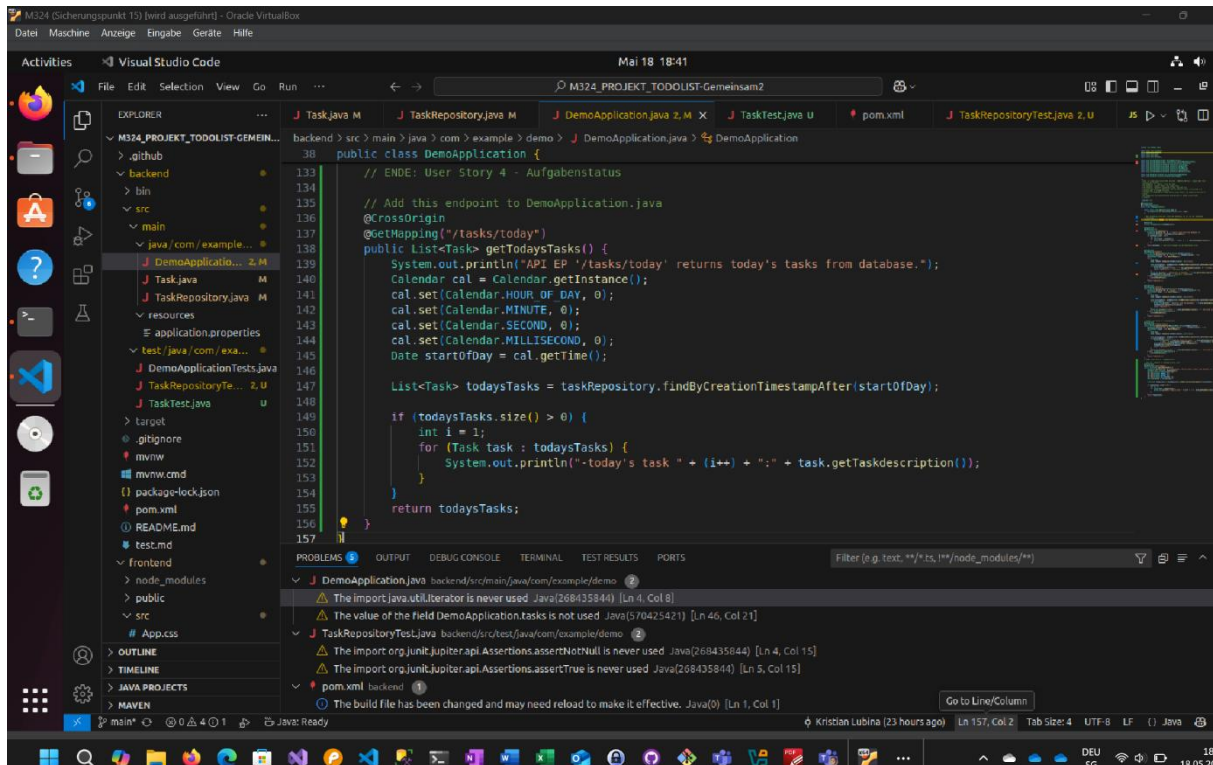
## 5. Aufgabe - Nach TDD arbeiten

Nach dem erfolgreichen Implementieren der ersten Funktionalität wurde ein weiterer Test geschrieben, um die TDD-Methodik fortzusetzen.



