

**SUPSI**

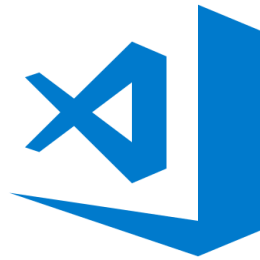
# Sviluppo di una estensione per VSCode

Studente/i	Relatore	Correlatore
Beffa Bryan	Coluzzi Massimo	-
Corso di laurea	Modulo / Codice Progetto	Anno
Ingegneria Informatica	C10652	2022 - 2023
Committente	Data	
Coluzzi Massimo	26.08.2023	

## Motivazione e contesto

- Programmazione fluent
- Librerie Nerd4J
- Visual Studio Code

```
public boolean equals( Object other )  
{  
  
}  
  
public  
{  
    return Hashcode.of( field1, field2, field3, field4 );  
}
```



Visual Studio Code

# Caratteristiche estensione

- Supporto java
- Generazione codice
- Programmazione più efficiente

Sfruttando:

- Librerie Nerd4J
- Programmazione fluent

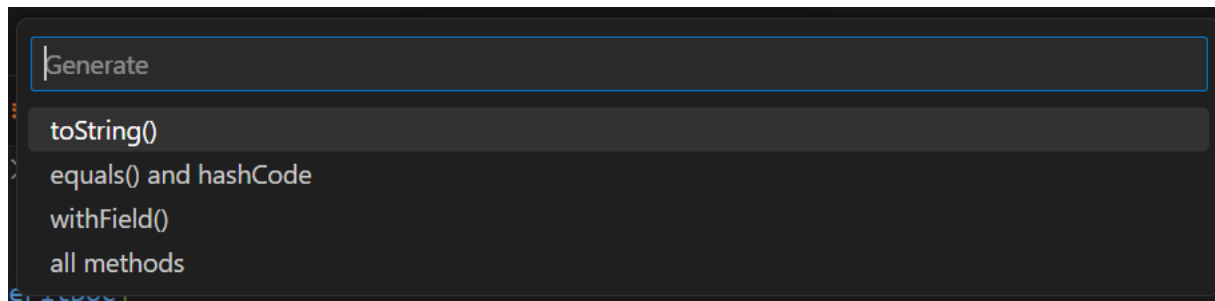
# Obiettivi

- Studio e comprensione:
  - Visual Studio Code (VSCode)
  - Sviluppo di estensioni per VSCode
  - tecniche di generazione automatica di codice
- Sviluppo:
  - Estensione per la generazione automatica di codice

# Sviluppo

Generazione codice:

