**Cohesie**

Cohesie wil zeggen dat je code schrijft die een duidelijke en functionele taak heeft, geen overbodige lange codes, maar alles word ingekort tot het zo overzichtelijk mogelijk is. Ook wil dit zeggen dat een stukje code (methode) een duidelijke opdracht heeft, dus dat je je niet afvraagt wat deze methode doet.

**Inkapseling**

Een goede inkapseling zorgt ervoor dat de code los van elkaar schrijft. Dit wil zeggen als je een klasse of methode gemakkelijk kunt aanpassen zonder dat je al te veel klassen of andere methodes moet aanpassen. Ook zorgt dit voor een goede afhankelijkheid van de code.

**Cohesie en inkapseling toepassen in vossen en konijnen**

In vossen en konijnen hebben we dit zo goed mogelijk aangepakt in zowel alle methodes en in alle klassen. Dit kun je bijvoorbeeld goed zien in de klasse PieChart3, en PieChartview. Hier zie je dat de methodes zo klein mogelijk zijn geschreven; Zo zijn meerdere functies in 1 regel verwerkt en hiermee voorkom je veel overbodige regels code. Wel word hierdoor de code iets moeilijker te lezen, maar je kunt de code hierdoor makkelijker hergebruiken, en extra dingen toevoegen. Dit geld voor zowel de extra views, als extra bereken methodes.