Scuola universitaria professionale della Svizzera italiana
Dipartimento tecnologie innovative

SUPSI

File System Command Interpreter

Studente/i

Kristian Boldini
Maria Grazia Corino
Ahmed El Aidy

Docente

Prof. Amos Brocco
Prof. Giancarlo Corti

Corso di laurea

Informatica

Modulo

Software Engineering and Development II

Anno

2023/24

Data

18 dicembre 2023

STUDENTSUPSI

Sommario

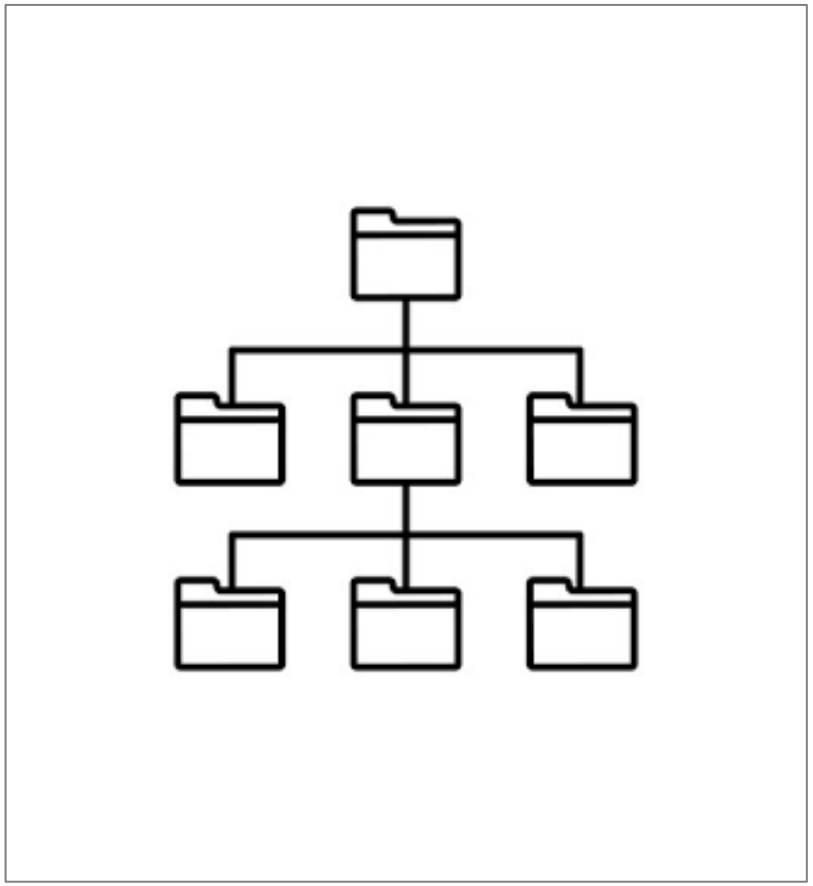
- Contesto e motivazioni
- Problema
- Requisiti
- Tecnologie di sviluppo
- Metodologia di lavoro
- Stato dell'arte
- Approccio
- Engine - Filesystem
- Client – Command Creation
- Client – Command Execution
- Risultati
- Conclusioni

Contesto e motivazioni

- Progetto nell'ambito dell'ingegneria del software
- Requisito per la certificazione del corso di Software Engineering and Development II

