

УДК 519.683.8

Обучающая система работы с контейнерными классами

А.И. Ольшевский

Донецкий национальный технический университет
a_olshevskiy@mail.ru

Ольшевский А.И. Обучающая система работы с контейнерными классами. Рассмотрена классификация по типам основных структур данных контейнерных классов. Описаны возможности реализации их на языке Java с помощью коллекций, а в C++ – библиотекой STL. Определен подход формирования поддерживающий разные наборы операций, среди которых есть совпадающие операции. Реализован простой и понятный в использовании пользовательский интерфейс с подсказками и справкой по теоретической части для каждого типа контейнерных классов, а также примерами программ для конкретной реализации. Проанализирована эффективность разных методов соответствующих операций.

Ключевые слова: структуры данных, контейнерные классы, наборы операций, эффективность разных методов.

Введение

Компьютеризация общества и его различных институтов должна быть направлена не только (не столько) на пассивную (статическую) актуализацию информационных ресурсов общества, но и на создание и актуализацию новых информационных технологий, ресурсов, динамическое их переупорядочивание, а также их взаимопроникновение и взаимообогащение в системах вида «человек–технология–знания–система».

Применение контейнеров обуславливается возросшими объемами обрабатываемой информации. Когда счет используемых объектов идет на сотни тысяч, массивы не обеспечивают ни должной скорости, ни экономии ресурсов.

Но контейнерные классы – довольно сложная структура. Для того, чтобы в ней разобраться и изучить целесообразней использовать обучающую систему, созданную для этого.

Постановка задачи

Для каждого типа контейнера определены методы для работы с его элементами, не зависящие от конкретного типа данных, которые хранятся в контейнере, поэтому один и тот же вид контейнера можно использовать для хранения данных различных типов. В Java эта возможность реализована с помощью коллекций, а в C++ – стандартной библиотекой шаблонов STL (Standard Template Library).

Коллекции – это хранилища, поддерживающие различные способы накопления и упорядочения объектов с целью

обеспечения возможностей эффективного доступа к ним. Они представляют собой реализацию абстрактных типов (структур) данных, поддерживающих три основные операции: добавление нового элемента в коллекцию, удаление элемента из коллекции, изменение элемента в коллекции. В качестве других операций могут быть реализованы следующие: просмотреть элементы, подсчитать их количество и др.

Библиотека STL – это набор шаблонных классов и функций общего назначения. Но основное внимание стоит уделить контейнерам. Контейнеры – это объекты, содержащие другие объекты. Существует несколько различных типов контейнеров, которые реализуют различные алгоритмы обработки содержащихся в них данных [1].

Объектом разработки является коллекций языка Java и шаблоны классов, часть библиотеки C++, в которую входят контейнерные классы, а также алгоритмы и итераторы.

Цель проекта – создание программного обеспечения для повышения эффективного обучения пользователей возможностям коллекций, содержащих контейнеры, реализующие основные структуры данных, проектирования и написания программ, за счет использования мультимедийных технологий. Использование контейнеров позволяет значительно повысить надежность программ, их переносимость и универсальность, а также уменьшить сроки их разработки.

Разрабатываемый пакет программ позволит уменьшить время обучения и упростить затраты на вдумчивое освоение библиотеки для эффективного использования контейнеров.

Исследования

Выполненный обзор существующих интеллектуальных обучающих систем дает следующие виды технологий в интеллектуальных обучающих системах:

- построение последовательности курса обучения;
- интеллектуальный анализ ответов обучаемого;
- интерактивная поддержка в решении задач;
- помощь в решении задач, основанная на примерах.

Первым этапом разработки является классификация по типам основным структурам данных контейнерных классов [2].

Все контейнерные классы языка Java можно разделить на два типа: последовательные и ассоциативные. В C++ добавляется такой тип, как контейнеры – адаптеры.

Последовательные контейнеры обеспечивают хранение конечного количества однотипных величин в виде непрерывной последовательности. К ним относятся векторы (vector), двусторонние очереди (deque) и списки (list), а также так называемые адаптеры, то есть варианты, контейнеров – стеки (stack), очереди (queue) и очереди с приоритетами (priority_queue).

