

Care din urmatoarele afirmatii legate de Build Tools sunt FALSE?

- a. Gradle nu poate crea artefacte de maven, ci pentru a face acest lucru, are nevoie de un plugin, numit Maven. *e compatibil cu maven, plugin maven-publish*
- b. Un artefact de Maven contine minim 3 elemente: groupId, artifactID si version.
- c. Spre deosebire de Maven, Gradle nu suporta dependente trazitive. *suportă*
- d. Comanda **mvn install** creeaza un jar, un artefact de maven si ii face deploy in repository-ul Maven Central. *nu ii face deploy (mvn deploy)*
- e. Gradle nu poate folosi ca dependente artefacte care au fost "build-uite" cu Maven. *căt timp are acces la repo, nu contestă*

Care din urmatoarele afirmatii legate de executia comenii **gradle build**, in gradle, sunt ADEVARATE?

- a. Executa un task numit **build**
- b. Va executa task-ul **build** si toate task-urile de care acesta depinde
- c. Apeleaza un **goal** numit **build** *task, (goal e la maven)*
- d. Va executa faza **build** din "default lifecycle" *(lifecycle e la maven)*
- e. Executa plugin-ul numit **build** *task + plugin : (tot la maven)*

Care din urmatoarele afirmatii legate de Build Tools sunt FALSE?

- a. In Gradle, termenul de **task** reprezinta o unitate de executie, care poate fi executata folosind comanda "**gradle numele_taskului**". *fără plugin : (tot la maven)*
- b. In Gradle, termenul de **goal** reprezinta o unitate de executie, iar termenul de **plugin** reprezinta o colectie de **goal-uri**.
- c. In Gradle, termenul de **task** reprezinta o unitate de executie, care poate fi executata folosind comanda "**gradle numele_pluginului:numele_taskului**".
- d. In Gradle, termenul de **goal** reprezinta o unitate de executie, care poate fi executata folosind comanda "**gradle numele_goalului**".
- e. In Gradle, termenul de **goal** reprezinta o unitate de executie, care poate fi executata folosind comanda "**gradle numele_pluginului:numele_goalului**".

Care din urmatoarele afirmatii legate de executia comenzii `mvn package`, in maven, sunt FALSE?

- a. Va executa toate **goal-urile** care sunt asignate tuturor fazelor care preced faza de package (inclusiv a celor din faza package) *nu apela un goal, ci un lifecycle phase*
- b. Apelaaza un goal numit **package** *nu apela un goal, ci un lifecycle phase*
- c. Printeaza toate pachetele gasite in proiect *nu apela*
- d. Va executa task-ul **package** si toate task-urile de care acesta depinde *nu task, goal*
- e. Va executa doar faza **package** din "default lifecycle" *+ precedent*

Care din urmatoarele afirmatii legate de Build Tools sunt FALSE?

- a. Un proiect NU poate contine atat un fisier **pom.xml** cat si un fisier **build.gradle**, deoarece acestea ar duce la conflicte in momentul build-urilor. *peste*
- b. Un proiect poate contine atat un fisier **pom.xml** cat si un fisier **build.gradle**, si poate fi build-uit atat cu Maven, cat si cu Gradle, deoarece atat input-urile, cat si output-urile tool-urilor sunt complet distincte. *✓*
- c. Un proiect NU poate contine atat un fisier **pom.xml** cat si un fisier **build.gradle**, si poate fi build-uit atat cu Maven, cat si cu Gradle, deoarece chiar daca input-urile sunt distincte, folder-ul de output este acelasi. *peste*
- d. Un proiect poate contine atat un fisier **pom.xml** cat si un fisier **build.gradle**, deoarece fisierul **pom.xml** contine informatii legate doar de compilare, iar cel **build.gradle** informatii doar despre testare. *fals*
- e. Un proiect NU poate contine atat un fisier **pom.xml** cat si un fisier **build.gradle**, deoarece nu putem genera decat un wrapper intr-un proiect.

a. "Un proiect NU poate contine atât un fișier pom.xml cât și un fișier build.gradle, deoarece acestea ar duce la conflicte în momentul build-urilor."

→ FALS

Maven și Gradle sunt build tools separate. Nu se încurcă unul cu celălalt. Depinde ce comenzi rulezi: mvn build folosește doar pom.xml, iar gradle build folosește build.gradle.

b. "Un proiect poate conține atât un fișier pom.xml cât și un fișier build.gradle, și poate fi build-uit atât cu Maven, cât și cu Gradle, deoarece atât input-urile, cât și output-urile tool-urilor sunt complet distincte."

→ ADEVĂRAT

Exact, poți avea ambele, tool-urile sunt independente, și input-urile (configurările) sunt separate.

c. "Un proiect NU poate conține atât un fișier pom.xml cât și un fișier build.gradle, deoarece chiar dacă input-urile sunt distincte, folder-ul de output este același."

→ FALS

Chiar dacă ambele tool-uri folosesc implicit același folder (target pentru Maven, build pentru Gradle), se pot configura foldere de output distincte fără probleme.

d. "Un proiect poate conține atât un fișier pom.xml cât și un fișier build.gradle, deoarece fișierul pom.xml conține informații legate doar de compilare, iar cel build.gradle informații doar despre testare."