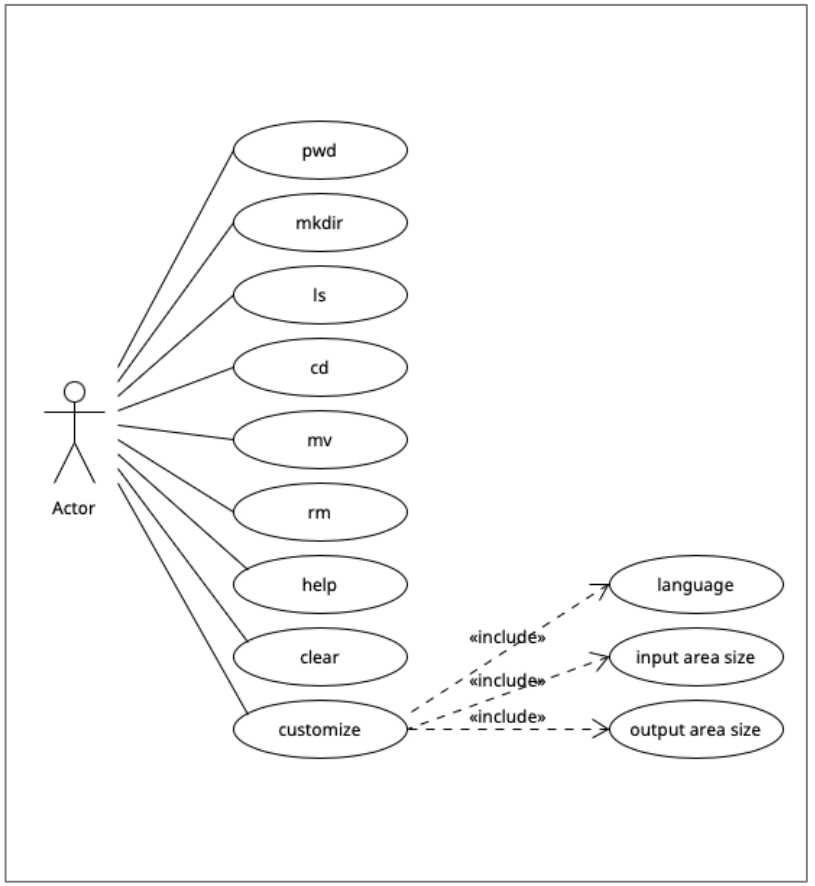
Problema

- Fornire un’applicazione standalone per un interprete di comandi che simuli un filesystem



Requisiti

- Manipolazione di un filesystem
  - pwd : stampare percorso corrente
  - mkdir: creare nuova directory
  - ls: listare contenuto directory
  - cd: cambiare directory corrente
  - mv: spostare directory
  - rm: eliminare directory
  - help: elencare comandi disponibili
  - clear: pulire area di output
- Personalizzazione
  - Lingua
  - Numero di colonne per l'inserimento
  - Numero di righe dell'area di output



Tecnologie di sviluppo

- Applicativo
  - Java
- Graphic User Interface (GUI)
  - JavaFX
- Versioning
  - GIT
- Dependencies management e build
  - Maven
- Testing
  - Unit testing: Junit e Mockito
  - End-to-end: TestFX