Ассоциативные контейнеры хранят в себе данные вида «ключ – значение» и обеспечивают быстрый доступ к ним по ключу. Эти контейнеры построены на основе сбалансированных деревьев. Существует пять типов ассоциативных контейнеров: словари (map), словари с дубликатами (multimap), множества (set), множества с дубликатами (multiset) и битовые множества (bitset).

Ядро стандартной библиотеки шаблонов включает три основных элемента: контейнеры, алгоритмы и итераторы. Они работают совместно один с другим, представляя тем самым готовые решения различных задач програм-мирования.

Контейнеры – это объекты, которые содержат другие объекты.

Каждый вид контейнера обеспечивает свой набор действий над данными. Выбор вида контейнера зависит от того, что требуется делать с данными в программе.

Алгоритмы обрабатывают содержимое контейнеров. Их возможности включают средства инициализации, сортировки, поиска и преобразования содержимого контейнеров. Многие алгоритмы работают с заданным диапазоном элементов контейнера.

Итераторы – это объекты, которые в той или иной степени действуют подобно указателям. Они позволяют циклически опрашивать содержимое контейнера практически так же, как это делается с помощью указателя при

циклическом опросе элементов массива.

Проектирование программного обеспечения обучающей системы

Для построения курса обучения главное меню состоит из иерархической последовательности вызова форм. В формах, в свою очередь, размещены элементы интерфейса, взаимодействие с которыми вызывает другие элементы или формы. На рисунке 1 показана общая структура вызовов элементов программы и их взаимодействия.

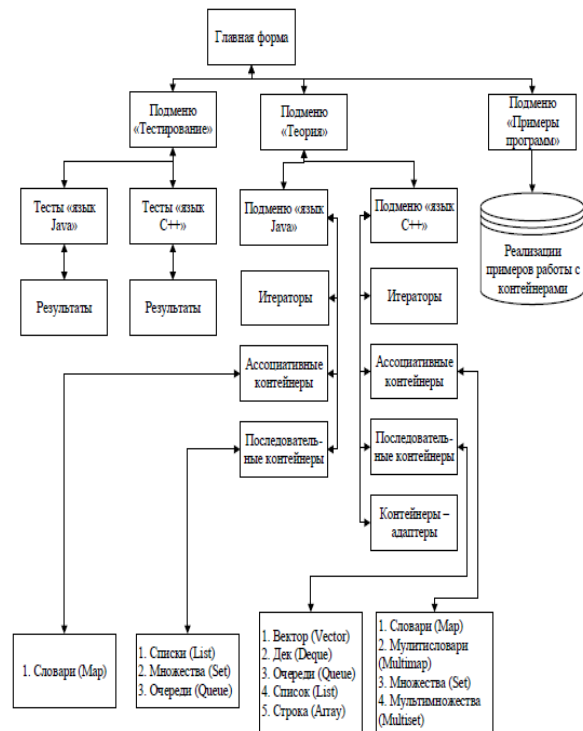


Рисунок 1 – Общая структура вызовов элементов программы и их взаимодействия

В главном меню можно выбрать один из трёх вариантов: «Теория», «Тестирование» или «Примеры программ».

В системе рассматриваются два языка программирования, для которого будет отображена теоретическая информация в соответствующих формах. Каждый язык содержит в себе теорию по различным типам контейнеров, приводятся графические схемы организации данных, как обеспечить доступ к элементам и за какое время, как будет организованно выделение памяти под очередной элемент или перераспределение памяти при удалении.

Подменю «Тестирование» позволяет генерировать случайных образов разные тесты из Базы Вопросов. Выбрав необходимый язык, открывается окно с тестовыми заданиями. Если в результате будут допущены ошибки, программа

автоматически укажет на них.

На вершине иерархии в Java Collection Framework располагаются 2 интерфейса: Collection и Map (см. рис.2). Эти интерфейсы разделяют все коллекции, входящие во фреймворк на две части по типу хранения данных: простые последовательные наборы элементов и наборы пар «ключ – значение» (словари) [3].

