

В качестве основного примера было взято приложение для имитации движения лифта в многоэтажном доме. Первоначальная версия включала в себя некоторые большие недостатки, как применение антипаттернов проектирования, как «Макаронный код», «Жесткое связывание» и так далее. Моей задачей было то, чтобы устранить данные недостатки в проектировании программного обеспечения. И в качестве основного инструмента, я использовал контейнер Spring.

Для начала были созданы настройки с помощью файла XML и Java. В данные файлы были вынесены те объекты, создание которых является затратным, как пул потоков. Также в них были определены источник значений из файла, поиск компонентов внутри программы. После этого был модифицирован класс-строитель *ElevatorTaskBuilder*, который отвечает за создание специфичных компонентов, как лифта, контроллера лифта, этажей и пассажиров. Ключевой особенностью этого момента является то, что здесь применяется внедрение зависимостей через параметры методов с помощью аннотации *@Autowired*. Тем самым удается избежать повторного или сложного создания этих фабричных методов.

Далее были модифицированы отдельные компоненты, как элементы журналирования событий, валидация компонентов, загрузчики параметров. Здесь в этих компонентах мы убрали ненужную логику работы и поместили аннотацией *@Component*, чтобы контейнер находил данные классы и помещал их в свой контейнер.

И, наконец, был также модифицирован запуск приложения, в котором инициализируется загрузка контейнера и в данный контейнер, после его инициализации, попадают вышеописанные объекты, с которыми в дальнейшем работает наше приложение. И в результате всей работы, код стал читабельным и его объём сократился примерно в два раза.

Н.С. Исаков (УО «ГГУ имени Ф. Скорины», Гомель)

Науч. рук. **П.Л. Чечет**, канд. техн. наук, доцент

РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ДВИЖЕНИЯ ЛИФТА В МНОГОЭТАЖНОМ ДОМЕ

В наше время применение имитационных моделей и систем моделирования является обыденным делом в сферах инженерной промышленности. Основная причина использования данного моделирования заключается в том, что моделирование на реальном объекте является очень дорогостоящим и при этом существуют всевозможные риски при её построении и эксплуатации.

В качестве основной задачи существует задача по разработке имитационной модели движения лифта в многоэтажном доме. В данную модель входит моделирование не только одного лифта, но и отдельных компонентов, как этажей, дома и пассажиров, которые будут участвовать в процессе имитации движения.

На рынке ПО существуют всевозможные решения в сфере имитационного моделирования, такие как Arena, GPSS World, Simulink и т. д. В каждой из систем существуют определенные преимущества и недостатки, как наличие дополнительных инструментов, способы визуализации моделирования в целом и т. д.

Было принято решение построить собственную имитационную модель на базе Java Virtual Machine с применением языка программирования Java. При использовании данного языка была применена концепция многопоточности, чтобы каждое выполняемое событие определяло в виртуальной машине, как отдельный выполняемый поток. Таким образом, мы добьемся большей гибкости при реализации данной модели. Также в качестве вспомогательной библиотеки был взят IoC-контейнер Spring, который помещает созданные компоненты имитационной модели в свой контейнер и данными компонентами можно пользоваться в течение жизненного цикла приложения. Этот прием избавит нас от многократного создания похожих компонентов и тем самым увеличится производительность приложения.

Данные приложения вводятся в конфигурационный файл до его запуска в JVM, и после этого начинается запуск приложения и вывод результата моделирования. Соответственно после результата моделирования, запускается процесс валидации компонентов, который проверяет состояние имитационной модели и дает результат валидности. Для вывода информации используется библиотека журналирования Log4J. И в качестве вспомогательного компонента для журналирования используется отдельный класс AppLogger, который выводит результат имитационной модели в консоль и в отдельный файл.

О.И. Камейша (УО «ГГУ имени Ф. Скорины», Гомель)

Науч. рук. **П.Л. Чечет**, канд. техн. наук, доцент

СТАНДАРТ УПРАВЛЕНИЯ ЗВУКОВЫМИ УСТРОЙСТВАМИ MIDI

Musical Instrument Digital Interface – цифровой интерфейс музыкальных инструментов. Создан в 1983 году ведущими производителями электронных музыкальных инструментов – Yamaha, Roland, Korg,