

DOI: 10.24411/1993-8314-2019-10044

*Н. Н. Прокимов, канд. техн. наук, доцент Московского финансово-промышленного университета «Синергия»,
prokimnovnn@mail.ru*

Концепции построения и принципы реализации приложения для автоматизации непрофильных функций преподавателя

Описаны технология, назначение и конструктивные особенности программного инструментария для автоматизированной поддержки выполнения рутинных процедур, составляющих неотъемлемую часть обязанностей преподавателя современной высшей школы. Предлагаемый подход и реализованный образец построен на основе принципов, предложенных автором ранее, и представляет собой их развитие и расширение. В качестве самостоятельной задачи рассмотрены концепции внутренней организации приложения, носящие в значительной степени универсальный характер. Приводятся факты и выводы, сделанные на основе практического опыта разработки и сопровождения, что может представлять интерес для разработчиков приложений подобного типа.

Ключевые слова: образование, учебный процесс, вуз, преподаватель, автоматизация, тестирование, проверка знаний, системный подход

Введение

Анализируя текущее состояние и тенденции современного высшего образования, можно отметить целый ряд его особенностей, существенно отличающих его нынешний облик от того, что было для него характерно в еще недавнем прошлом. В число таковых входят, в частности, массовый характер и широкое внедрение машинных средств и методов [3, 4], что преобразует привычную роль преподавателя и оказывает влияние на характер его работы. Одно из следствий происшедших и продолжающихся происходить перемен – рост доли учетно-административной нагрузки и, соответственно, временных издержек.

В качестве одной из возможных мер смягчения негативного эффекта от возросшего

объема непрофильных затрат автором ранее предлагался подход [2], позволяющий сократить затраты на выполнение рутинных операций и, таким образом, высвободить время на решение главных задач, иными словами, методической и аудиторной работы. За время, прошедшее с момента публикации упомянутой работы, тенденция усиливалась, и это стимулировало наращивание возможностей первоначальной версии.

С другой стороны, стремление адаптировать базовую технологию к происходящим переменам привело к естественным и ожидаемым последствиям в виде заметного роста объема программного кода, что, в свою очередь, внесло определенную дезорганизацию во внутреннюю структуру приложения, отрицательно сказалось на управляемости и увеличило временные затраты на внесение по-