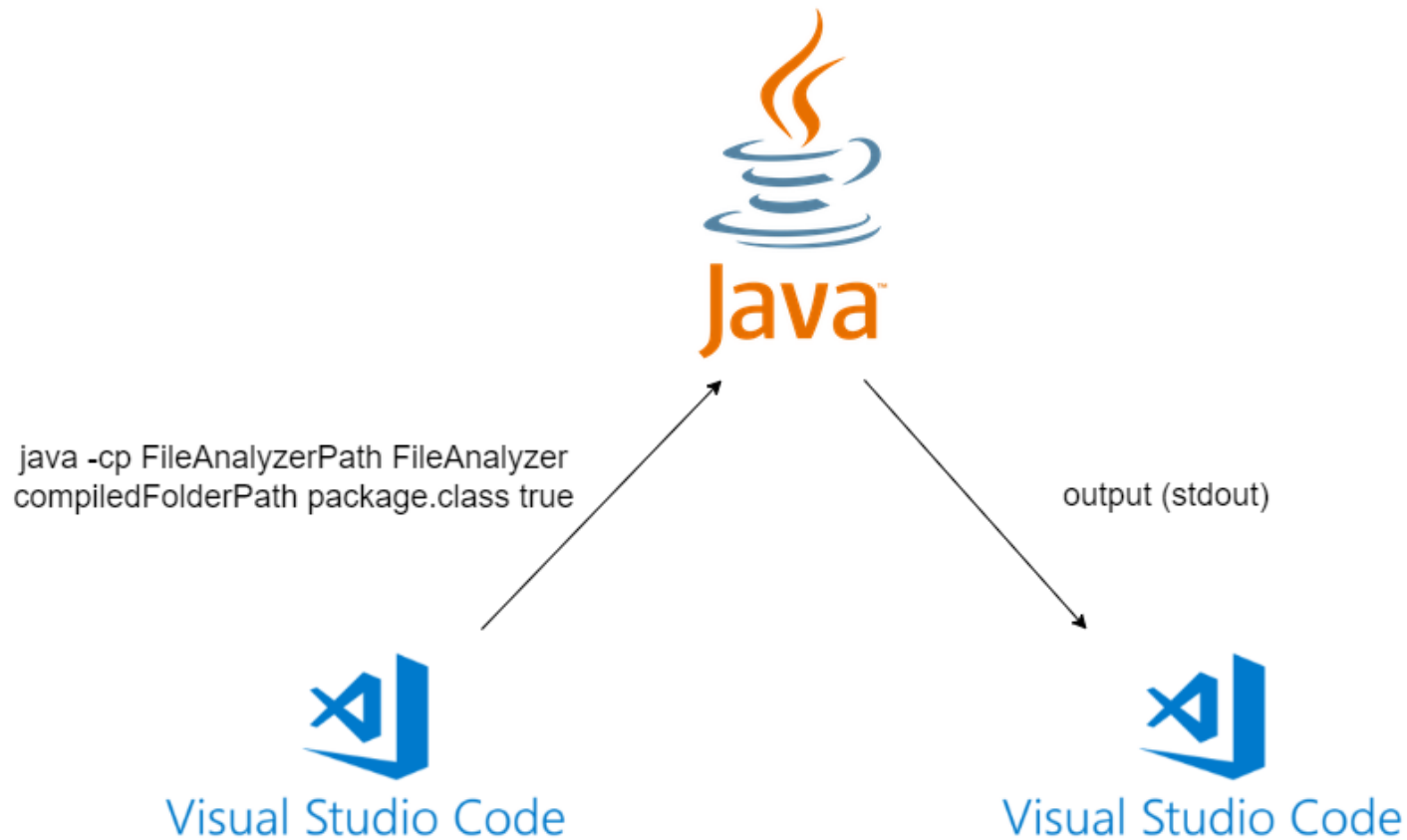
- Nerd4J:
  - toString()
  - equals() e hashCode()
- Programmazione Fluent
  - withField()



# FileAnalyzer

- Utilizza la reflection
- Metodi analisi campi classe:
  - `getParentsVisibleFields()`
  - `getClassFields()`
- Produce i seguenti output:
  - Nome della classe
  - Campi visibili o campi modificabili

## Schema funzionamento



# Gestione JDK

- Controllo versione JDK
- Configurazione JDK
- Ricompilazione FileAnalyzer

```
{  
  "java.jdt.ls.java.home": "c:\\Users\\Prova\\.jdk\\openjdk-18.0.1.1"  
}
```



## toString()

- Selezione dei campi
- Scelta della rappresentazione dati
  - likeEclipse
  - likeIntellij
  - likeFunction
  - likeTuple
  - like
- Aggiunta import automatica
- Rigenerazione e sostituzione codice

```
/**
 * {@inheritDoc}
 */
@Override
public String toString() {
    return ToString.of(this)
        .print(name: "porte", porte)
        .print(name: "brand", brand)
        .print(name: "electric", electric)
        .print(name: "registered", registered)
        .likeFunction();
}
```

## equals() e hashCode()

- Selezione dei campi
- Generazione hashCode() opzionale
- Aggiunta degli import automatica
- Rigenerazione e sostituzione codice

```
/**
 * {@inheritDoc}
 */
@Override
public boolean equals(Object other) {
    return Equals.ifSameClass(this, other,
        o -> o.brand,
        o -> o.electric,
        o -> o.registered
    );
}

/**
 * {@inheritDoc}
 */
@Override
public int hashCode() {
    return Hashcode.of(brand, electric, registered);
}
```

## withField()

Paragonabile ad un metodo setter sfruttando la programmazione fluent

- Selezione dei campi modificabili:
  - Not-final o non private ereditati
- Codice che non viene rigenerato

```
public SportCar withBrand(String value) {  
    this.brand = value;  
    return this;  
}  
  
public SportCar withElectric(boolean value) {  
    this.electric = value;  
    return this;  
}
```

# Code snippets

Tramite auto-completamento dell'editor:

- Import delle librerie
- Import delle dipendenze
  - Apache Maven, Ant, Buildr
  - Groovy Grape
  - Grails
  - Leiningen
  - SBT

```
src > main > java > com > mvnproject > Car.java > ...  
1  package com.mvnproject;  
2  
3  imp  
4  import  
5  import nerd4j.utils.cache      cache.*  
6  import nerd4j.utils.lang      lang.*  
7  pub import nerd4j.utils.math   math.*  
8  import nerd4j.utils.tuple     tuple.*  
9  public Car() {  
10 }
```

