**Maven**

* **Εγκατάσταση του Maven στην γραμμή εντολών**
* <https://maven.apache.org/download.cgi>
* Download Binary zip archive
* Unzip το αρχείο στον φάκελο που επιθυμείτε
* My Computer -> Properties -> Advanced system settings -> Environment Variables
* Στις μεταβλητές περιβάλλοντος στο πρώτο πλαίσιο πατάω “new” και ορίζω μια νέα μεταβλητή με όνομα MAVEN\_HOME και value το path που βρίσκεται το αρχείο. Πχ. C:\apache-maven-3.3.9
* Στο ίδιο πλαίσιο δημιουργώ μια ακόμη μεταβλητή με όνομα M2\_HOME και το ίδιο path.
* Στο δεύτερο πλαίσιο (System Variables) κάνω edit στο path της java και προσθέτω το προηγούμενο path + την κατάληξη ” \bin” στο τέλος .

Για να επιβεβαιωθώ πως έχει γίνει σωστή εγκατάσταση:

* Κλείνω τη γραμμή εντολών αν αυτή ήταν ήδη ανοιχτή και την ανοίγω εκ νέου.
* Πληκτρολογώ στην γραμμή εντολών mvn –version και αναμένω να μου δείξει την έκδοση της maven που έχω στον υπολογιστή μου αν εκείνη έχει εγκατασταθεί σωστά.
* Στη συνέχεια αν πατήσετε mvn –help θα σας βγάλει όλες τις βασικές εντολές της και μια περιγραφή για την κάθε μία.
* **Εγκατάσταση του Maven στο eclipse**
* Eclipse -> Help -> Eclipse Marketplace
* Επιλογή του Maven Integration for Eclipse
* Finish για εγκατάσταση
* **Δημιουργία πρώτου Maven Project στη γραμμή Εντολών**
* Ανοίγουμε την γραμμή εντολών και με το “cd” ορίζουμε τον φάκελο που θέλουμε να πάμε ώστε να δημιουργηθεί εκεί το project.
* Γράφουμε mvn archetype:generate (Η εντολή αυτή κατεβάζει plugins από το internet τα οποία είναι απαραίτητα για την υλοποίηση του project).
* Μόλις τελειώσει η διαδικασία αυτή, θα βγει στην οθόνη σας μια πρόταση όπου θα ζητηθεί να δοθεί ένας αριθμός από τα παραπάνω plugin. Κάθε plugin έχει το δικό του αριθμό και είναι στην επιλογή του καθενός το τι θα χρησιμοποιήσει (οι αριθμοί και το ποια είναι η λειτουργία του κάθε plugin φαίνεται στην γραμμή εντολών αφού έχει εκτελεστεί η προηγούμενη εντολή). Πάντα μετά την πρόταση δίνεται ένας προτεινόμενος αριθμός για το ποιο plugin να χρησιμοποιήσουμε οπότε κατά προτίμηση βάζουμε αυτόν.
* Στη συνέχεια ζητείται ποια έκδοση να χρησιμοποιηθεί για το archetype της maven. Όπως είναι φυσικό πάντα επιλέγουμε τον αριθμό της τελευταίας έκδοσης. (πχ αν είναι 6 οι εκδόσεις που εμφανίζονται, γράφουμε τον αριθμό 6). Η τελευταία έκδοση θα προτείνεται ξανά ως η καλύτερη επιλογή.
* Στην συνέχεια ορίζεται το groupid του project και γενικότερα της ομάδας μας. Εμείς ως τώρα χρησιμοποιούσαμε το com.Webmasters.
* Τώρα ζητείται το artifactid δηλαδή η ονομασία του project.
* Στην έκδοση που θα ζητηθεί στη συνέχεια γράφουμε ακριβώς το προτεινόμενο δηλαδή 1.0-SNAPSHOT.
* Έπειτα θα ζητηθεί το package όπου το προτεινόμενο θα ζητηθεί να συμπίπτει με την ονομασία του groupid. Οπότε ξαναγράφουμε για παράδειγμα το com.Webmasters.
* Τέλος θα εμφανιστούν όλα τα στοιχεία που έχουμε ζητάει και θα ζητάει επιβεβαίωση με ένα “Y” για να δημιουργηθεί ο φάκελος όπου θα περιέχει το project μας.
* Εφόσον όλα έχουν κυλήσει ομαλά, θα εμφανιστεί στην οθόνη “build success”.
* Στη συνέχεια εφόσον υπάρχουν όλες οι κλάσεις του project μας στον φάκελο, κάνουμε τα εξής βήματα τα οποία αναλύονται πιο κάτω με τη χρήση eclipse ( είναι παρόμοια διαδικάσια):

1. mvn clean
2. mvn compile
3. mvn test-compile
4. mvn test
5. mvn install

Υ.Γ. **Αν έχετε κάνει σωστά τις εντολές, με την εκτέλεσή τους θα σας βγάζει «build success”. Προσοχή οι εντολές εξαρτώνται από τις προηγούμενές τους οπότε θα πρέπει να έχουν υλοποιηθεί όλες σωστά για να δημιουργηθεί το αρχείο .jar.**

* **Δημιουργία πρώτου Maven Project στo Eclipse**
* File -> New -> Project (Προσοχή όχι το Java Project, σκέτο Project)
* Στο παράθυρο που ανοίγει ψάχνουμε για τον φάκελο Maven -> Maven Project
* Επιλέγουμε («τικάρουμε») create a simple project
* Εισάγουμε τα στοιχεία του project (γίνεται αναφορά πιο πάνω για το τι σημαίνει το καθένα).
* Αν δεν σας είναι ήδη ανοιχτό, από τις ρυθμίσεις του eclipse, εμφανίστε τον πίνακα package explorer. Θα εμφανιστεί αριστερά από το πλαίσιο στο οποίο γράφεται ο κώδικας.
* Από το παράθυρο του package explorer θα φαίνεται ο φάκελος που έχει δημιουργηθεί με τις διακλαδώσεις του ( επιπλέον φακέλους μέσα στον πρώτο).
* Για να δούμε το αρχείο pom.xml (τον κώδικά του) το ανοίγουμε (βρίσκεται κάτω κάτω στο package explorer) και στο παράθυρο που μας βγάζει κοιτάμε στο κάτω μέρος της σελίδας όπου έχει φύλλα εργασίας σαν το excel. Πάμε στο τελευταίο.
* Στο package explorer επιλέγουμε src/main/java, δεξι κλικ -> new -> class
* Eισάγουμε το package (com.Webmasters) και το όνομα της κλάσης).
* Γράφουμε την κλάση μας
* Επιλέγουμε src/test/java -> new -> other
* Στο παράθυρο που ανοίγει βρίσκουμε και επιλέγουμε junit -> junit test case (είναι βιβλιοθήκη που απαιτείται για το project μας)
* Εισάγουμε πακέτο, όνομα κλάσης Test
* Κάνουμε add την βιβλιοθήκη στο νέο παράθυρο που θα εμφανιστεί(δηλαδή δεν αλλάζουμε τίποτα, απλά πατάμε οκ)
* **Στον αρχικό φάκελο που δημιουργήσαμε πατάμε δεξί κλικ -> run as:**

1. 5 Maven build -> goals -> clean (καθαρίζει όλα τα compiles (εκτελέσιμα αρχεία και διαγράφει το φάκελο targets όπου αποθηκεύονται αυτά μέσα).
2. 6 Maven build… -> goals -> compile (μεταγλωττίζει όλα τα αρχεία μέσα στον φάκελο)
3. 6 Maven build… -> goals -> test- compile (μεταγλωττίζει όλα τα test αρχεία μέσα στον φάκελο)
4. 6 Maven build… -> goals -> test (τρέχει τα test classes και εμφανίζει που βρίσκονται τα λάθη και πόσα είναι αυτά σε ειδικό αρχείο)
5. 6 Maven build… -> goals -> install (εφόσον όλα έχουν κυλήσει σωστά δημιουργείται το τελικό αρχείο jar το οποίο εφαρμόζει όλες τις κλάσεις μαζί και μας δίνει το αποτέλεσμα του project μας).

Υ.Γ. **Αν έχετε κάνει σωστά τις εντολές, με την εκτέλεσή τους θα σας βγάζει «build success”. Προσοχή οι εντολές εξαρτώνται από τις προηγούμενές τους οπότε θα πρέπει να έχουν υλοποιηθεί όλες σωστά για να δημιουργηθεί το αρχείο .jar.**