→ FALS

Nu e adevărat, ambele fișiere (pom.xml și build.gradle) pot conține detalii despre compilare, testare, packaging, deployment etc.

e. "Un proiect NU poate conține atât un fișier pom.xml cât și un fișier build.gradle, deoarece nu putem genera decât un wrapper într-un proiect."

→ FALS

Fiecare tool are wrapper-ul său:

- Maven: mvnw
- Gradle: gradlew

Le poți avea pe ambele în același proiect fără probleme.

Care din urmatoarele afirmatii legate de Build Tools sunt ADEVARATE?

- a. In Gradle, termenul de **goal** reprezinta o unitate de executie, care poate fi executata folosind comanda "gradle numele_pluginului:numele_goalului".
- b. In Gradle, un plugin poate fi executat folosind sintaxa "gradle numele_pluginului" *gradle task*
- c. In Gradle, termenul de **task** reprezinta o unitate de executie, care poate fi executata folosind comanda "gradle numele_pluginului:numele_taskului". *nope, doar task*
- d. In Gradle, termenul de **task** reprezinta o unitate de executie, iar termenul de **plugin** reprezinta o colectie de **task-uri**.
- e. In Gradle, termenul de **task** reprezinta o unitate de executie, care poate fi executata folosind comanda "gradle numele_taskului". *✓*

✓ Concluzie:

Litera	Adevărat / Fals	Explicație
a	<input checked="" type="checkbox"/> Fals	"Goal" e din Maven, nu Gradle.
b	<input checked="" type="checkbox"/> Fals	Plugin-urile nu se execută direct.
c	<input checked="" type="checkbox"/> Fals	Nu se folosește plugin:task, ci doar gradle task.
d	<input checked="" type="checkbox"/> Adevarat	Plugin-urile definesc task-uri, iar task-urile sunt unitățile de execuție.
e	<input checked="" type="checkbox"/> Adevarat	Task-urile se execută simplu: gradle nume_task.

Care din urmatoarele afirmatii legate de executia comenzii mvn package, in maven, sunt ADEVARATE?

- a. Va executa task-ul **package** si toate task-urile de care acesta depinde
- b. Executa plugin-ul numit **package** *phase*
- c. Toate variantele sunt false
- d. Apeleaza un goal numit **package**
- e. Va executa doar faza **package** din "default lifecycle" *+ precedente*

Care din urmatoarele afirmatii legate de executia comenzii gradle build, in gradle, sunt ADEVARATE?

- a. Executa un task numit **build**
- b. Va executa task-ul **build** si toate task-urile de care acesta depinde
- c. Apeleaza un goal numit **build**
- d. Va executa faza **build** din "default lifecycle" *maven*
- e. Executa plugin-ul numit **build**

Care din urmatoarele afirmatii legate de executia comenzii `mvn package`, in maven, sunt ADEVARATE?

- a. Va executa task-ul **package** si toate task-urile de care acesta depinde
- b. Va executa toate **goal-urile** care sunt asignate tuturor fazelor care preced faza de package (inclusiv a celor din faza package) phase
- c. Va executa doar faza **package** din "default lifecycle"
- d. Executa plugin-ul numit **package**
- e. Printeaza toate pachetele gasite in proiect

Care din urmatoarele afirmatii legate de Build Tools sunt FALSE?

- a. In Gradle, termenul de **task** reprezinta o unitate de executie, care poate fi executata folosind comanda "`gradle numele_taskului`".
- b. In Gradle, termenul de **goal** reprezinta o unitate de executie, iar termenul de **plugin** reprezinta o colectie de **goal-uri**. task
- c. In Gradle, termenul de **task** reprezinta o unitate de executie, care poate fi executata folosind comanda "`gradle numele_pluginului:numele_taskului`".
- d. In Gradle, termenul de **goal** reprezinta o unitate de executie, care poate fi executata folosind comanda "`gradle numele_goalului`".
- e. In Gradle, termenul de **goal** reprezinta o unitate de executie, care poate fi executata folosind comanda "`gradle numele_pluginului:numele_goalului`".

Care din urmatoarele afirmatii legate de Build Tools sunt ADEVARATE?

- a. Un proiect NU poate contine atat un fisier **pom.xml** cat si un fisier **build.gradle**, deoarece nu putem genera decat un wrapper intr-un proiect.
- b. Un proiect NU poate contine atat un fisier **pom.xml** cat si un fisier **build.gradle**, deoarece acestea ar duce la conflicte in momentul build-urilor.
- c. Un proiect NU poate contine atat un fisier **pom.xml** cat si un fisier **build.gradle**, si poate fi build-uit atat cu Maven, cat si cu Gradle, deoarece chiar daca input-urile sunt distincte, folder-ul de output este acelasi.
- d. Un proiect poate contine atat un fisier **pom.xml** cat si un fisier **build.gradle**, deoarece fisierul **pom.xml** contine informatii legate doar de compilare, iar cel **build.gradle** informatii doar despre testare.
- e. Un proiect poate contine atat un fisier **pom.xml** cat si un fisier **build.gradle**, si poate fi build-uit atat cu Maven, cat si cu Gradle, deoarece atat input-urile, cat si output-urile tool-urilor sunt complet distincte.

Maven

pom.xml

. /maven complete

phase + goal

plugin

maven phase

src / main / java
cod

/ test / java
test

src / main / resources

Gradle

build.gradle

. /gradle vs build

task

plugin = task-in;
gradle task