# Comandi estensione

Generate:

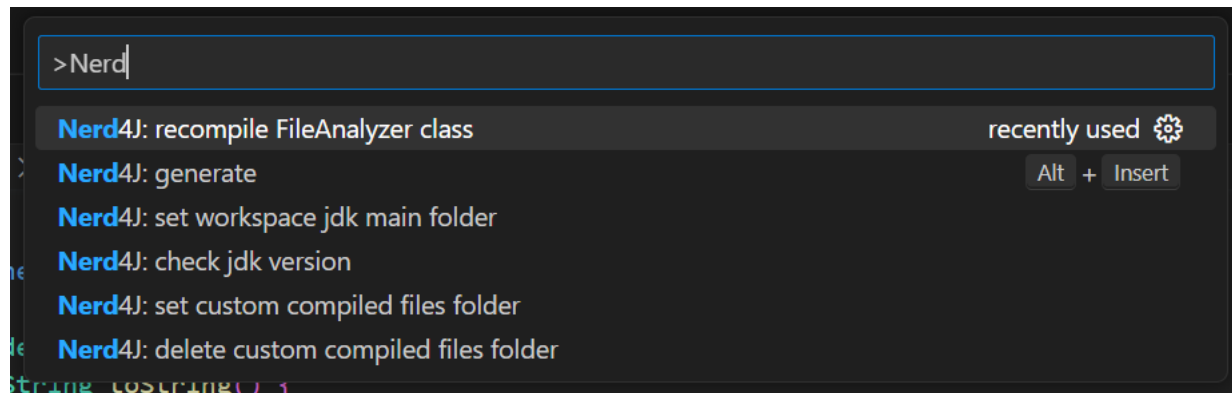
- toString(), equals(), hashCode() e withField()

JDK:

- Controllo versione, configurazione e ricompilazione

Compiled file folder:

- Configurazione percorso

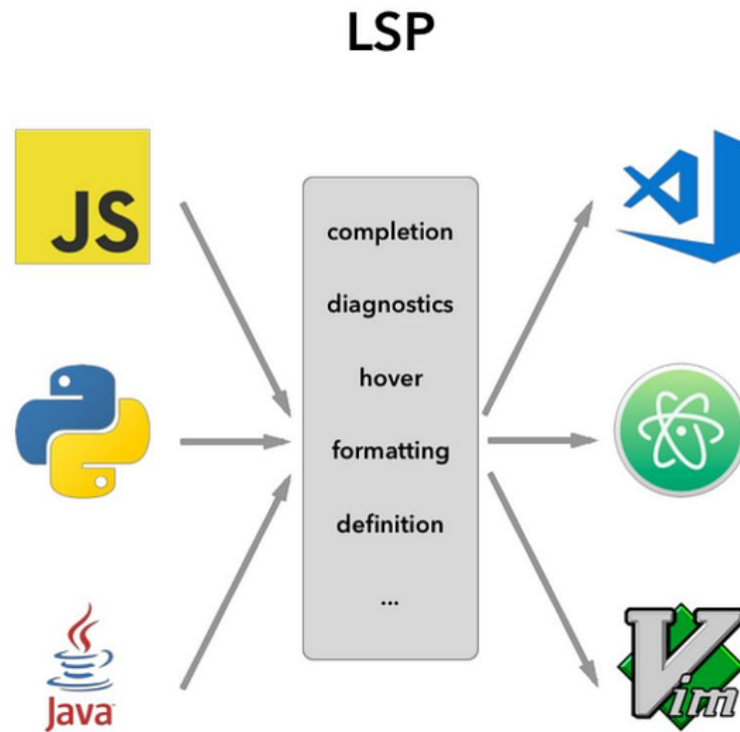


# Risultati

- Generazione automatica del codice
- Rigenerazione e sostituzione codice
- Gestione della JDK del progetto
- Gestione della cartella file compilati
- Snippet import Nerd4J
- Snippet dipendenze Nerd4J

## Sviluppi futuri

- Aggiunta snippet di codice
- Implementazione LSP
- Compilazione sorgenti



# Conclusioni

- Sviluppo semplificato e velocizzato
- Generazione e sostituzione codice
- Selezione campi ereditati



Grazie per l'attenzione