Metodologia di lavoro

- Agile con Scrum
- 1 settimana per Sprint

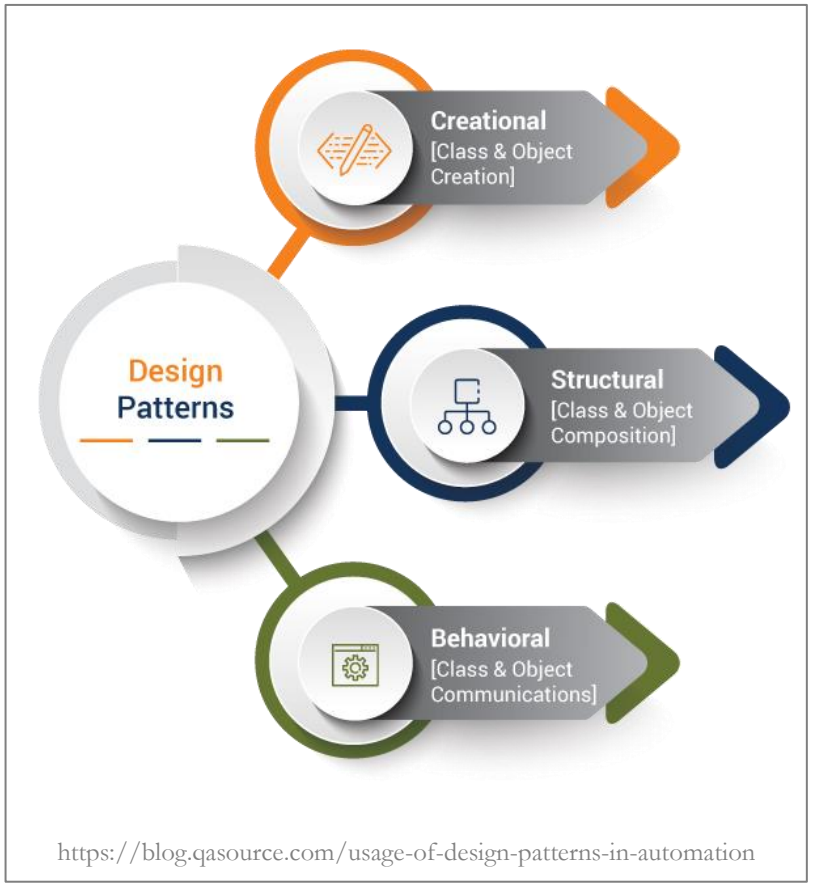
Stato dell’arte

- Esistono soluzioni simili già sviluppate
- Lavoro didattico di sviluppo di un’applicazione
  - Metodologia agile
  - Design patterns
  - Testing

Approccio

- Pattern usati
  - MVC
  - Observer
  - Visitor
  - Command
  - Interpreter
  - Factory Method
  - Singleton

