# Preguntas Objetivo 6

Pregunta 1

Which two statements are correct about modules in Java? (Choose two.)

A. java.base exports all of the Java platforms core packages.

B. module-info.java can be placed in any folder inside module-path.

C. A module must be declared in module-info.java file.

D. module-info.java cannot be empty.

E. By default, modules can access each other as long as they run in the same folder.

Pregunta 2

Given this requirement:

Module vehicle depends on module part and makes its com.vehicle package available for all other modules.

Which module-info.java declaration meets the requirement?

A.



B.



C.



D.



Pregunta 3

Which command line runs the main class com.acme.Main from the module com.example?

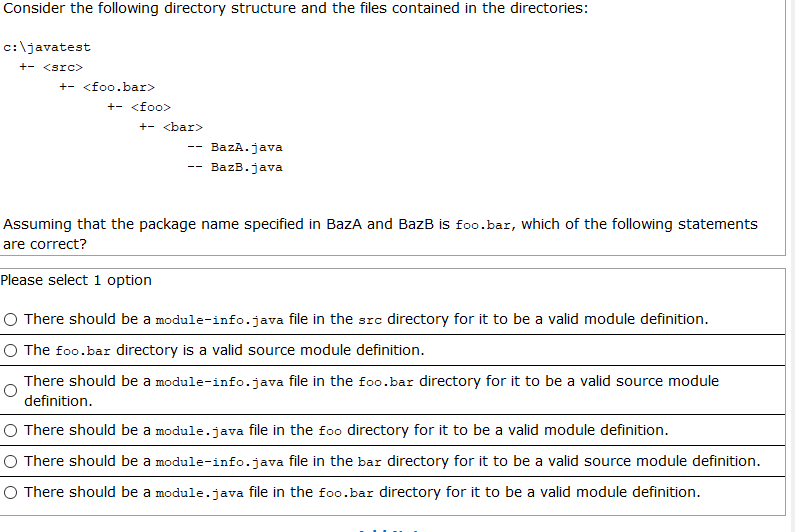
A. java --module-path mods com.example/com.acme.Main

B. java -classpath com.example.jar com.acme.Main

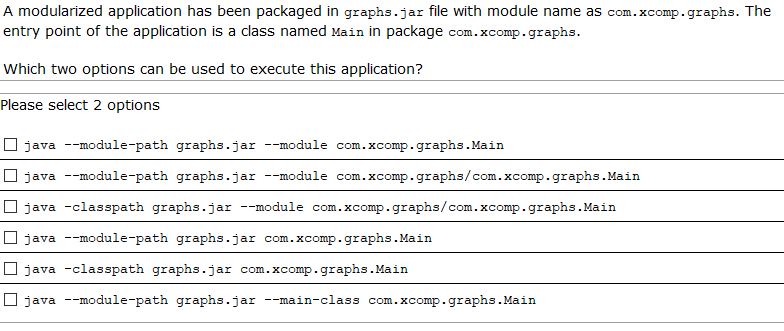
C. java --module-path mods -m com.example/com.acme.Main

D. java -classpath com.example.jar -m com.example/com.acme.Main

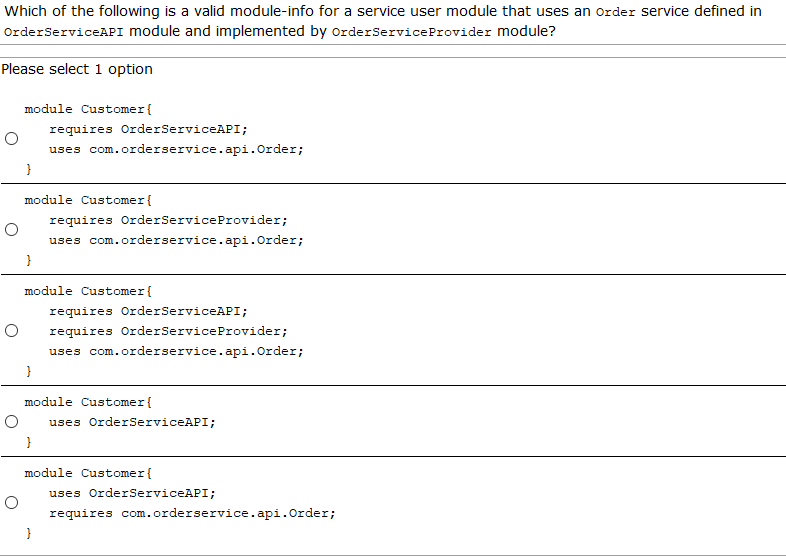
Pregunta 4



Pregunta 5



Pregunta 6



Pregunta 7

Which two statements are true about Java modules? (Choose two.)

