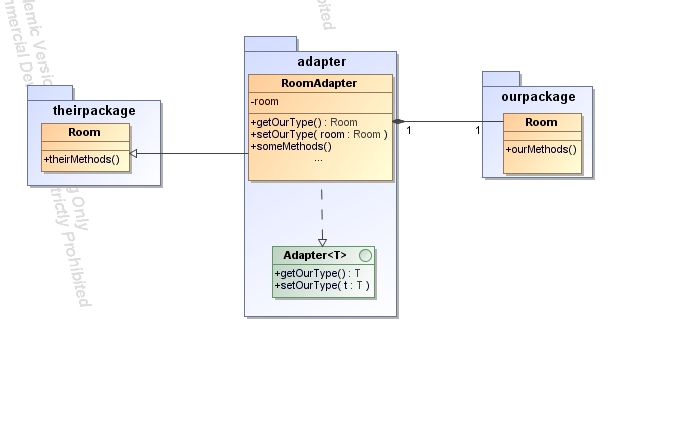
Designentscheidung Adapter für Programmteile von Team F

Wie schon in der Vorlesung gezeigt wurde ist Integration mittels Adapter eine eigentlich einfache Sache. Mittels Schnittstellen, definiert durch Interfaces, ist dieser Vorgang sehr schnell abgeschlossen. Da Team F leider solche Schnittstellen nur für ihre GUI hatte aber leider nicht für ihre Domänenklassen, mussten wir improvisieren. Nach einer Recherche haben wir festgestellt, dass auch eine Adaption mittels Vererbung möglich ist. Somit haben wir uns entschieden diese Variante anzuwenden.

Um zu ermöglichen eine einfache und doch effiziente Integration zu gewährleisten, haben wir uns das Adapterpattern genauer analysiert und bei uns in folgender Form angewendet:



Wie man in der Grafik erkennen kann, haben wir jede zu adaptierende Klasse als Super-Klasse unserer Adapter. Diese Adapter implementieren ein Interface „Adapter<T>“, welches den Typ unserer Domänenobjekte spezifizier, zusätzlich hat dieses je eine Methode definiert in denen wir unser Domänenobjekt setzen oder holen können.

Die Managerklassen von Team F waren etwas einfacher zu adaptieren. Wir haben aus diesen Interfaces generiert und diese Interfaces implementiert. Die Managerfactory, welche die verschiedenen Manager der Pakete erzeugt, musste von uns umgeschrieben werden, damit diese unseren adaptierten Manager generiert.