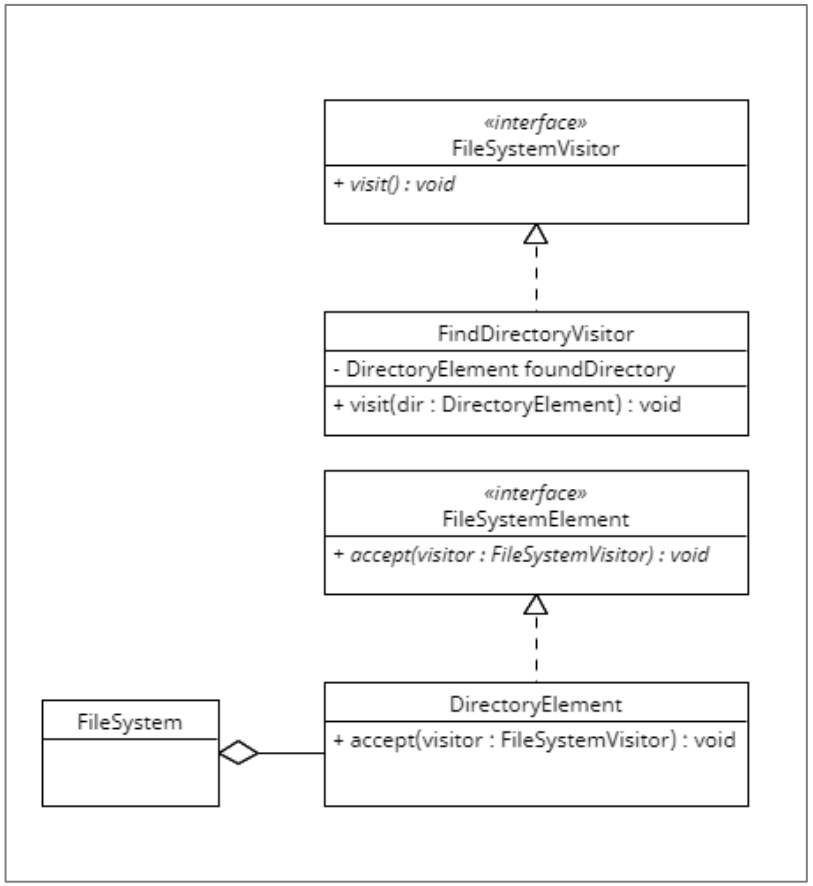
- Struttura
  - Client
  - Engine



https://blog.qasource.com/usage-of-design-patterns-in-automation

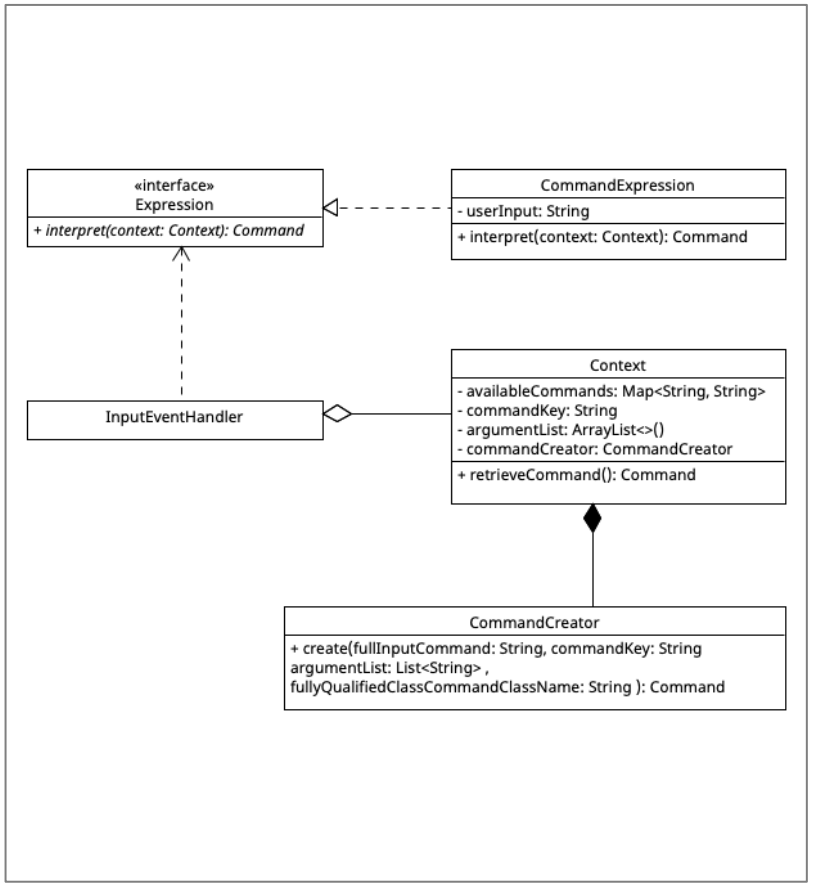
Engine - Filesystem

- Struttura ad albero
- Manipolazione elementi
- Visitor pattern
- Singleton pattern



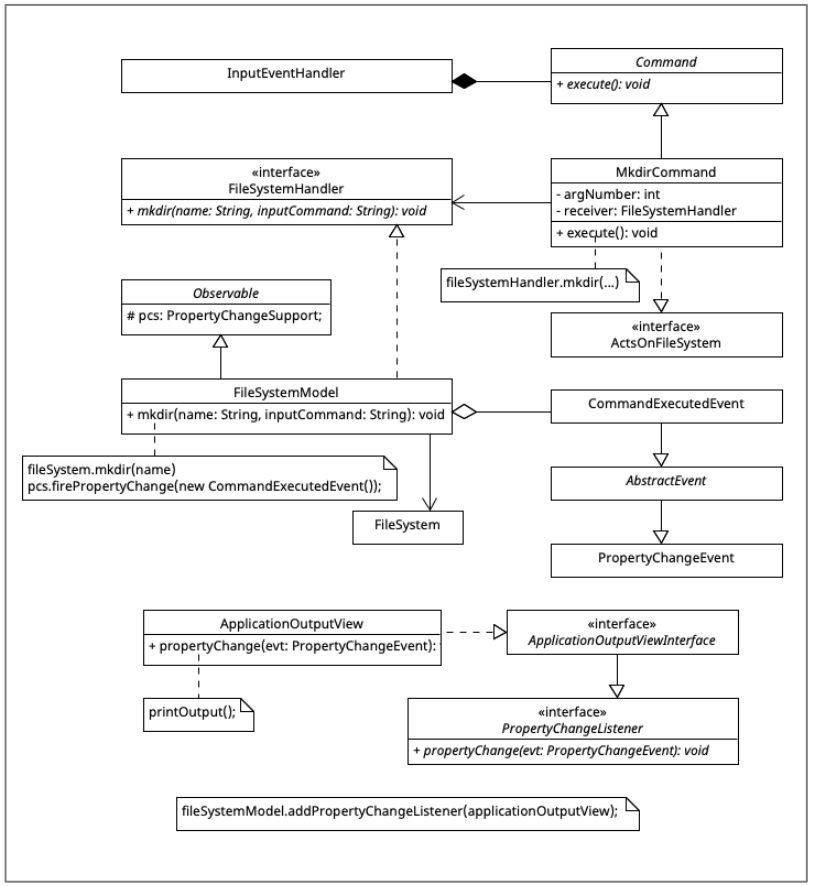
Client – Command Creation

- InputEventHandler
  - Riceve input utente
  - Chiede il comando adeguato
  - Lancia esecuzione comando
- Interpreter pattern
  - Controlla validità comando
  - Estrae argomenti
  - Chiede creazione comando
- Factory Method
  - Controlla validità di argomenti
  - Crea comando adeguato



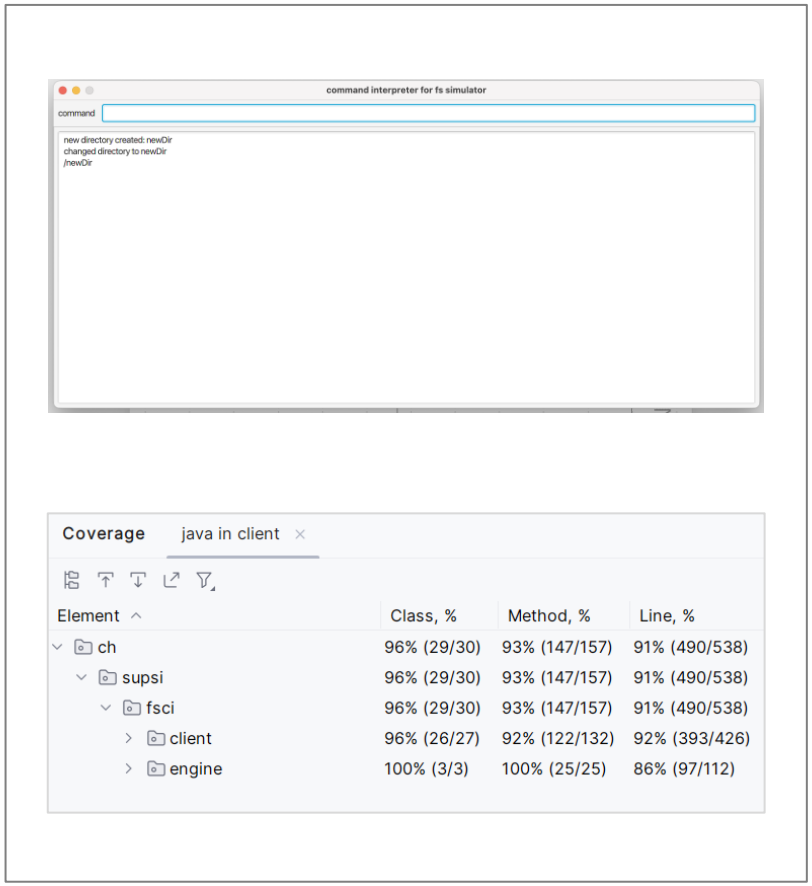
Client – Command Execution

- Command pattern
  - Esegue comando appropriato
  - Invoker: InputEventHandler
- Observer pattern:
  - Per mezzo di classi Java
    - PropertyChangeListener
    - PropertyChangeSupport
    - PropertyChangeEvent



