

устойчив, то все начальные точки свободного корневого годографа $p_k(s)$, за исключением нулевой, располагаются в левой полуплоскости корней.

Следовательно, для настройки полинома (1) с целью обеспечения заданного качества достаточно найти в расширении (2) полинома (1) устойчивый полином степени $k = l$ ближайшей к n и настроить последовательно каждый коэффициент a_j полинома (1) в интервале $l < j \leq n$ посредством настройки свободного члена a_k соответствующего k -го полинома расширения (2), приняв $k = j$. При этом для каждого полинома расширения степени k решается задача устойчивости относительно соответствующей границы $\beta_k = \beta_j$

Литература:

1. Несенчук, А.А. Корневой метод синтеза устойчивых полиномов путем настройки всех коэффициентов / А.А. Несенчук // Автоматика и телемеханика. 2010. № 8. С. 13–24.

УДК 681.142.2

Решение проблемы слабого связывания компонентов в программной среде

Павлунин В. М., Скудняков Ю.А.

Белорусский национальный технический университет

Возрастающая сложность современных программных продуктов требует четкой организации структуры кода приложения, а именно разделение его на переиспользуемые компоненты. При этом между компонентами возникают зависимости, в которых один класс предоставляет услуги, а другой – их потребляет. Чем больше один компонент зависит от другого, тем сложнее в будущем поддерживать их работу, в случае, когда нужно внести изменения в работу зависимого компонента.

Для решения этой проблемы используется такой паттерн проектирования, как внедрение зависимостей. Суть его состоит в том, что зависимые объекты не создаются напрямую классами, их использующими, а достаются из стороннего источника, так называемого контейнера внедрения зависимостей (DI контейнеры). Это сторонние библиотеки, которые предоставляют API для внедрения зависимостей. Для настройки DI контейнеров используются основные три подхода: конфигурационные XML файлы (с расширением `.config`), регистрация с использованием кода и регистрация с использованием конвенций.

Вместе со взятием на себя роли по созданию зависимых компонентов, контейнеры также управляют их жизненным циклом, тем самым устраняя проблемы с утечками памяти, и добиваясь максимально

возможной производительности приложения. К основным преимуществам использования внедрения зависимостей относятся: уменьшается связность классов; увеличивается переиспользование кода; улучшается поддерживаемость кода; улучшается тестируемость приложения. Исследования по влиянию паттерна внедрения зависимостей на производительность и тестируемость приложения производились на платформе .NET с использованием программного языка C# и DI контейнера Autofac. При всех очевидных достоинствах использования контейнеров внедрения зависимостей существуют некоторые недостатки: необходимость изучения паттерна “внедрение зависимостей”; дополнительная нагрузка на производительность системы, связанная с использованием дополнительных сторонних библиотек; неправильное использование принципов обратимости контроля может принести больше вреда, чем пользы вашему приложению.

УДК 681.3

Иерархическая структура представления знаний студента в обучающей системе на основе нечетких множеств

Попова Ю.Б., Бураковский А.И.

Белорусский национальный технический университет

Знания студента по некоторому предмету можно представить в виде иерархии данных, имеющих алгоритм (механизм) интерпретирования. Каждый элемент такой иерархии (знания конкретный терминов, определений, формул и т.д.) с точки зрения обучающей системы является концептом – структурной единицей представляемых данных.

В таком случае проектируемые знания, определенные программой курса, можно представить как совокупность всех концептов для лекций, практических занятий, семинарских, тестов и прочее. Для определения степени принадлежности знаний студента в конкретном концепте предлагается рассматривать этот концепт как нечеткое подмножество множества знаний преподавателя. Такой подход позволяет выполнять операции без потерь точности, в отличие от привычного подхода на основе рейтингов/оценок. Это достигается за счет использования характеристических функций нечетких множеств, а также операций над нечеткими множествами. Для получения значений на нижних уровнях иерархии используется среднеарифметическое значение функций принадлежности:

$$\mu_{\text{лек } k} = \frac{1}{n} \sum \omega_{\text{лек } i} \mu_{\text{лек } i}. \quad (1)$$