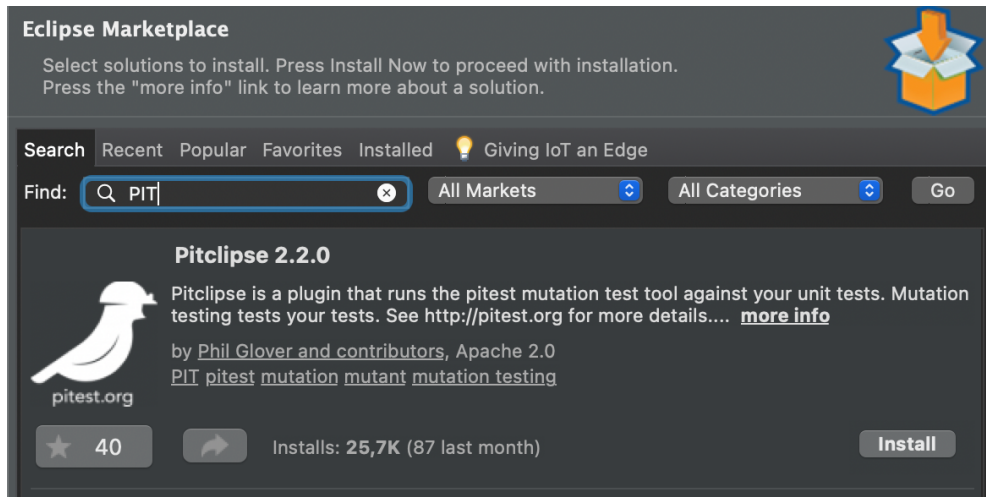


# Tutorial PITest

Testing e Verifica del Software  
Dott. Andrea Bombarda

## Mutation testing con PIT

Per prima cosa installiamo il plugin di PIT dall'eclipse Marketplace:



Accettiamo tutti i termini di licenza ed i certificati per confermare l'installazione.

Prendiamo ora il progetto sul quale vogliamo eseguire mutation testing e selezioniamo o una classe di test, oppure un intero package. Questi test saranno coloro che verranno eseguiti per valutare il mutation score ottenuto dalla test suite.

Clicchiamo con il tasto destro, quindi su [Run As] > [PIT Mutation Test].

In automatico, PIT eseguirà le mutazioni di default rispetto alle classi richiamate all'interno dei casi di test selezionati e produrrà un report simile al seguente:

```
> org.pitest.mutationtest.engine.gregor.mutators.PrimitiveReturnsMutator
>> Generated 4 Killed 3 (75%)
> KILLED 3 SURVIVED 1 TIMED_OUT 0 NON_VIABLE 0
> MEMORY_ERROR 0 NOT_STARTED 0 STARTED 0 RUN_ERROR 0
> NO_COVERAGE 0
=====
- Timings
=====
> pre-scan for mutations : < 1 second
> scan classpath : < 1 second
> coverage and dependency analysis : < 1 second
> build mutation tests : < 1 second
> run mutation analysis : < 1 second
=====
> Total : 1 seconds
=====
- Statistics
=====
>> Line Coverage: 12/12 (100%)
>> Generated 8 mutations Killed 7 (88%)
>> Mutations with no coverage 0. Test strength 88%
>> Ran 25 tests (3.12 tests per mutation)
```

ATTENZIONE: E' possibile anche selezionare mutazioni diverse. Per fare questo, quando clicchiamo su [Run As] selezioniamo [Run Configurations...] e nella schermata di configurazione di PIT scegliamo a quali mutazioni siamo interessati:

Name: RegressionTest0

PIT

Mutators

Arguments

JRE

Classpath

Common

Select the mutators used to alter the code. [See the documentation on Pitest.org](https://pitest.org)

Mutate with: ☒ Defaults ☐ Stronger defaults ☐ All ☐ Old defaults ☐ Mutators selected below

Name	Description
<input checked="" type="checkbox"/> Invert Negatives	Inverts negation of integer and floating point numbers
<input type="checkbox"/> Return Values	Mutates the return values of method calls. Depending on the return type of the method an
<input type="checkbox"/> Inline Constant	Mutates inline constants. An inline constant is a literal value assigned to a non-final variabl
<input checked="" type="checkbox"/> Math	Replaces binary arithmetic operations for either integer or floating-point arithmetic with ar
<input checked="" type="checkbox"/> Void Method Call	Removes method calls to void methods
<input checked="" type="checkbox"/> Negate Conditionals	Mutates all conditionals found
<input checked="" type="checkbox"/> Conditionals Boundary	Replaces the relational operators <, <=, >, >=
<input checked="" type="checkbox"/> Increments	Mutates increments, decrements and assignment increments and decrements of local vari
<input type="checkbox"/> Remove Increments	Removes local variable increments
<input type="checkbox"/> Non Void Method Call	Removes method calls to non void methods. Their return value is replaced by the Java Def
<input type="checkbox"/> Constructor Call	Replaces constructor calls with null values
<input type="checkbox"/> Remove Equal Conditionals If	Remove equal conditions and replace with true, execute if part
<input type="checkbox"/> Remove Equal Conditionals Else	Remove equal conditions and replace with false, execute else part
<input type="checkbox"/> Remove Order Checks If	Remove order conditions and replace with true, execute if part
<input type="checkbox"/> Remove Order Checks Else	Remove order conditions and replace with false, execute else part
<input type="checkbox"/> Remove Conditionals	Removes all conditionals statements such that the guarded statements always execute

Revert

Apply