

4) Обеспечить для администратора возможность оперативного доступа в любое время к необходимой информации программного средства.

Программное средство организации тренинга по курсу «Основы алгоритмизации и программирования» предусматривает выполнение основных функций:

- 1) просмотр всех доступных тестов;
- 2) просмотр теоретических материалов по выбранному тесту – перед каждым тестом студенту предлагается изучить теоретический материал, разобрать примеры по теме теста;
- 3) прохождение теста студентом – после изучения теоретического материала и разбора примеров студент имеет возможность пройти выбранный тест;
- 4) просмотр результатов тестирования с подробным объяснением ответов на задания теста;
- 5) создание нового теста преподавателем (или администратором) – возможность добавления нового теста, который включает в себя: теоретический материал по теме создаваемого теста; примеры (если допускается); тест с вариантами ответов и указанием правильного ответа; пояснение правильного ответа к каждому заданию теста;
- 6) редактирование и удаление теста преподавателем (или администратором);
- 7) просмотр информации о пройденных тестах и полученной оценке за их прохождение.

Поддержка работоспособности программного средства под управлением 32- и 64-битных операционных систем Windows и MacOS достигается выбором платформы и языка написания программного средства. В данном случае программное средство реализовано на языке Java под управлением кроссплатформенной среды разработки Java Runtime Environment. В качестве сервера к программному средству выступает отдельно работающий веб-сервер Apache Tomcat, принимающий и посылающий запросы в формате HTTP.

Таким образом, разработанное программное средство организации тренинга по курсу «Основы алгоритмизации и программирования» упрощает процесс тестирования знаний студентов, позволяет студентам совершенствовать уже полученные знания и легко изучать новый материал. Программное средство может быть использовано студентами всех форм обучения, в том числе студентами дистанционной формы.

Список использованных источников:

1. Хорстманн, К. С., Корнелл, Г. Java 2. Библиотека профессионала, том 1. Основы / К.С. Хорстманн, Г. Корнелл. – М.: Вильямс, 2007. – 896 с.
2. Глухова, Л. А. Основы алгоритмизации и программирования : лаборатор. практикум для студентов специальности I-40 01 01 «Программное обеспечение информ. технологий» днев. формы обучения : в 4 ч. Ч. 2 / Л. А. Глухова, Е. П. Фадеева, Е.Е.. Фадеева. – Мн. : БГУИР, 2005. – 52 с.

ЯЗЫК ПРОГРАММИРОВАНИЯ ДЛЯ ОПИСАНИЯ МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ НА ОСНОВЕ ПРИНЦИПОВ DOMAIN- DRIVEN DESIGN

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Костевич А.А.

Блинов И.Н. – кандидат физ.-мат. наук

Любое программное обеспечение связано с той или иной предметной областью. Построение модели предметной области, ее отображение в архитектуре системы и программная реализация являются одними из наиболее сложных и трудоемких задач проектирования. Создание языка программирования для описания модели предметной области на основе принципов Domain-Driven Design, позволило бы существенно упростить процесс моделирования.

Проектирование любой программной системы включает в себя создание модели предметной области, в которой будет работать пользователь. Возрастающая естественная сложность моделируемых предметных областей и, соответственно, моделей и архитектур систем обусловили появление нового подхода к проектированию – Domain-Driven Design (DDD).

DDD представляет собой набор приемов, подходов и принципов проектирования для ускорения разработки ПО для сложных областей деятельности, основывающихся на двух основных тезисах:

В большинстве программных проектах основное внимание должно быть сосредоточено на логической структуре предметной области и взаимосвязях с ней.

Архитектура сложного программного обеспечения должна основываться на модели предметной области.

Модель, согласно DDD, представляет собой «строгую организованную выборку знаний», представляющую собой приблизительное отображение реальности, удобную для ее программной

реализации. Модель позволяет выразить сложность процессов и богатство понятий той или иной предметной области в конструкциях языка программирования.

По словам Э. Эванса, автора DDD, “нужны четкие методы исследования моделей предметных областей и выражения их в работающих прикладных программах”. Современные высокоуровневые языки программирования общего назначения (C#, F#, Java, Scala, Swift) предоставляют богатый набор возможностей для выражения моделей различных предметных областей прикладной систем в коде программы, однако требуют для этого дополнительных трудозатрат и строгости на этапе проектирования.

Одним из способов увеличить производительность труда разработчиков и предоставить новые возможности по исследования предметных областей является создание нового языка программирования для описания модели предметной области, реализующего в себе основные принципы DDD: предметно-ориентированного языка программирования (domain-driven programming language).

Данное исследование нацелено на проектирование предметно-ориентированного языка программирования на основании объектно-ориентированного парадигмы и принципов Domain-Driven Design. На данном этапе, основными задачами проектирования предметно-ориентированного языка программирования являются:

Реализовать концепции объектно-ориентированной парадигмы программирования.

Предоставить типы для основных элементов модели предметной области: объектов-значений, сущностей и корней агрегации. Поведение и характеристики каждого элемента модели должны предоставляться соответствующим типом.

Предоставить специализированный синтаксис для описания служб и фабрик модели предметной области.

Предоставить возможность идентифицировать ограниченные контексты модели предметной области.

Предоставить возможность разбиения модели предметной области на модули.

Для интеграции с существующими инструментами проектирования и разработки исследуется компиляция исходного кода в Common Intermediate Language для платформы .NET или Java bytecode для платформы Java.

Список использованных источников:

1. V. Vernon Implementing Domain-Driven Design / V. Vernon – Addison-Wesley, 2013 – 612 p.
2. Эванс, Э. Предметно-ориентированное проектирование (DDD). Структуризация сложных программных систем / Э. Эванс – Вильямс, 2011. – 448с.
3. V. Vernon Effective Aggregate Design [Electronic resource] / V. Vernon – Mode of access: http://dddcommunity.org/library/vernon_2011

ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА СИГНАЛА СПЕКТРА, ПРИМЕНЯЕМАЯ В СИСТЕМЕ ПОДДЕРЖКИ ЛАЗЕРНОЙ ЭКСПРЕССНОЙ ЭКСПЕРТИЗЫ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Лазарь Д. В.

Рудикова Л. В. – д-р. техн. наук, профессор

В настоящее время немаловажное значение в научных исследованиях, связанных с экспрессной лазерной экспертизой, играет обработка полученных данных снятых спектров за короткое время, их накопление в базе данных и дальнейшее использование на различных этапах лазерной экспрессной экспертизы. Хранение информации в базе данных подразумевает определенный формат данных.

Предварительная обработка спектра необходима для выделения полезной информации из сигнала спектра для последующего приведения к определенному формату и сохранении в базу данных. Под полезной информацией подразумеваются пики, их интенсивность и площадь.

Алгоритм поиска пиков основан на методе деконволюции сигналов и осуществляется в несколько этапов.

1. Вычисление фона сигнала. Результатом является информация, содержащая пики и фон. Данный метод основан на SNIP-алгоритме (Sensitive Nonlinear Iterative Peak):

$$V_p \leftarrow \min \left\{ V_{p-1} \left[\frac{I_{p-1} (+p) + V_{p-1} (-p)}{2} \right] \right\}$$

где P – номер итерации.

2. Вычисление сглаженного сигнала с помощью алгоритма, который базируется на дискретной цепи Маркова, которая имеет простое инвариантное распределение: