

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

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

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**.
- ☒ 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 ADEVARATE?

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

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*".
- ☐ 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 **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*".

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.

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)
- ☐ c. Va executa doar faza **package** din "default lifecycle"
- ☐ d. Executa plugin-ul numit **package**
- ☐ e. Printeaza toate pachetele gasite in proiect

Care din următoarele afirmații legate de Build Tools sunt ADEVARATE?

- ☐ a. 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.
- ☒ 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.
- ☐ c. 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.
- ☐ d. Un proiect NU 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 chiar dacă input-urile sunt distincte, folder-ul de output este același.
- ☐ e. Un proiect NU poate conține atât un fișier **pom.xml** cât și un fișier **build.gradle**, deoarece acestea ar duce la conflicte în momentul build-urilor.

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

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



Care din urmatoarele afirmatii legate de Build Tools sunt FALSE?

CORECT

- ☐ 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.
- ☒ 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)

☒ b. Apeleaza un goal numit **package**

CORRECT

☒ c. Printeaza toate pachetele gasite in proiect

☒ d. Va executa task-ul **package** si toate task-urile de care acesta depinde

☒ e. Va executa doar faza **package** din "*default lifecycle*"

Care din urmatoarele afirmatii legate de Build Tools sunt FALSE?

CORRECT

- ☒ a. 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.
- ☒ b. 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.
- ☒ 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 NU poate contine atat un fisier **pom.xml** cat si un fisier **build.gradle**, deoarece acestea ar duce la conflicte in momentul build-urilor.
- ☐ 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.

Care din urmatoarele afirmatii legate de Build Tools sunt ADEVARATE?

CORRECT

- ☒ a. In Gradle, termenul de **task** reprezinta o unitate de executie, iar termenul de **plugin** reprezinta o colectie de **task-uri**.
- ☒ b. In Gradle, termenul de **task** reprezinta o unitate de executie, care poate fi executata folosind comanda "*gradle numele\_taskului*".
- ☐ c. In Gradle, termenul de **goal** reprezinta o unitate de executie, care poate fi executata folosind comanda "*gradle numele\_pluginului:numele\_goalului*".
- ☐ d. In Gradle, un plugin poate fi executat folosind sintaxa "*gradle numele\_pluginului*".
- ☐ e. In Gradle, termenul de **task** reprezinta o unitate de executie, care poate fi executata folosind comanda "*gradle numele\_pluginului:numele\_taskului*".

Care din urmatoarele afirmatii legate de Build Tools sunt FALSE?

CORRECT

- ☐ 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**.
- ☐ 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?

CORRECT

- ☐ 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*".
- ☐ 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 **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*".

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.
- ☐ b. Un artefact de Maven contine minim 3 elemente: groupId, artifactID si version.
- ☒ c. Spre deosebire de Maven, Gradle nu suporta dependente tranzitive.
- ☒ d. Comanda **mvn install** creeaza un jar, un artefact de maven si ii face deploy in repository-ul Maven Central.
- ☒ e. Gradle nu poate folosi ca dependente artefacte care au fost "build-uite" cu Maven.

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.
- ☐ b. Un artefact de Maven contine minim 3 elemente: groupId, artifactID si version.
- ☒ c. Spre deosebire de Maven, Gradle nu suporta dependente tranzitive.
- ☒ d. Comanda **mvn install** creeaza un jar, un artefact de maven si ii face deploy in repository-ul Maven Central.
- ☐ e. Gradle nu poate folosi ca dependente artefacte care au fost "build-uite" cu Maven.





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.
- ☐ 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.
- ☒ 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 NU poate contine atat un fisier **pom.xml** cat si un fisier **build.gradle**, deoarece nu putem genera decat un wrapper intr-un proiect.

Selectati notiunile care **exista** in contextul unui proiect de **Gradle**:

 a. build.gradle - scris in DSL (domain specific language) creat peste Groovy

 settings.build - fisier de configurare

 c. folderul "target" (= folderul destinatie)


 d. folderul "build" (= folderul destinatie)


 e. build.xml - fisier de configurare


Care din urmatoarele afirmatii legate de Build Tools sunt ADEVARATE?


