

Acme – part one

In deze opdracht gaan we een programma schrijven om de werkzaamheden die het bedrijf 'Acme' te simuleren.

Acme is een multinational bedrijf dat zich specialiseert in reparaties van voertuigen. Van elk voertuig (*Vehicle*) weten wij de dagwaarde (*price*) en het bouwjaar(*productionYear*). Alle voertuigen kunnen bewegen maar elke voertuig doet dat op een andere manier. Er zijn drie verschillende soorten voertuigen: *Car*, *Bike* en *Airplane*.

Acme heeft veel filialen (*Branch*) in verschillende landen. Van het filiaal weten we het adres (*address*) en het land(*country*) waarin deze zich bevindt. In deze filialen kunnen drie soorten medewerkers (*Employee*) werken. Mechanics (*Mechanic*), inkopers (*Purchaser*) en managers (*Manager*). Van elke medewerker weten we de naam(*name*), het adres(*address*) en het salaris(*salary*).

Klanten van Acme kunnen besluiten om hun voertuig aan één van de inkopers te verkopen. Inkopers hebben tot een bepaald bedrag (*budget*) vrij spel om voertuigen in te kopen. Als een voertuig duurder is dan het bedrag van deze inkoper, kan deze niet ingekocht worden. Als een voertuig binnen het budget van de inkoper valt, wordt deze altijd ingekocht. Er wordt bijgehouden welke voertuigen een inkoper heeft ingekocht. Het budget dat die inkoper nog kan spenderen wordt verminderd met de aanschafprijs. Uiteraard kan een inkoper ook de voertuigen die hij heeft gekocht verkopen. Zijn budget wordt dan opgehoogd. Er word wel eerst gecontroleerd of deze inkoper ook daadwerkelijk het voertuig gekocht heeft. Acme is natuurlijk geen filantropische instelling en verkoopt voertuigen voor 10% meer dan de inkoopsprijs.

Bij Acme werken ook managers. Een manager '*manages*' meerdere werknemers en uiteraard weet hij wie dat zijn. Een employee weet ook altijd wie zijn manager is. Verder kan een manager een employee *promoten()* of *demoten()*. Het salaris verandert dan met 5%. Ook hier word weer gecontroleerd of een manager daadwerkelijk de manager is van deze employee.

Een mechanic repareert (*repair()*) uiteraard voertuigen. Zodra een voertuig gerepareerd wordt, veranderd zijn status naar "Fixed" en wordt er klasse specifieke informatie geprint. bv: "*This car has been fixed. Replaced the carburetor*" of "*This airplane has been fixed. Replaced the pilot*".

Maak, voordat je deze simulator gaat programmeren, eerst een plan (UML-klassendiagram).