Unit Testing & PHPUNIT

Unit Testing e' un'attivita' di verifica della correttezza di un sistema software, che procede una unita' alla volta, dove una unita' e'

- una funzione
- un metodo

I do not have time to write tests, as I am too busy debugging!

serve a determinare la correttezza delle singole componenti del programma

cioe' che il comportamento (delle parti) e' quello atteso, cosi' che il comportamento del Sistema, nelle varie fasi del suo uso, sia quello atteso

Unit Testing & PHPUNIT

Unit Testing e' un'attivita' di verifica della correttezza di un sistema software, che procede una unita' alla volta, dove una unita' e'

- una funzione
- un metodo

I do not have time to write tests, as I am too busy debugging!

Un TEST e' un modulo software (insomma un pezzo di codice) che prova un comportamento di una funzione, in certe condizioni, per verificare che il comportamento sia quello atteso.

L'idea e' che i test siano

- sviluppati durante lo sviluppo stesso del codice del Sistema
- molto veloci e circoscritti
- raccolti in gruppi (testsuite) che riguardino il test di una o certe funzioni
- ripetibili molte volte, anche dopo che si pensa che una certa funzione sia corretta e si e' passati ad altro … cosi' da evitare che il Sistema regredisca (funzioni che erano corrette smettono di esserlo per effetto di cambiamenti al contorno).

La vita di un test

In generale un TEST e' realizzato per compiere questa sequenza di azioni:

- preparare il terreno per il test (variabili di appoggio, oggetti … i dati di prova …. SetUP)
    - es. $x=10; $y=11;
- eseguire la unita' da testare sui dati di prova: qui si devono ottenere dei risultati che possano essere controllati;
    - es. $result = somma($x,$y);
- Fare un'ASSERZIONE, cioe' dichiarare che vale (dovrebbe valere) una certa proprieta' sui risultati della chiamata;
    - es. assertEqual(21, $result)
- eventualmente chiudere il test bonificando l'ambiente di esecuzione dagli oggetti creati ad hoc, o preparando operazioni di garbage collection (TearDown)

- consistenza (se ripetuto da' i medesimi risultati ...)
- atomico (non e' che puo' passare o non passare "un po'")
- (I test in un gruppo dovrebbero dare i medesimi risultati indipendentemente dal loro ordine)
- self-descriptive
  - che di solito significa "il nome deve essere evocativo"
    - es. `effetto collaterale su som durante una somma`
    - es. `No effetto collaterale su som durante una sottrazione`
  - ma anche che ci sono commenti esplicativi che chiariscono cosa e' successo, ad esempio in caso di fallimento

Le qualita' di un test

- consistenza (se ripetuto da' i medesimi risultati …)
- atomico (non e' che puo' passare o non passare "un po'"
- (I test in un gruppo dovrebbero dare i medesimi risultati indipendentemente dal loro ordine)
- self-descriptive
  - che di solito significa "il nome deve essere evocativo"
    - es. effetto collaterale su som durante una somma
    - es. No effetto collaterale su som durante una sottrazione
  - ma anche che ci sono commenti esplicativi che chiariscono cosa e' successo, ad esempio in caso di fallimento

- un test deve controllare un solo comportamento, o il comportamento dell'unita' in un solo scenario (MonoResponsabile)
- semplice: anche un if puo' essere troppo, non parliamo di un while …
- automatizzabile (cioe' interi flussi di test dovrebbero partire automaticamente dopo ogni modifica del software da testare)
- non interattivo

Ricadute (utili ...)

- sul "refactoring": quando si fanno modifiche migliorative, sapere che tutto quello che c'e' intorno, e anche la parte modificata, funzionavano, aiuta a circoscrivere eventuali problemi.
- l'insieme di test sviluppati dal programmatore mentre lavorava, costituisce una sorta di documentazione in filigrana, utile a capire quale logica e comportamento erano attesi/e durante lo sviluppo.

phpunit

- ambiente per lo sviluppo e l'esecuzione di test per sistemi sviluppati in php
- https://phpunit.de/
- In sostanza un file di test contiene l'estensione di una classe di Sistema (TestCase), in cui definire varie funzioni, ciascuna delle quali realizza un test.
- L'esecuzione dei test in una di queste "suite" avviene con il comando `phpunit` e produce in output l'elenco successi/fallimenti dei test

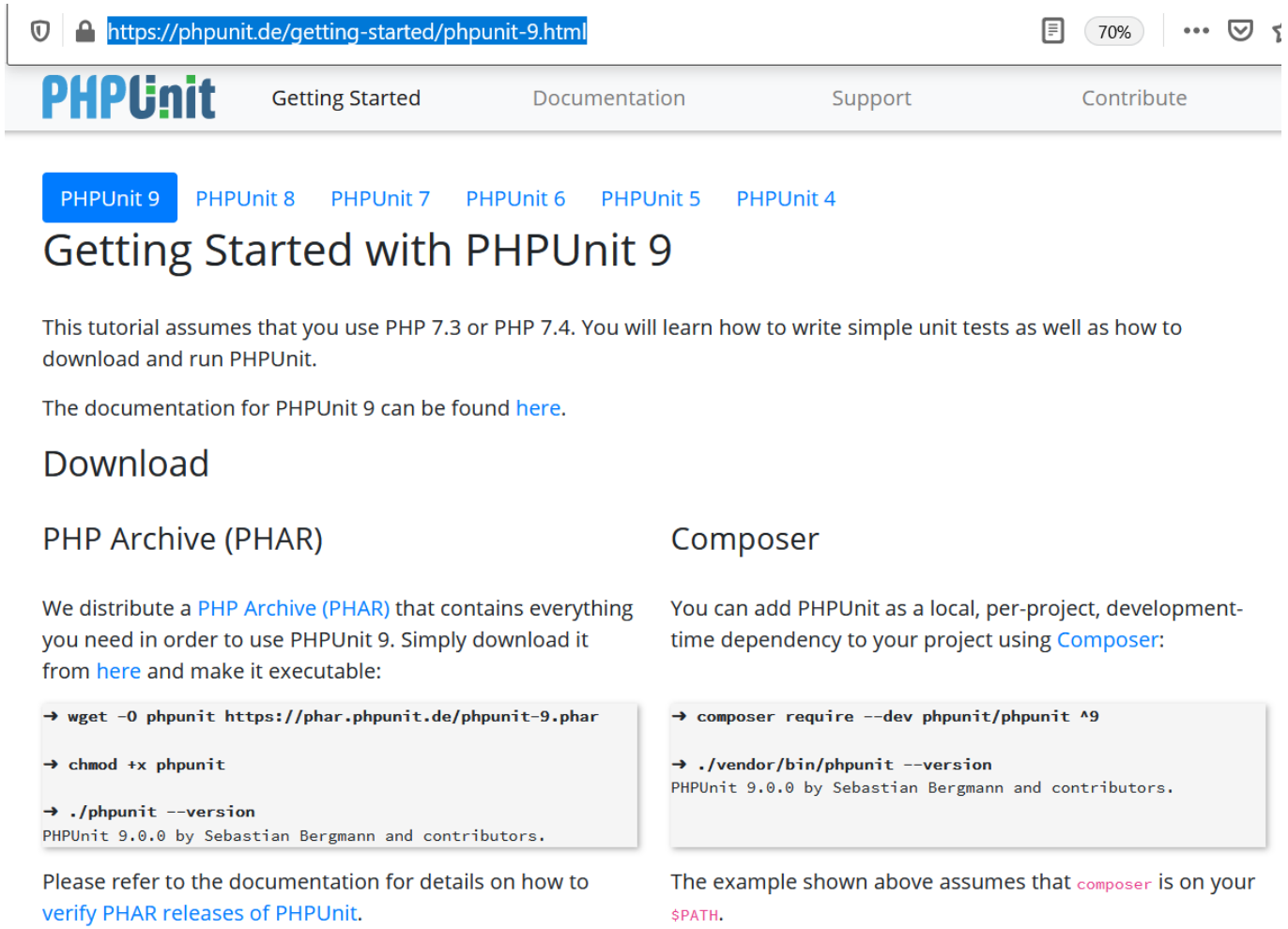
phpunit

- ambiente per lo sviluppo e l'esecuzione di test per sistemi sviluppati in php
- https://phpunit.de/
- In sostanza un file di test contiene l'estensione di una classe di Sistema (TestCase), in cui definire varie funzioni, ciascuna delle quali realizza un test.
- L'esecuzione dei test in una di queste "suite" avviane con il comando **`phpunit`** e produce in output l'elenco successi/fallimenti dei test
- l'istallazione di phpunit puo' avvenire direttamente dal file phar fornito da phpunit.de, oppure usando composer
- https://getcomposer.org/

phpunit: istallazione attraverso composer

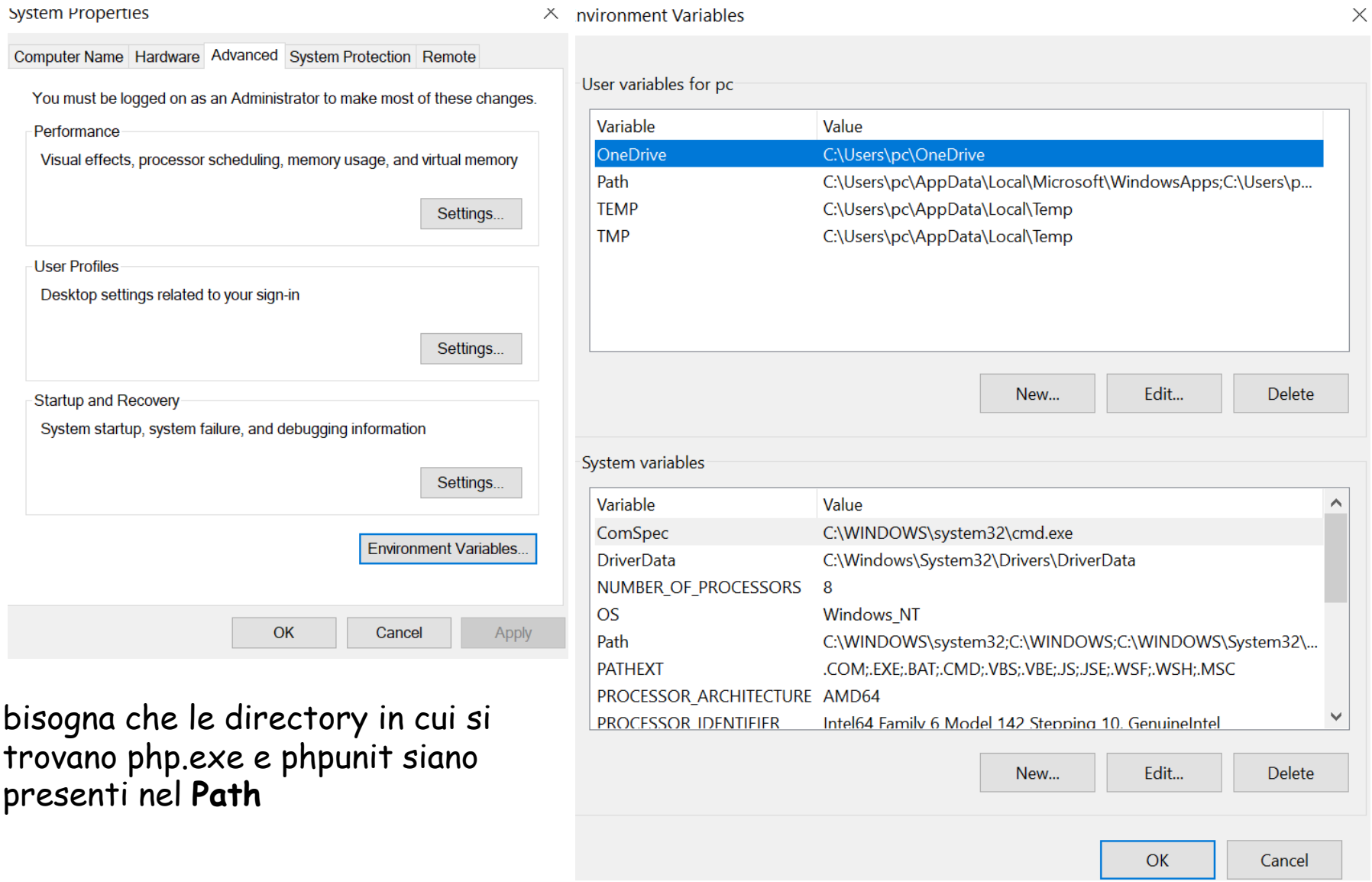
https://phpunit.de/getting-started/phpunit-9.html

l'istallazione di phpunit puo' avvenire direttamente dal file phar fornito da phpunit.de, oppure usando composer https://getcomposer.org/



phpunit sotto windows

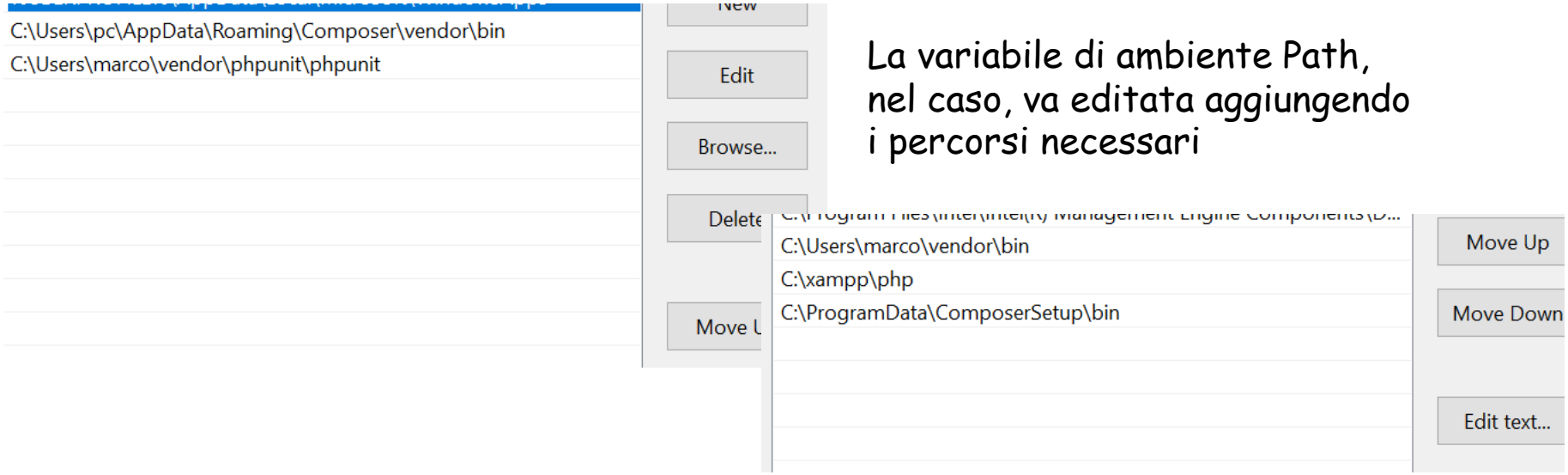
- bisogna affrontare con pazienza i soliti problem di path …



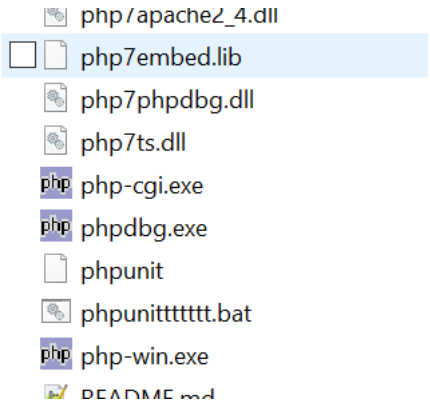
bisogna che le directory in cui si trovano php.exe e phpunit siano presenti nel **Path**

phpunit sotto windows

- bisogna affrontare con pazienza i soliti problem di path …



La variabile di ambiente Path, nel caso, va editata aggiungendo i percorsi necessari



qualcuno/a puo' avere (es. in xampp, directory php) gia' istallato phpunit. C'e' un phpunit.bat che viene eseguito automaticamente chiamando phpunit da linea di commando. Ma forse e' meglio "sterilizzare" questa istallazione e procedure con quella fatta secondo le slide precedenti. Per farlo basta cambiare nome al .bat, cosi' non partira' quando scriviamo phpunit nella linea di comando.

risorse

istallazione

installare composer: https://getcomposer.org/

e poi seguire https://phpunit.de/getting-started/phpunit-9.html per istallare phpunit

la documentazione (fondamentale e che copre molto piu' della nostra introduzione ...) e' in
https://phpunit.de/documentation.html
da cui si arriva, per gli anglofoni e le anglofone, in https://phpunit.readthedocs.io/en/9.5/

[
Per l'istallazione sotto windows possono esserci difficolta' (per i path piu' che altro, cioe' per le variabili di ambiente di windows).
Il prof e' convinto che quel che vi ha detto e efatto vedere basta e avanza ;)
Ma qui c'e' un tutorial per istallazione: https://www.youtube.com/watch?v=fYuVyQC-edQ
e' basato su una versione vecchia di phpunit, ma i principi valgono ancora. Usatelo se proprio non riuscite a istallare altrimenti.
]

letture utili

In https://phpunit.readthedocs.io/en/9.5/ le letture consiglilabili sono in particolare

- *Sezione 2 writing tests*
- *Sezione 3 The command line and testdox*
- *Appendix1 Assertions*
- *Appendix3 The configuration file …phpunit.xml*

risorse utili

calculator class con esempio che parte dalla istallazione; qualcosa qui e la' di alternativo a quello che abbiamo fatto a lezione. Ma e' una visione utile. SI puo' provare a seguirlo passo passo come fatto per la lezione. Guardate anche come usa l'elemento testsuite in phpunit.xml.
https://www.youtube.com/watch?v=a5ZKCFINUkU

Piu' ricca, un po' piu' difficile da seguire
https://www.youtube.com/watch?v=84j61_aI0q8

risorse

risorse utili per cose che non abbiamo toccate a lezione

Su namespace e specifica psr-4 (noi abbiamo usato solo classmap)
https://www.youtube.com/watch?v=VGSerlMoIrY

un'alternativa alla documentazione ufficiale di phpunit e' su w3resources. Mi sembra che sia solo una ripetizione della documentazione, ma potrei sbagliare ...
Qui in particolare si parla di Risky Test https://www.w3resource.com/php/PHPUnit/risky-tests.php

Una fixture e' un "apparato" che crea un ambiente per i test da fare, con setUp() e poi se ne libera con tearDown().
Nella documentazione ufficiale c'e un esempio che usa una classe Stack e che puo' essere istruttivo.
https://phpunit.de/manual/3.7/en/fixtures.html

uso di mock: https://www.youtube.com/watch?v=kkU43JdJQBE

tutorial su uso di phpunit con un database. Usa phpunit/dbunit, una estensione ora abbandonata.
https://www.wingsquare.com/blog/phpunit-database-testing/
In effetti non c'e' piu' una libreria ufficiale per gestire un db con phpunit, al di fuori di framework organizzati, come laravel, o (collezioni di) librerie come doctrine.