A. Modular jars loaded from --module-path are automatic modules.

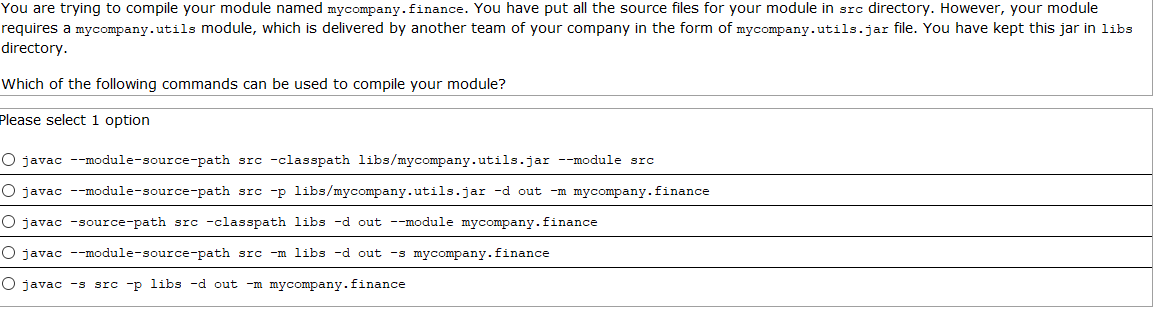
B. Any named module can directly access all classes in an automatic module.

C. Classes found in -classpath are part of an unnamed module.

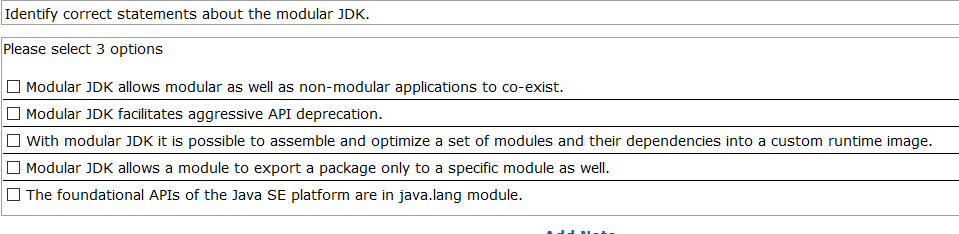
D. Modular jars loaded from -classpath are automatic modules.

E. If a package is defined in both the name module and the unnamed module, then the package in the unnamed module is ignored.

Pregunta 8



Pregunta 9



Pregunta 10

You are creating a com.sales module that makes its two classes - com.sales.service.Clients and com.sales.service.Sales - available to all other modules for use. Also, you need to use all the class packages exported from module testing.clients

Which of the following files correctly defines this com.sales module?

1. module com.sales{

exports com.sales.\*;

requires testing.clients;

}

1. module com.sales{

exports com.sales.service;

requires testing.clients;

}

1. module com.sales{

exports com.sales.service.\*;

requires testing.clients;

}

1. module com.sales{

exports com.sales.\*;

imports testing.clients;

}

1. module com.sales{

exports com.sales to testing.clients;

imports from testing.clients;

}

Pregunta 11

Which statements are true? (choose 2)

1. An automatic module exports all packages to named modules
2. An unnamed module exports no packages to named modules
3. An automatic module exports no packages to named modules
4. An unnamed module exports all packages to named modules
5. An automatic module exports only the specified packages to named modules

Pregunta 12

Your group has an existing application (reports.jar) that uses a library (analytics.jar) from another group in your company. Both - the application and the library - use a JDBC driver packaged in ojdbc8.jar.

Which of the following options describes the steps that will be required to modularize your application?

1. 1. Convert analytics.jar and ojdbc8.jar into automatic modules 2. Convert reports.jar into a named module. 3. Add requires clauses for analytics and ojdbc8 in reports.jar in its module-info.java.
2. 1. Modularize analytics.jar and ojdbc8.jar into modules by adding module-info.java to these jars. 2. Convert reports.jar into a named module. 3. Add requires clauses for all packages contained in analytics.jar and ojdbc8.jar that are directly referred to by classes in reports.jar in its module-info.java
3. 1. Convert reports.jar into a named module. 2. Add requires clauses for analytics and ojdbc8 modules in reports.jar in its module-info.java. 3. Use analytics.jar and ojdbc8.jar as unnamed modules.
4. 1. Convert ojdbc8.jar into automatic module. 2. Convert analytics.jar into a named module by adding module-info.java to it. In this module-info, export all packages that are used by reports.jar and add requires clauses for all packages of ojdbc8.jar that are used by analytics.jar. 3. Convert reports.jar into a named module. Add requires clause for analytics module in reports's module-info.java.