

Всі основні функції проекту реалізовані:

Код доступний за посиланням: [https://github.com/NeNaO4ke/ATM\\_MOOP](https://github.com/NeNaO4ke/ATM_MOOP)

Діаграми додаю у архів з кодом.

#### Результати тестів

✓ <default package>	1 sec 97 ms
✓ AccountTest	392 ms
✓ getTransactional()	392 ms
✓ BankTest	27 ms
✓ bankTest()	17 ms
✓ contextLoads()	10 ms
✓ CardTest	154 ms
✓ verifyAtmSupportBank()	121 ms
✓ updatePin()	33 ms
✓ LoginTest	220 ms
✓ invalidLoginWithCardBlock()	158 ms
✓ loginAvailableForAll()	7 ms
✓ getAuth()	21 ms
✓ login()	15 ms
✓ invalidLogin()	19 ms
✓ RegularTransactionTest	137 ms
✓ scheduleTransaction()	93 ms
✓ regularTransaction()	44 ms
✓ TransferTransactionTest	167 ms
✓ withdraw()	55 ms
✓ transferFromTransactionalToTransactional()	38 ms
✓ deposit()	29 ms
✓ historyOfAccount()	45 ms

Тести виконуються у спеціально створеному для них контексті, тобто для них використовується інша база даних(H2) яка при цьому функціонує у пам'яті і також після кожного тесту відкочує стан до початкового, що значно полегшує тестування.

Були додані транзакції відкладені за часом та регулярні транзакції з відкладеним стартом або ні.

Функціонал накопичувального рахунку можливо буде реалізовано до дедлайну Тфін, тим паче що модель даних уже визначена й залишилось лише реалізувати регулярну обробку великих даних за допомогою одного із способів: Spring Batch + Quartz, Apache Kafka + FLink, Apache Spark