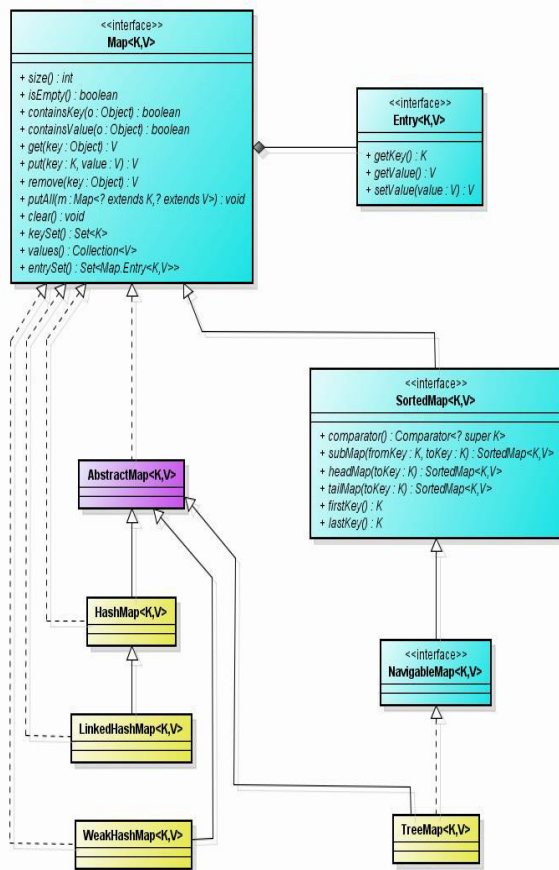


Рисунок 2 – Диаграмма классов ассоциативных контейнеров Collection Framework

Подмену «Примеры программ» содержит в себе базу программных реализаций примеров работы с контейнерными классами. Выбрав необходимый контейнер, будет запущено приложение, в котором демонстрируются функции, обрабатывающие контейнер, что позволяет изучить не только теоретическую информация по этому виду объекта, но и обеспечить выбор набора действий над данными (программную реализацию).

Теоретическая информация по типу контейнера. Например: вектор – это структура, эффективно реализующая произвольный доступ к элементам, добавление в конец и удаление из конца; двусторонняя очередь эффективно реализует произвольный доступ к элементам, добавление в оба конца и удаление из обоих

концов; список эффективно реализует вставку и удаление элементов в произвольное место, но не имеет произвольного доступа к своим элементам и т.п.

Векторы, двусторонние очереди и списки поддерживают разные наборы операций, среди которых есть совпадающие операции. Они могут быть реализованы с разной эффективностью.

В теоретической части приводятся схемы организации данных, как обеспечить доступ к элементам и за какое время, как будет организованно выделение память под очередной элемент или перераспределение памяти при удалении.

Выбор программной реализации позволяет выбрать программу из набора, запустить ее и получить результат. Например работы с вектором: в файле находится произвольное количество целых чисел; программа считывает их в вектор и выводит на экран в том же порядке.

Реализация обучающей системы

Программная реализация создана на языке Java, т.к. он является кроссплатформенным и при помощи JavaFX и приложения Scene Builder очень просто создать необходимый пользовательский интерфейс [4].

JavaFX – платформа для создания RIA, позволяет строить унифицированные приложения с насыщенным графическим интерфейсом пользователя для непосредственного запуска из-под операционных систем, работы в браузерах и на мобильных телефонах, в том числе работающих с мультимедийным содержимым. Среди возможностей этой платформы можно отметить: кроссплатформенность, поддержка каскадных таблицей стилей, поддержка анимации компонентов, возможность работы и отображение 3D графики, поддержка сенсорных экранов и многое другое [5].

Scene Builder – приложение, при помощи которого очень легко и просто создать необходимый интерфейс. Для этого необходимо просто перетащить необходимый компонент (кнопку или текстовое поле) из библиотеки на создаваемую форму. Параметры элементов изменяются в соответственном меню. Так же можно подключить каскадные таблицы стилей (CSS), что в свою очередь позволяет создавать более насыщенный и красивый интерфейс. Приложение автоматически генерирует FXMML-код, который включается в проект.

