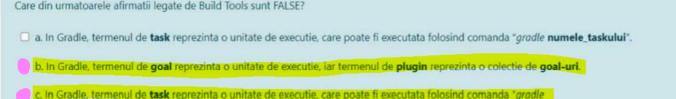
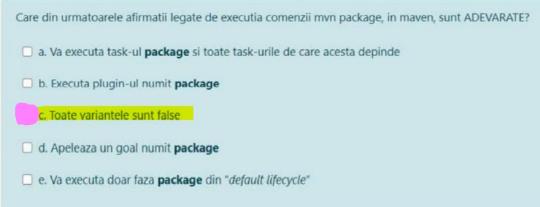
Care din urmatoarele afirmatii legate de executia comenzii mvn pockage, in maven, sunt ADEVARATE?		
•	a. Apeleaza un goal numit <b>package</b>	
	b. Va executa toate goal-urile care sunt asignate tuturor fazelor care preced faza de package (inclusiv a celor din faza package)	
	c. Printeaza toate pachetele gasite in proiect	
	d. Va executa task-ul package si toate task-urile de care acesta depinde	
	e. Va executa doar faza <b>package</b> din "defoult lifecycle"	



- d. In Gradie, termenul de **goal** reprezinta o unitate de executie, care poate fi executata folosind comanda "gradie numele\_goalului".
- e. In Gradle, termenul de goal reprezinta o unitate de executie, care poate fi executata folosind comanda "gradle.
   numele\_pluginului:numele\_goalului".

numele pluginului:numele taskului".



Care din urmatoarele afirmatii legate de Build Tools sunt ADEVARATE?	
<ul> <li>a. In Gradle, termenul de goal reprezinta o unitate de executie, care poate fi executata folosind comanda "gradle numele_pluginului:numele_goalului".</li> </ul>	
□ b. In Gradle, un plugin poate fi executat folosind sintaxa "gradle numele_pluginului"	
<ul> <li>c. In Gradle, termenul de task reprezinta o unitate de executie, care poate fi executata folosind comanda "gradle numele_pluginului:numele_taskului".</li> </ul>	
d. In Gradle, termenul de task reprezinta o unitate de executie, iar termenul de plugin reprezinta o colectie de task-	ır
e. In Gradle, termenul de task reprezinta o unitate de executie, care poate fi executata folosind comanda "gradle numele_taskului".	

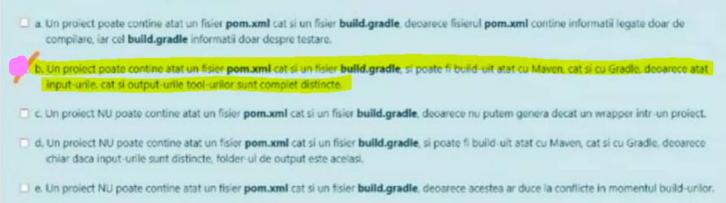
Care din urmatoarele afirmatii legate de Build Tools sunt ADEVARATE? a. Un proiect NU poate contine atat un fisier pom.xml cat si un fisier build.gradle, deoarece nu putem genera decat un wrapper intr-un project. ✓ b. Un proiect NU poate contine atat un fisier pom.xml cat si un fisier build.gradle, deoarece acestea ar duce la conflicte in

momentul build-urilor.

on

- 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 urmatoareie afirmatii iegate de executia comenzii mvn package, in maven, sunt ADEVAKATE?	
a. Va executa task-ul package si toate task-urile de care acesta depinde	
b. Va executa toate <b>goal-urile</b> care sunt asignate tuturor fazelor care preced faza de package (inclusiv a celor din faza pa	ickage)
□ c. Va executa doar faza <b>package</b> din "default lifecycle"	
d. Executa plugin-ul numit package	
e. Printeaza toate pachetele gasite in proiect	



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

numele pluginului:numele taskului".

