Raspodela odgovornosti

- Preduslov za postizanje opisanih vrlina jeste *dobro razdvajanje brige* (*separation of concerns*) i *raspodela odgovornosti* (*distribution of responsiblity*) po apstrakcijama, klasama, modulima
- Elemente softvera (podatke, funkcionalnosti) treba grupisati u klase ili module po principu *jake kohezije i slabe sprege* (*strong cohesion/loose coupling*):
  - elementi unutar iste klase ili modula treba da budu *jako i tesno međusobno povezani* (*kohezija, cohesion*)
  - elementi iz različitih klasa i modula treba da budu *slabo spregnuti* (*loose coupling*)
- Ponekad se ovo izražava i kao *princip jedinstvene odgovornosti klase* (*single responsibility principle*): klasa treba da ima ograničenu odgovornost, tako da postoji samo jedan razlog za njenu izmenu
- Ako se različite odgovornosti mogu nezavisno menjati, postoji ozbiljan razlog za razdvajanje tih odgovornosti u različite klase
- Klasa svakako ne treba da ima previše odgovornosti, posebno ako su one slabo povezane ili nepovezane - raspodela odgovornosti je jedan element *objektne dekompozicije*

Raspodela odgovornosti

- Na primer, logično je da u odgovornosti klase za lika u arkadnoj igrici bude izračunavanje njegovih sledećih koordinata tokom kretanja po polju, kao i iscrtavanje tog karaktera na polju; međutim, ako je neka od ove dve odgovornosti suviše složena (npr. zavisi od složenih stanja, struktura ili algoritama), ili se mogu nezavisno kombinovati, treba razmisliti o njihovom razdvajanju u različite klase: lik, kao apstrakcija, može imati, kao deo svoje implementacije, objekte klasa zaduženih za izračunavanje pomeraja i za iscrtavanje
- Objekat klase *Character delegira* (*delegates*) odgovornost za ove obaveze tako što poziva odgovarajuće operacije tih pridruženih objekata kada mu je potrebno izračunavanje pomeraja ili iscrtavanje
- Na taj način, način izračunavanja pomeraja i/ili iscrtavanja može se nezavisno menjati specijalizacijama ovih klasa, jer navedeni pozivi mogu biti polimorfni
- Naravno, sve ovo ima smisla samo ako su ove promene predvidive: nema potrebe raditi dodatan posao i činiti softver složenijim ako takve promene nisu očekivane. Čak i ako nisu predviđene, ukoliko je ispoštovana enkapsulacija apstrakcije lika, softver se može *refaktorisati* (*refactoring*) sa ciljem ovakve dekompozicije