Единственным недостатком платформы Java является обязательное наличие на компьютере пользователя виртуальной Java-машины, которая необходима для запуска приложения. Во избежание проблем с запуском приложения установочный файл с виртуальной машиной будет включён в состав архива

программы.

При нажатии на кнопку «ТЕОРИЯ» и выборе языка программирования (Java или C++) открывается окно тип контейнера последовательный или ассоциативные (см. рис.3).

Выбрав меню «Списки» открывается окно с информацией о данном интерфейсе, а также есть возможность переключения между теоретической информацией о каждом контейнере.

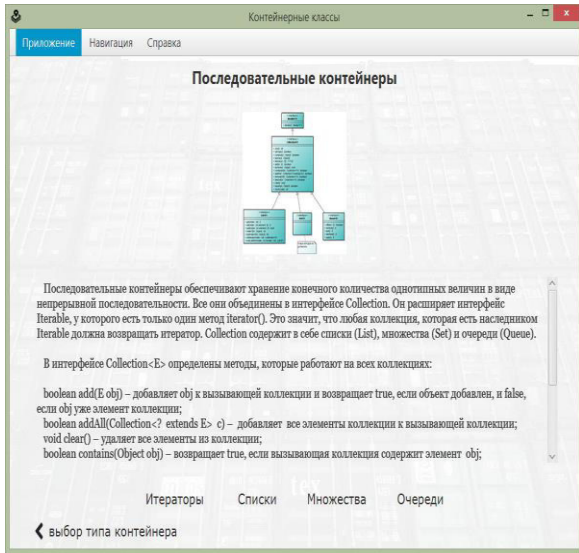


Рисунок 3 – Последовательные контейнеры

Кнопка «принцип работы...» открывает окно, в котором визуализирована и детально описана схема работы конкретного контейнера (см. рис.4).

Принцип работы меню «Множества» и «Очереди» подобен меню «Списки» и отличается лишь составом контейнеров.

Аналогично осуществляется работа при изучении ассоциативных контейнеров.

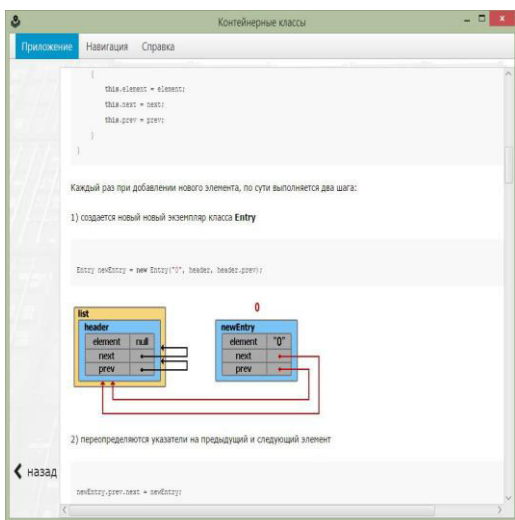


Рисунок 4 – Принцип работы контейнера

Меню «ТЕСТИРОВАНИЕ» открывает генерацию теста. Можно выбрать только один правильный ответ, но некоторые задания имеют несколько правильных ответов (см. рис.5). Допущенные ошибки выделяются красным цветом. После окончания работы пользователя система анализирует сумму ответов.

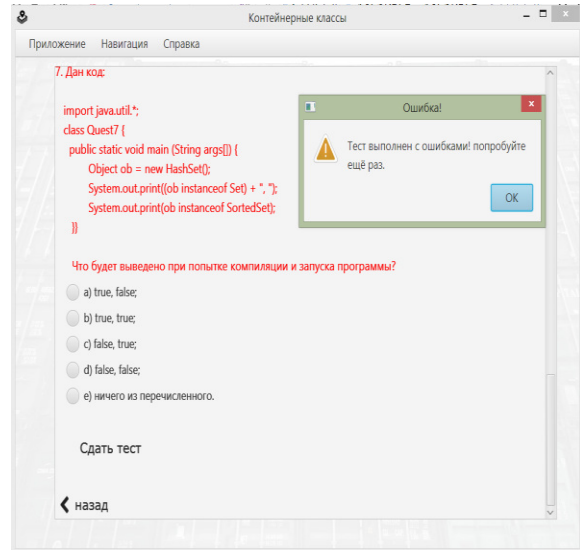


Рисунок 5 – Окно тестирования

Меню «ПРИМЕРЫ ПРОГРАММ» открывает окно со списком доступных примеров программ. При выборе одного из пунктов открывается приложение, демонстрирующее возможности контейнера (см. рис.6).

