

Для выполнения поставленной задачи были выбраны технологии от корпорации Microsoft, а именно .NET Framework, язык программирования C#, ASP.NET, WebAPI, WPF, WCF, AngularJS, Xamarin, MS SQL Server, для написания универсального интерфейса используется WebAPI 2 в виде удалённого REST сервиса. Для десктопного приложения на Windows использована система для разработки клиентских приложений WPF. Для создания веб-клиента использована технология ASP.NET, а для облегчённой версии этого клиента построено одностраничное приложение SPA с использованием фреймворка AngularJS. Для создания клиентских приложений на мобильных платформах используется технология Xamarin. Для роли СУБД выбран MS SQL Server 2012.

Основным плюсом разработанного программного интерфейса является его универсальность, т. е. как видно, одним и тем же сервисом будут пользоваться приложения для совершенно разных платформ, и всем им этот разработанный сервис будет предоставлять удобный интерфейс. Другим явным плюсом является использование актуальных технологий, как для серверного приложения, так и для клиентских приложений. Так же в разрабатываемых средствах был сделан упор на модульность компонентов. Например, т. к. у каждого аукционного дома будут свои бизнес данные в отдельных базах данных, их можно будет удобно заменять в необходимых случаях. Данные же пользователей, которые используют разрабатываемые приложения, будут и вовсе храниться отдельно от бизнес-данных, что повышает безопасность.

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

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

ПРИМЕНЕНИЕ ФРЕЙМВОРКА SPRING В КАЧЕСТВЕ ИНСТРУМЕНТА ДЛЯ РЕФАКТОРИНГА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Рефакторинг – это контролируемый процесс улучшения кода, без написания новой функциональности. Результатом рефакторинга является чистый код и простой дизайн приложения.

Spring Framework (или коротко *Spring*) – универсальный фреймворк с открытым исходным кодом для Java-платформы. Данный фреймворк применяет концепцию Inversion of Control (IoC) и приемы аспектно-ориентированного программирования. При реализации IoC реализуется шаблон Dependency Injection. Spring содержит в себе так называемый IoC-контейнер, который хранит

В качестве основного примера было взято приложение для имитации движения лифта в многоэтажном доме. Первоначальная версия включала в себя некоторые большие недостатки, как применение анти-паттернов 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.