

ПРИМЕНЕНИЕ ПАТТЕРНА «MODEL-VIEW-PRESENTER» ПРИ ПРОЕКТИРОВАНИИ ИНТЕЛЛЕКТУАЛЬНОЙ ТРАНСПОРТНОЙ ГЕОИНФОРМАЦИОННОЙ СИСТЕМЫ

Михеев С.В., Пупынин К.В.

Самарский национальный исследовательский университет имени академика С.П. Королева, ф-т информатики, каф. информационных систем и технологий,
Россия, 443086, г. Самара, Московское шоссе, 34,
E-mail Mikheevati@its-spc.ru

При разработке плагина для интеллектуальной транспортной геоинформационной системы «ITSGIS» необходимо решить, как будет строиться пользовательский интерфейс. При этом возникает проблема отделения бизнес-логики приложения от отображения данных. Кроме того, работая с такой большой и сложноорганизованной системой ITSGIS, хотелось бы иметь возможность модульного тестирования разрабатываемых плагинов, а также уменьшить трудозатраты на его разработку, используя готовые унифицированные решения.

ITSGIS построена на принципах объектно-ориентированного проектирования с использованием паттернов. В ходе проектирования системы дислокации объектов транспортной инфраструктуры на автозаправочных станциях выявлены объекты предметной области, определены классы и интерфейсы классов, определена иерархия наследования, установлен регламент отношений между классами.

В основе проектирования плагинов в ITSGIS лежат паттерны, состоящие из классов и взаимодействующих объектов, с помощью которых решаются задачи проектирования: порождающие, структурные и паттерны поведения.

- ✓ порождающие паттерны связаны с процессом создания объектов, частично делегируют ответственность за создание объектов своим подклассам (например, дислокация дорожных знаков и светофоров);
- ✓ структурные паттерны имеют отношение к композиции объектов и классов, используют наследование для составления классов или описывают способы сборки объектов из частей (например, участок улично-дорожной сети и дислоцированные на нем дорожные знаки; или отчеты по характеристикам улично-дорожной сети, ДТП и т.п.);
- ✓ паттерны поведения характеризуют взаимодействие классов или объектов при использовании механизмов наследования для описания алгоритмов и потока управления; на уровне объектов устанавливают регламент отношений в некоторой группе совместно функционирующих объектов, выполняющих задачу, которая ни одному отдельному объекту не под силу.

Использование паттерна «MVP» позволило отделить бизнес-логику от логики пользовательского интерфейса, и сделать код удобочитаемым, что упрощает его автоматизированное модульное тестирование. Для реализации данного подхода были созданы: модель, хранящая в себе бизнес-логику; представление, отвечающее за отображение данных и обращающееся к представителю за обновлениями; представитель, обеспечивающий взаимодействие модели и представления.