- ☒ a. Spre deosebire de Maven, Gradle nu suporta dependente tranzitive.
- ☒ b. Gradle nu poate crea artefacte de maven, ci pentru a face acest lucru, are nevoie de un plugin.
- ☒ c. Gradle nu poate folosi ca dependente artefacte care au fost "build-uite" cu Maven.
- ☐ d. Daca avem **N** proiecte de Maven pe masina noastra locala, si fiecare dintre ele depind de aceeasi dependenta *junit:junit:4.3.12*, aceasta dependenta va fi downloadata o singura data din repository-ul Maven Central, si cache-uita in repository-ul local de Maven.
- ☐ e. Un artefact de Maven contine minim 3 elemente: *groupId*, *artifactID* si *version*.


Selectati notiunile care **exista** in contextul unui proiect de **Maven**:

 a. goal

 b. folderul "src" (folderul de surse)

 c. pom.maven - fisierul de configurare

 d. plugin

 e. task

Care din urmatoarele afirmatii legate de *structura de foldere standard*, sugerata de Maven, sunt ADEVARATE?

- ☒ a. Folderul **src/main/java** este directorul "radacina" al tuturor pachetelor de java, care contin codul functional al aplicatiei.
- ☒ b. Folderul **src** contine toate fisierele sursa, adica orice fisier java care face parte din aplicatie.
- ☒ c. Folderul **target** este folderul default de output al fazelor din *maven lifecycle*.
- ☒ d. Folderul **src/main/resources** ar trebui sa contina fisiere de resurse pe care codul functional le foloseste.
- ☒ e. Toate sunt false.

Care din urmatoarele afirmatii legate de Build Tools sunt FALSE?

- ☒ a. In Maven exista un singur fisier de configurare unde sunt trecute toate informatiile despre proiect, si anume **build.maven**
- ☐ b. In Maven exista un singur fisier de configurare unde sunt trecute toate informatiile despre proiect, si anume **pom.xml**
- ☒ c. Pentru Maven si Gradle "**convention over configuration**" inseamna ca modul de configurare este extrem de rigid, deoarece exista conventii pentru toate configurarile
- ☒ d. Toate sunt adevarate
- ☐ e. Pentru Maven si Gradle "**convention over configuration**" inseamna ca nu toti parametri trebuie setati si ca majoritatea setarilor au deja valori default

Care din urmatoarele afirmatii legate de *structura de foldere standard*, sugerata de Maven, sunt FALSE?



☒ a. Toate sunt adevarate



☐ b. Folderul **src/main/resources** ar trebui sa contina fisiere de resurse pe care codul functional le foloseste ADV

☐ c. Folderul **target** este folderul default de output al fazelor din maven lifecycle ADV

☐ d. Folderul **src/main/java** este directorul "radacina" al tuturor pachetelor de java a codului functional al aplicatiei ADV

☐ e. Folderul **src** contine toate fisierele sursa, adica orice fisier java care face parte din aplicatie ADV



Care din urmatoarele afirmatii legate de Build Tools sunt FALSE?

☐ a. Toate sunt adevarate

☒ b. In Gradle exista un singur fisier de configurare unde sunt trecute toate informatiile despre proiect, si anume **build.gradle**

☐ c. In Gradle exista un singur fisier de configurare unde sunt trecute toate informatiile despre proiect, si anume **settings.gradle**

☒ d. In Gradle exista un singur fisier de configurare unde sunt trecute toate informatiile despre proiect, si anume **build.xml**

☐ e. In Gradle exista un singur fisier de configurare unde sunt trecute toate informatiile despre proiect, si anume **gradle.properties**



Which of the following are true with respect to Maven repositories and dependency management?

☒ a. library developers deploy their artifacts to Maven repositories

☒ b. each user has a local Maven repository

☒ c. library users also download all transitive dependencies of their needed artifacts


☐ d. library users only download the dependencies they have explicitly specified

☐ e. library users always download their dependencies from the central Maven repository


Care din urmatoarele afirmatii legate de *structura de foldere standard*, sugerata de Maven, sunt ADEVARATE?


 a. Folderul **src/main/resources** nu poate contine fisiere java



 b. In gradle, structura de foldere a aplicatiei trebuie sa fie cu totul diferita fata de cea standard definita de maven

 c. Ajuta la standardizarea proiectelor si ofera astfel suport pentru automatizarea proceselor de build

 d. Folderul **src/main/resources** ar trebui sa contina fisiere de resurse pe care codul functional le foloseste

 e. Folderul **target** este folderul default de output al fazelor din *maven lifecycle*

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"
- ☐ e. Executa plugin-ul numit **build**