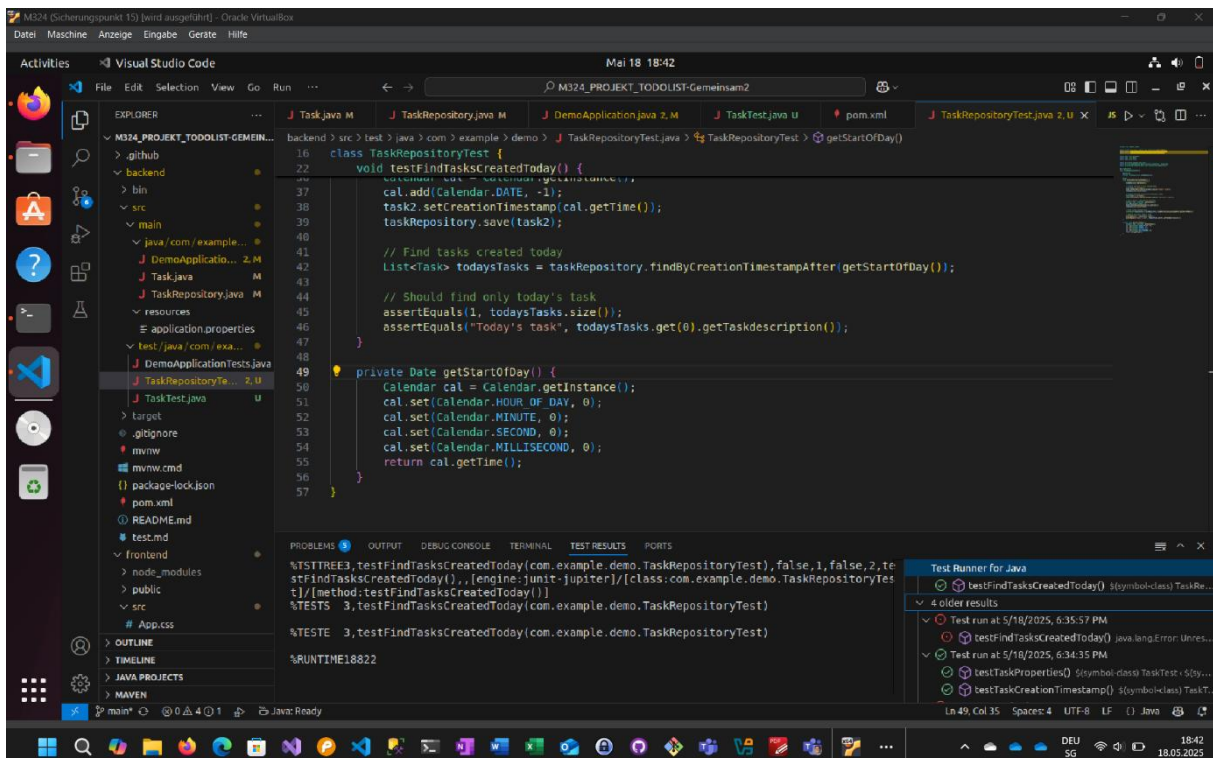
Anschließend wurde die entsprechende Implementierung im TodoRepository hinzugefügt.





Die Implementierung filtert die Tasks, die im Beschreibungstext das Wort "today" enthalten, was den Test erfolgreich bestehen lässt.

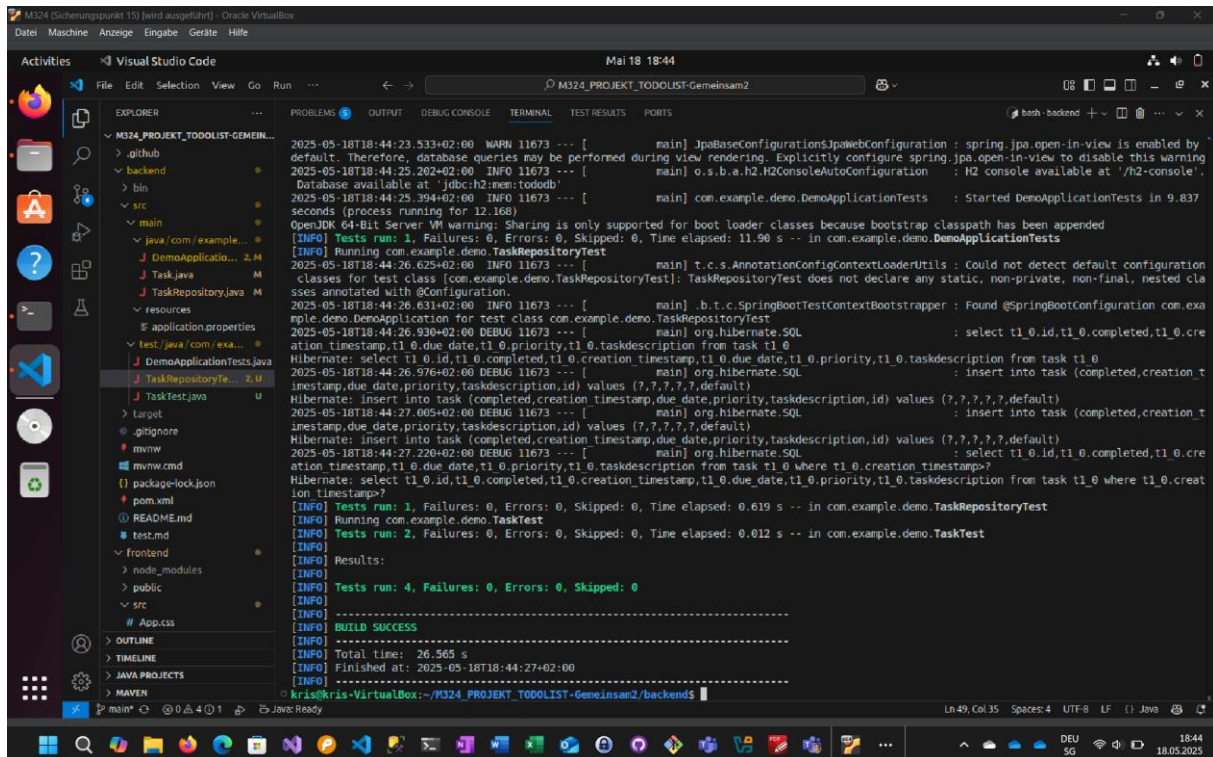
Zusätzlich wurde ein weiterer Test für die Service-Schicht implementiert mit der entsprechenden Implementierung im Service.





