- są konwencje, ale łatwo zmieniać możemy

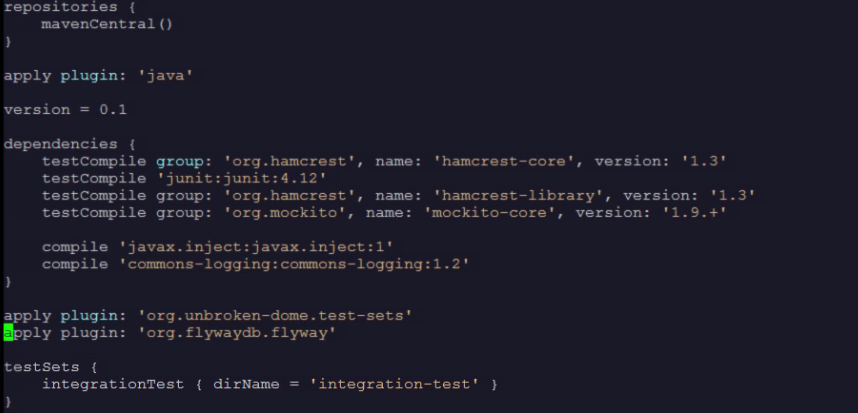
- piszemy DSL

- dependency ivy i maven saportowane

- wsparcie dla multiproject

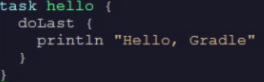
- dla budowania język, skupia sie co chcemy zrobic, nie jak

- określamy repo, dependencje, pluginy, testy



- sciągamy i unzip, dodaje bin do path

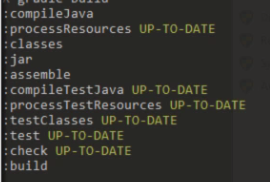
- build.gradle oprócz plugin i dependency ma też taski



- wywołanie gradle hello

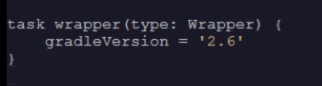
- gradle tasks – wypisuje taski

- gradle build robi po kolei :



- w build-> libs jest stworzony jar

- Task wraper



- wywołuje gradle wraper, powstaje gradlew i gradle.bat

- teraz gradlew build – sciąga gradle i buduje

- pod .gradle tworzy pliki, zbudowane pod tą wersją , mozna trzymać pod róznymi wersjami

Defining tasks, lifecycle

- task to kod, który grdle dla nas odpala , ma lifecycle, ma properties common i własne, ma akcje – czyli wykonywany kd, zdefiniowane first i last action, ma dependencjes, mogą od siebie zalezeć – pożądek

- wszystko w skrybcie obiektami, - jak groovy, głównym jest project, więc do niego się odnosimy , project to top level object

- jak zrobie project.task(„T1”) i zrobię gradle tasks – to wypisze mi taski , to samo można task(„T1”), lub task „T1” an nawet task T1

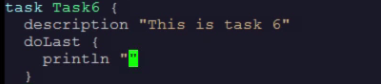
- mozna propercje np. T1.description = „aaa”

- wywołanie gradle T1, więc musze zdefiniować akcję

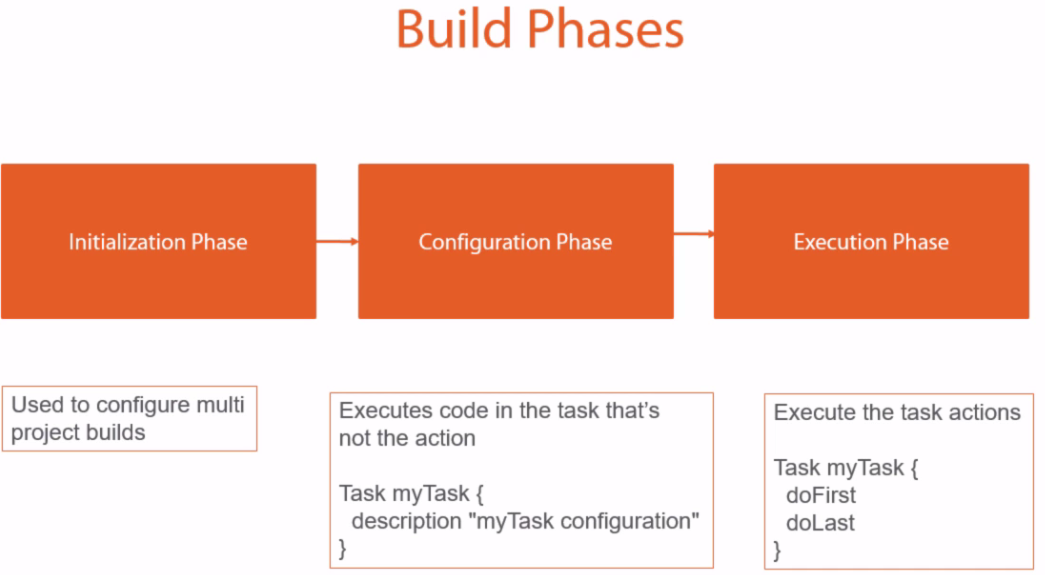
- T1.doLast {println „This is me”}, to samo co T1 << {println „This is me”}

- mozna od razu przy definicji : task T1 << {}, jak wywołam na tym << kolejne, to robi append

- mozna definicje na osobny sposob



- fazy :



- czyli mozna metodę do first zdefiniować

- mozna wiele duFirst i doLast

- jak dam T2.dependsOn T1 i wywołam gradle T2 – to mi najepier wykona T1 a potem T2

-- mozna to od razu w definicji

Task T3 {

dependsOn T2

}

- properties w gradle na kilka sposobów

-- local variable – w pliku def zmienna = 10, mozna jej uzyć we wszystkich taskach/akcjach , scope to build file obecny

-- project.ext.zmienna = 10, widoczne w całym projekcie, więc w tym pliku i pozostałych jak multiproject

-- w string „$zmienna”