1) ##Lines of Code
Gender: 5
Movie: 30
MovieManger: 117
Performer: 25

== Unterschiede, weil die Klassen einfach unterschiedlich groß sind.

##Anzahl der Quelltextzeilen in allen Operationen einer Klasse
##Lines of Code - Average Numer of Fields per Type - 1 (import) - 2 (class ... & })
Gender: 0
Movie: 23
MovieManager: 113
Performer: 18

== Gender hat keine, weil es nur ein Enum ist, der Rest hat ein paar Setter/Getter. Bei MovieManger kommt noch die recht lange Main-Methode dazu, daher sticht ihr Wert heraus.

##Average Lines of Code per Method
Gender: 0.00
Movie: 2.87
MovieManger: 13.62
Performer: 1.80

== Gender hat keine Methoden, da es nur ein Enum ist, bei Movie und Performer sind hauptsächlich die Setter/Getter dominierend, bei MovieManager kommt die recht lange Main-Methode dazu und hebt den Schnitt deutlich

##Average Number of Parameters
Gender: ----
Movie: 0.42
MovieManger: 1.42
Performer: 0.44

== Was soll man dazu sagen? Bei Movie und Performer dominieren die Setter/Getter; die Setter haben je ein, die Getter kein Argument, daher ein Schnitt um 0.5.

##Average Number of Fields per Type
Gender: 2.00
Movie: 4.00
MovieManger: 1.00
Performer: 4.00

== Das ist halt die Anzahl der Variablen in jeder Klasse. Unterschiede kommen daher, dass die Klassen eben Unterschiedlich viele Variablen beinhalten.

##Average Number of Methods per Type
Gender: 0.00
Movie: 7.00
MovieManger: 7.00
Performer: 9.00

== Anzahl der Variablen * 2 + n. Bei Movie muss man dabei die Klassenvariable nextNumber raus lassen, da es für diese keine Setter/Getter gibt.

2) +Die Verwendung von Metriken im Code führt vor allem dazu, dass sich Programmierer mehr Gedanken darüber machen, welche Eigenschaften ihr Code hat.
- Vorteil: Programmierer hinterfragen Code und können ungewöhnliche Eigenschaften erkennen.
- Nachteil: Programmierer hinterfragen Code und versuchen dafür zu sorgen, dass er schöne Metriken ergibt.

+ Code Auditing wird vereinfacht
- Vorteil: Revisor kann sich auf auffällige Stellen konzentrieren
- Nachteil Revisor vernachlässigt möglicherweise unauffälige, aber dennoch falsche Codestellen.