## Integrations test

Da unit test tester hver klasse isoleret fra resten af systemet, tester man kun på at den enkelte klasse fungerer som forventet. Der kan dog forekomme nogle problemer, når de sættes sammen med andre klasser, hvis en klasse er udviklet af nogle og den anden klasse som den skal interagerer med har forstået at det fungere på en anden måde. Derfor bliver produktet integrationstestet så disse fejl bliver opdaget og kan blive rettet. Når man skal lave integrationstest skal man finde ud af hvordan klasserne er afhængige af hinanden og derfor hvor man har brug for at lave integrationstest henne. På Figur 1 kan man se et udsnit af Pristjek220’s dependency tree, hvor man kan se at produktet overholder DIP, da højniveau kalder lavniveau og ikke omvendt.



Figur 1: Udsnit af dependency tree for Pristjek220.

Der er forskellige måder at klare integrationstest på for at komme rundt om det hele. Integrationstest af Pristjek220 er gjort med strategien ”Bottom-up”, for let at kunne dække alle interfaces mellem klasserne. ”Bottom-up” kræver dog man skriver mange drivers til hvert lag når der testes, men der behøves dog ingen stubs når det testes med nogle få undtagelser som fx generering af tilfældige tal.