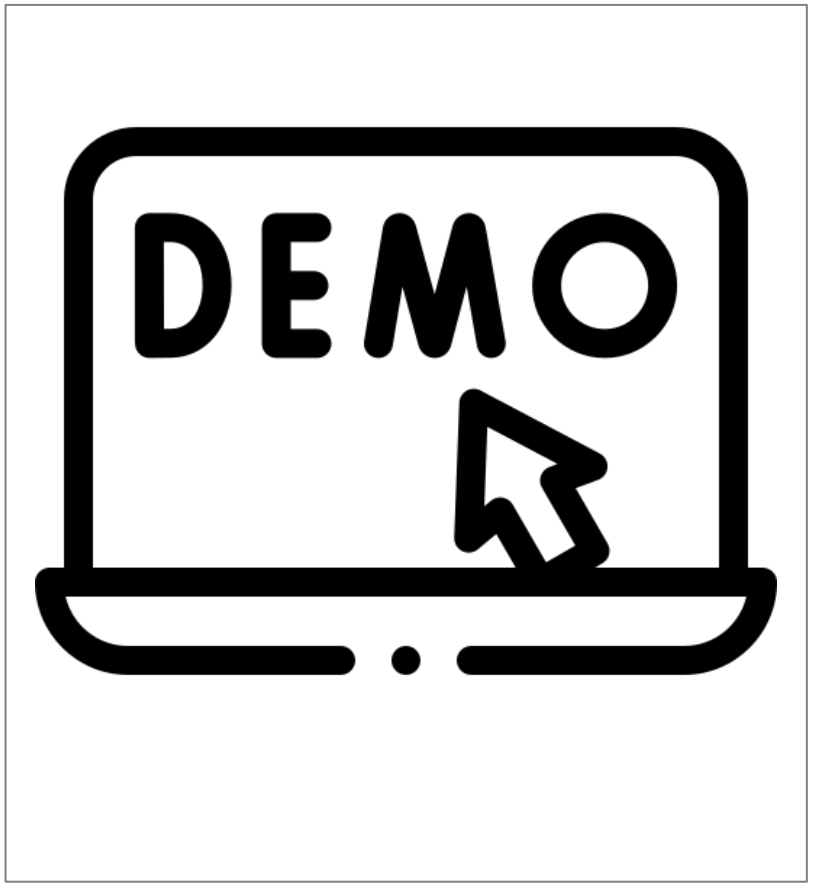
Risultati

- Applicazione che soddisfa i requisiti
  - Funzionali
  - Non funzionali
- Test
  - Eseguiti con JUnit per i componenti
  - Eseguiti con TestFX per l'interfaccia



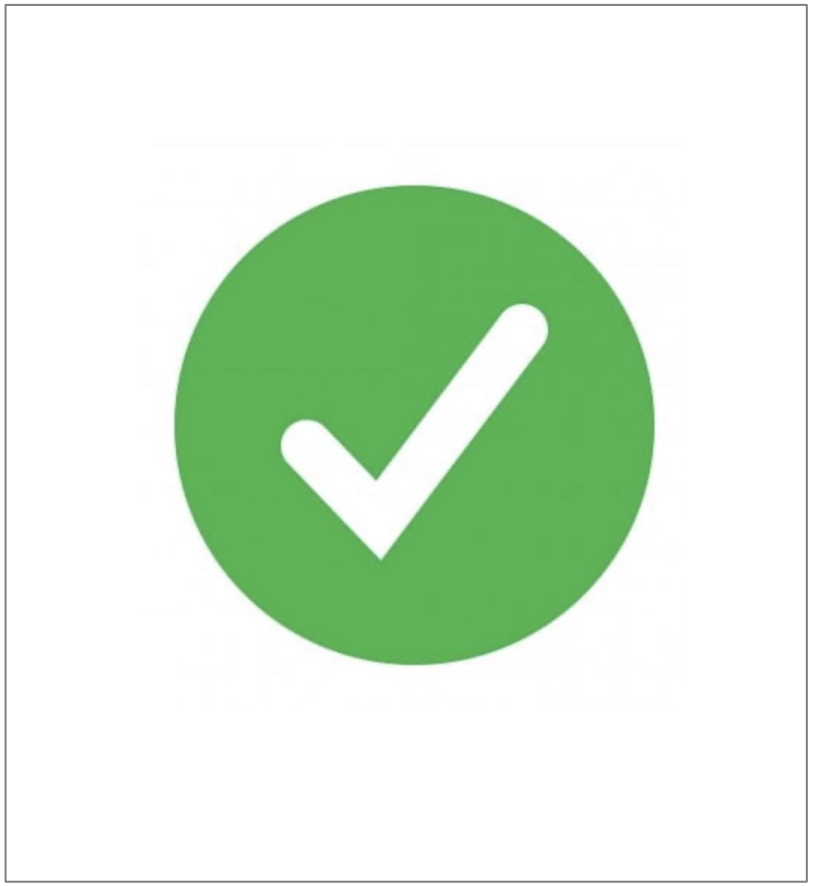
Demo

- Dimostrazione del prodotto



Conclusioni

- Obbiettivi raggiunti
  - Soddisfazione dei requisiti
  - Uso di vari pattern di design
  - Esperienza nella metodologia agile
  - Esperienza di lavoro in team
  - Comprensione dell'importanza dei test
  - Acquisizione di competenze nella scrittura dei test
- Sviluppi futuri
  - Integrare gestione di files
  - Arricchire le opzioni dei comandi



STUDENTSUPSI

Extra