Java 9 migration

WEBVÁLTÓ

EXPERTISE WITH NO LIMITS

Topics

- Top-down migration
- Automatic module
- Bottom-up migration
- Split package
- Cyclic references

Top-down migration

- Begin migration with the application JARs
- Handle library JARs as unnamed or automatic modules
- Resolve cyclic references and split packages
- Use jdeps to check dependencies
- Easier to migrate
- Library JARs handled in a messy way

The Unnamed Module

- All types must be associated with a module in Java SE 9.
- A type is considered a member of the unnamed module if it is:
  - In a package not associated with any module
  - Loaded by the application
- Unnamed modules:
  - Read all other modules
  - Export all their packages
  - Cannot have any dependencies declared on them
  - Cannot be accessed by a named module
    - A named module is one with a `module-info.java` file.

Automatic Module

- Is a JAR file that does not have a module declaration and is placed on the module path
- Is a “real” module
- Requires no changes to someone else's JAR file
- Is given a name derived from the JAR file (either from its name or from metadata)
- Requires all other modules
- Can be required by other modules
- Exports all of its packages

Bottom-up migration

- Begin migration with library JARs
- Resolve cyclic dependencies and split packages
- Use jdeps to check dependencies AND generate module-info.java to libraries
- Harder to migrate
- Library JARs modularized somewhat optimized

Splitting a Java 8 Application into Modules

game.jar

- game.Soccer
- game.SoccerPlayer
- game.SoccerTeam
- game.SoccerEvent
- game.Kickoff
- game.Goal
- game.Pass
- game.Dribble
- game.Freekick
- game.Shoot
- game.Corner
- …

game module

- game.Soccer
- game.SoccerPlayer
- game.SoccerTeam
- game.SoccerEvent

event module

- game.Kickoff
- game.Goal
- game.Pass
- game.Dribble
- game.Freekick
- game.Shoot
- game.Corner
- …

Migration of Split Package JARs to Java SE 9

Java SE 8
League.jar

game.Factory
game.Util
game.Game
game.GameEv
ent
game.Player
game.Team

Java SE 8
No conflict

game.Soccer
game.SoccerEve
nt
game.SoccerPla
yer
game.SoccerTea
m

Java SE 8
Soccer.jar

Java SE 9
league
module

game.Factory
game.Util
game.Game
game.GameEv
ent
game.Player
game.Team

Java SE 9
Split packages

game.Soccer
game.SoccerEve
nt
game.SoccerPla
yer
game.SoccerTea
m

Java SE 9
soccer
module

Cyclic Dependencies

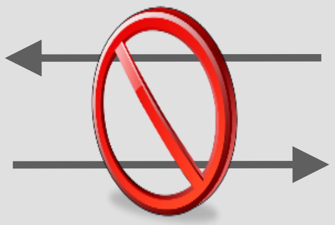
Cyclic module dependencies are not permitted in Java SE 9.

```
module league{
    requires
soccer;
}
```

```
open module soccer {
    exports soccer to
league;
    requires league;
}
```

Java SE 9 league module

- game.Factory
  game.Util
  game.Game
  game.GameEvent
  game.Player
  game.Team

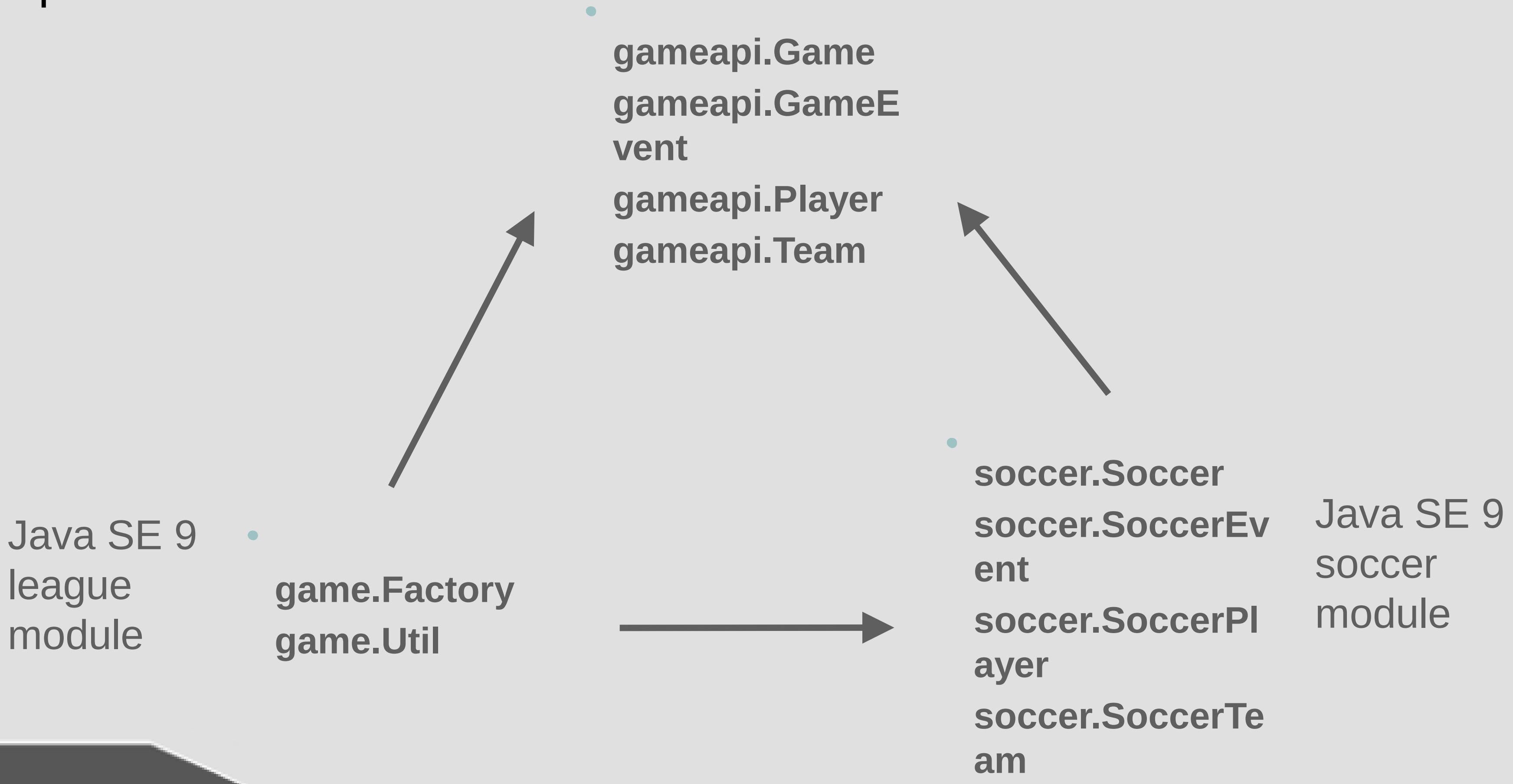


- soccer.Soccer
  soccer.SoccerEvent
  soccer.SoccerPlayer
  soccer.SoccerTeam

Java SE 9 soccer module

Addressing Cyclic Dependency

create a new module that both league and soccer are dependent on.



Java Cloud Service and PaaS

- Why Oracle Java Cloud Service?
- Cheap
- Easy to maintain
- Secure, isolated environment
- Optimal for testing migration