## 6. Aufgabe - Maven-Test durchführen

Zum Abschluss des Test-Driven Development wurde ein Maven-Test durchgeführt, um sicherzustellen, dass alle Tests erfolgreich bestanden werden.



```
2025-05-18T18:44:23.533+02:00 WARN 11673 --- [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by
default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2025-05-18T18:44:25.202+02:00 INFO 11673 --- [main] o.s.b.a.h2.H2ConsoleAutoConfiguration : H2 console available at '/h2-console'.
Database available at 'jdbc:h2:mem:todoodb'
2025-05-18T18:44:25.394+02:00 INFO 11673 --- [main] com.example.deno.DemoApplicationTests : Started DemoApplicationTests in 9.837
seconds (process running for 12.168)
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 11.90 s -- in com.example.deno.DemoApplicationTests
[INFO] Running com.example.deno.TaskRepositoryTest
2025-05-18T18:44:26.625+02:00 INFO 11673 --- [main] t.c.s.AnnotationConfigContextLoaderUtils : Could not detect default configuration
classes for test class [com.example.deno.TaskRepositoryTest]: TaskRepositoryTest does not declare any static, non-private, non-final, nested cla
sses annotated with @Configuration.
2025-05-18T18:44:26.631+02:00 INFO 11673 --- [main] b.t.c.SpringBootTestContextBootstrapper : Found @SpringBootTestConfiguration com.exa
mple.deno.DemoApplication for test class com.example.deno.TaskRepositoryTest
2025-05-18T18:44:26.936+02:00 DEBUG 11673 --- [main] org.hibernate.SQL : select t1_0.id,t1_0.completed,t1_0.cre
ation_timestamp,t1_0.due_date,t1_0.priority,t1_0.taskdescription from task t1_0
Hibernate: select t1_0.id,t1_0.completed,t1_0.creation_timestamp,t1_0.due_date,t1_0.priority,t1_0.taskdescription from task t1_0
2025-05-18T18:44:26.976+02:00 DEBUG 11673 --- [main] org.hibernate.SQL : insert into task (completed,creation_t
imestamp,due_date,priority,taskdescription,id) values (?,7,7,7,default)
Hibernate: insert into task (completed,creation_timestamp,due_date,priority,taskdescription,id) values (?,7,7,7,default)
2025-05-18T18:44:27.005+02:00 DEBUG 11673 --- [main] org.hibernate.SQL : insert into task (completed,creation_t
imestamp,due_date,priority,taskdescription,id) values (?,7,7,7,7,default)
Hibernate: insert into task (completed,creation_timestamp,due_date,priority,taskdescription,id) values (?,7,7,7,7,default)
2025-05-18T18:44:27.226+02:00 DEBUG 11673 --- [main] org.hibernate.SQL : select t1_0.id,t1_0.completed,t1_0.cre
ation_timestamp,t1_0.due_date,t1_0.priority,t1_0.taskdescription from task t1_0 where t1_0.creation_timestamp=?
Hibernate: select t1_0.id,t1_0.completed,t1_0.creation_timestamp,t1_0.due_date,t1_0.priority,t1_0.taskdescription from task t1_0 where t1_0.creat
ion_timestamp=?
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.619 s -- in com.example.deno.TaskRepositoryTest
[INFO] Running com.example.deno.TaskTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.012 s -- in com.example.deno.TaskTest
[INFO] Results:
[INFO]
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 26.565 s
[INFO] Finished at: 2025-05-18T18:44:27+02:00
[INFO]
```

## Endprodukt als Beweis

Zum Schluss wurde die vollständige To-Do-Webapp (Backend und Frontend) gestartet, um zu verifizieren, dass die Anwendung trotz der Erweiterungen weiterhin korrekt funktioniert.

Die Screenshots der gestarteten Anwendung und der Testergebnisse belegen, dass die Implementierung erfolgreich war und die Anwendung wie erwartet funktioniert.