## CORECT

- a. In Gradie, termenul de task reprezinta o unitate de executie, care poate fi executata folosind comanda "gradie numele\_taskului".
- b. In Gradie, termenul de goal reprezinta o unitate de executie, iar termenul de plugin reprezinta o colectie de goal-uri.
- In Gradie, termenul de task reprezinta o unitate de executie, care poate fi executata folosind comanda "gradie"
- d. In Gradie, termenul de goal reprezinta o unitate de executie, care poate fi executata folosind comanda "gradie numele, goalului".
  - 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)

Apeleaza un goal numit package

- d. Va executa task-ul package si toate task-urile de care acesta depinde
- ... Va executa doar faza package din "default lifecycle"

c. Printeaza toate pachetele gasite in project

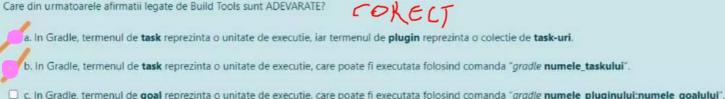
folder-ul de output este acelasi.

a. Un project 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 project NU poate contine atat un fisier pom.xml cat si un fisier build.gradle, deparece nu putem genera decat un wrapper intr-un project.

- d. Un project 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 project 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 project NU poate contine atat un fisier pom.xml cat si un fisier build.gradle, si poate fi build-uit atat cu Mayen, cat si cu Gradle, deoarece chiar daca input-urile sunt distincte,



- 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".

- 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".



- 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 "aradle numele pluginului"
- c. In Gradle, termenul de task reprezinta o unitate de executie, care poate fi executata folosind comanda "gradle numele pluginului:numele taskului".
- . 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".

- a. Gradle nu poate creea artefacte de maven, ci pentru a face acest lucru, are nevoie de un plugin, numit Maven.
- o. Un artefact de Maven contine minim 3 elemente: groupId, artifactID si version.
- c. Spre deosebire de Maven, Gradle nu suporta dependente trazitive.
- d. Comanda **mvn install** creeaza un jar, un artefact de maven si ii face deploy in repositoryul Maven Central.
- e. Gradle nu poate folosi ca dependente artefacte care au fost "build-uite" cu Maven.

- a. Gradle nu poate creea artefacte de maven, ci pentru a face acest lucru, are nevoie de un plugin, numit Maven.
- o. Un artefact de Maven contine minim 3 elemente: groupId, artifactID si version.
- c. Spre deosebire de Maven, Gradle nu suporta dependente trazitive.
- d. Comanda **mvn install** creeaza un jar, un artefact de maven si ii face deploy in repositoryul Maven Central.
- e. Gradle nu poate folosi ca dependente artefacte care au fost "build-uite" cu Maven.



d. Un proiect poate contine atat un fisier pom.xml cat si un fisier build.gradle, deoarece fisierul pom.xml contine informatii

e. Un project NU poate contine atat un fisier **pom.xml** cat si un fisier **build.gradle**, deoarece nu putem genera decat un

Gradle, deoarece chiar daca input-urile sunt distincte, folder-ul de output este acelasi.

legate doar de compilare, iar cel build.gradle informatii doar despre testare.

wrapper intr-un project.

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)

... folderul "build" (= folderul destinatie)

build.xml - fisier de configurare

- a. Spre deosebire de Maven, Gradle nu suporta dependente trazitive.
- b. Gradle nu poate creea artefacte de maven, ci pentru a face acest lucru, are nevoie de un plugin.
- Cradle nu poate folosi ca dependente artefacte care au fost "build-uite" cu Maven.
- 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)

pom.maven - fisierul de configurare

d. plugin

e. task

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

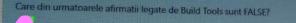
b. Folderul **src/main/java** este directorul "radacina" al tuturor pachetelor de java, care contin codul functional al aplicatiei.

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.

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

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

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

. 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?

- b. In gradle, structura de foldere a aplicatiei trebuie sa fie cu totul diferita fata de cea stardard definita de mayen
- c. Ajuta la standardizarea proiectelor si ofera astfel suport pentru automatizarea proceselor de build
- 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*

Folderul src/main/resources nu poate contine fisiere java

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 lifecylce"
- e. Executa plugin-ul numit build