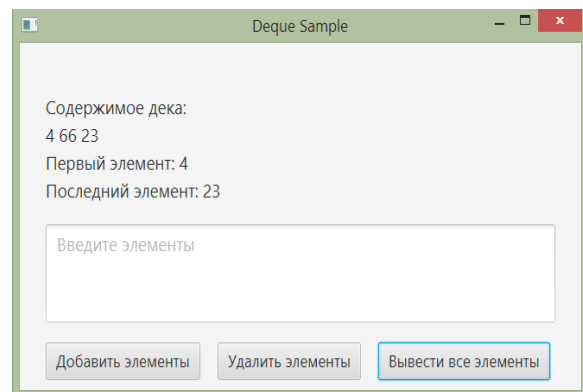


Рисунок 6 – Пример программы работы с деком

Выводы

Были изучены основные принципы проектирования обучающих систем; разработана последовательность курса обучения контейнерных классов высокоуровневых языков программирования: Java, так называемый Collections Framework, и C++ – STL контейнеры; спроектирована общая структура вызовов

элементов программы и их взаимодействия; разработан программный продукт, который позволяет изучить и эффективно использовать контейнерные классы.

Достоинство данной разработки заключается в том, что такая система дает не только теоретические знания в качестве справочника и графического представления объектов (структуры распределения и выделения памяти), но и позволяет провести интеллектуальный анализ ответов обучаемого, а также в интерактивном режиме проверить работоспособность алгоритмов работы с различными типами контейнерных классов.

Разработанную систему можно использовать в ВУЗах для обучения студентов контейнерным классам, а так же любому начинающему программисту.

Литература

1. Павловская Т. А. С/С++. Программирование на языке высокого уровня /

Павловская Т. А. – СПб.: Питер, 2003. – 461 с.

2. Ольшевский А. И. Разработка обучающего пакета программ работы с контейнерными классами / А. И. Ольшевский, А.В. Стась // Сборник материалов VII Международной научно-технической конференции в рамках II Международного Научного форума Донецкой Народной Республики 26 мая 2016г. «Информатика, управляющие системы, математическое и компьютерное моделирование» (ИУСМКМ-2016). – Донецк, С.523-525

3. Шилдт Г. С++: базовый курс / Шилдт Г. – [3-е изд.]; [пер. с англ.]. – М.: Вильямс, 2010. – 624 с.

4. Контейнеры STL. [Электронный ресурс]. – Режим доступа: <https://msdn.microsoft.com/ru/library/1fe2x6kt.aspx>

5. Коллекции в Java. [Электронный ресурс]. – Режим доступа: <http://www.quizful.net/post/Java-Collections>

Ольшевский А.И. Обучающая система работы с контейнерными классами. Рассмотрена классификация по типам основных структур данных контейнерных классов. Описаны возможности реализации их на языке Java с помощью коллекций, а в С++ – библиотекой STL. Определен подход формирования поддерживающий разные наборы операций, среди которых есть совпадающие операции. Реализован простой и понятный в использовании пользовательский интерфейс с подсказками и справкой по теоретической части для каждого типа контейнерных классов, а также примерами программ для конкретной реализации. Проанализирована эффективность разных методов соответствующих операций.

Ключевые слова: структуры данных, контейнерные классы, наборы операций, эффективность разных методов.

Olshevsky AI Training system for working with container classes. Classification by types of main data structures of container classes is considered. The possibilities of implementing them in the Java language with the help of collections are described, and in C++ the library of STL. A formation approach has been defined that supports different sets of operations, among which there are coincident operations. Implemented a simple and intuitive user interface with tips and help on the theoretical part for each type of container classes, as well as examples of programs for a particular implementation. The effectiveness of different methods of corresponding operations is analyzed.

Keywords: data structures, container classes, operation sets, efficiency of different methods.

Статья поступила в редакцию 20.09.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко