

у студентов при использовании в учебных проектах. Создание таких шаблонов и является основной целью данной работы.

Материал и методы. Материалом являются учебные программы и планы дисциплин специальностей ориентированных на подготовку специалистов в сфере ИТ. При разработке программного продукта использовались: интегральная среда разработки Microsoft Visual Studio 2015, пакеты MvcScaffolding 1.0.9 и T4Scaffolding 1.0.8, ORM библиотека EntityFramework. Работа выполнена в соответствии с основными принципами объектно-ориентированного программирования и шаблона Модель-Представление-Контроллер. В качестве основного метода доступа к данным выступает объектно-реляционное отображение.

Результаты и их обсуждение. Во время проектирования шаблонов мы следовали основным принципам разработки web-приложений на платформе ASP.NET изложенных в [1, 2], и применяли подход к написанию собственных Scaffolding шаблонов из [3, 4].

В ходе работы были проанализированы учебные планы ИТ специальностей и программы дисциплин, связанных с разработкой интернет-приложений и проектированием информационных систем. По результатам анализа были разработаны требования к шаблонам. В качестве базы для разработки выбраны шаблоны контроллеров с использованием ORM библиотеки EntityFramework. Результатом работы являются Scaffolding шаблоны, которые наилучшим образом подходят для написания учебных интернет-приложений.

Стоит отметить, что при изменении модели данных, контроллеры и представления сгенерированные на основе Scaffolding шаблонов не обновляются автоматически. Таким образом, следует уделять особое внимание проектированию базы данных.

Заключение. Разработанные шаблоны могут быть использованы студентами, изучающими web-программирование на платформе ASP.NET MVC, при освоении таких дисциплин как «Программирование интернет-приложений» и «Технологии разработки информационных систем», а также при написании курсовых и дипломных работ.

Список литературы

1. Фримен, А. ASP.NET MVC 4 Framework с примерами на C# 5.0 для профессионалов / А. Фримен, С. Сандерсон – 4-е изд. – М.: Вильямс, 2014. – 688 с.
2. Freeman, A. Pro ASP.NET MVC 5 Platform / A. Freeman – Apress, 2014. – 411 p.
3. Waldman, J. ASP.NET MVC: Override the Default Scaffold Templates // MSDN Magazine. – 2014. – V. 29, № 6 – P. 34–45.
4. MvcScaffolding: Creating custom scaffolders [Electronic resource] / S. Sanderson – College Park M.D., 1996. – Mode of access: <http://blog.stevensanderson.com/2011/04/07/mvcscaffolding-creating-custom-scaffolders>. – Date of access: 09.01.2017.

ОБОБЩЕННЫЙ ИНТЕРФЕЙС ЛИНЕЙНЫХ СТРУКТУР ДАННЫХ

*С.В. Сергеев, О.Г. Казанцева
Витебск, ВГУ имени П.М. Машерова*

Дисциплина «Алгоритмы и структуры данных» входит в государственный компонент цикла специальных дисциплин специальности прикладная математика и прикладная информатика. При изучении дисциплины студенты осваивают, в частности, основные подходы при разработке эффективных алгоритмов, способы организации структур данных и технологию их использования, а также получают навыки реализации и использования структур данных [1, 2].

В рамках раздела «Структуры данных» студенты учатся создавать различные абстрактные типы данных. Особую трудность вызывает создание иерархии линейных структур данных.

Цель исследования – создать общую архитектуру классов, включающую в себя такие абстрактные типы данных как стек, очередь, дек, однонаправленный и двунаправленный списки.

Материал и методы. Объектом исследования служат абстрактные типы данных. Предметом исследования выступают абстрактные типы данных, описывающие линейные структуры данных. Для достижения поставленной цели использовались такие методы как анализ, синтез, сравнение, моделирование (с использованием Unified Modelling Language), методы объектно-ориентированного проектирования.

Результаты и их обсуждение. При изучении структур данных основополагающим является понятие «Абстрактный тип данных» (АТД) – набор, включающий данные и выполняемые над ними операции [3].

Абстракция выделяет существенные характеристики некоторого объекта, отличающие его от всех других видов объектов, и четко описывает его концептуальные границы с точки зрения наблюдателя. Объектно-ориентированное проектирование – это метод проектирования, сочетающий процесс объектно-ориентированной декомпозиции и систему обозначений для представления логической и физической, а также статической и динамической моделей проектируемой системы [4].

В работе предлагается архитектура, состоящая из следующих компонентов: абстрактные типы для структур данных, группы операций, итераторы, различные реализации структуры данных, вспомогательные классы.

Анализируя линейные структуры данных можно выделить следующие группы операций над элементами: общие операции (получение итератора, указывающего на точку входа; проверка структуры на пустоту), добавление и удаление элементов в точках входа (каждая операция выделяется в отдельную группу), операции произвольного доступа (вставка элемента перед или после указанного итератором места, удаление указанного итератором элемента). Абстрактные типы данных включают в себя различные группы операций (см. рисунок 1).

Так как в большинстве случаев организация линейных структур данных не зависит от T – типа хранимых в них элементов, то все имеющиеся типы будут обобщенными с параметром T. Кроме того, группы операций будут обобщенными типами с параметрами: тип используемых итераторов, перечисление точек входа.

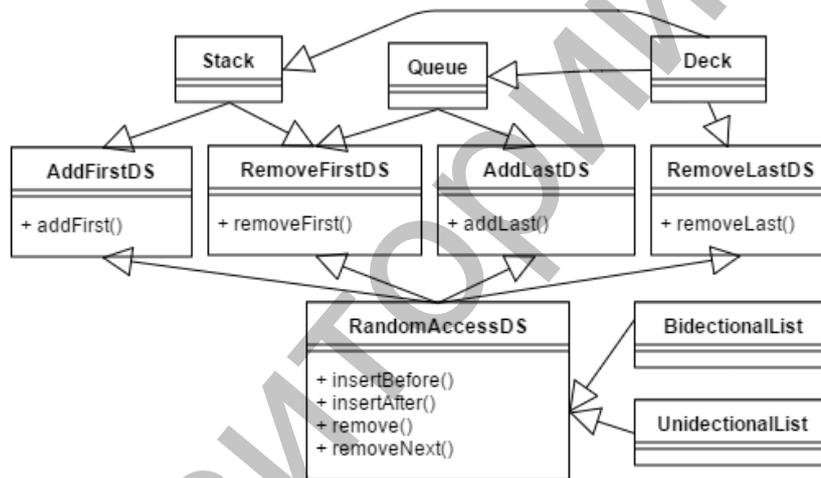


Рисунок 1 – Интерфейсы абстрактных типов данных

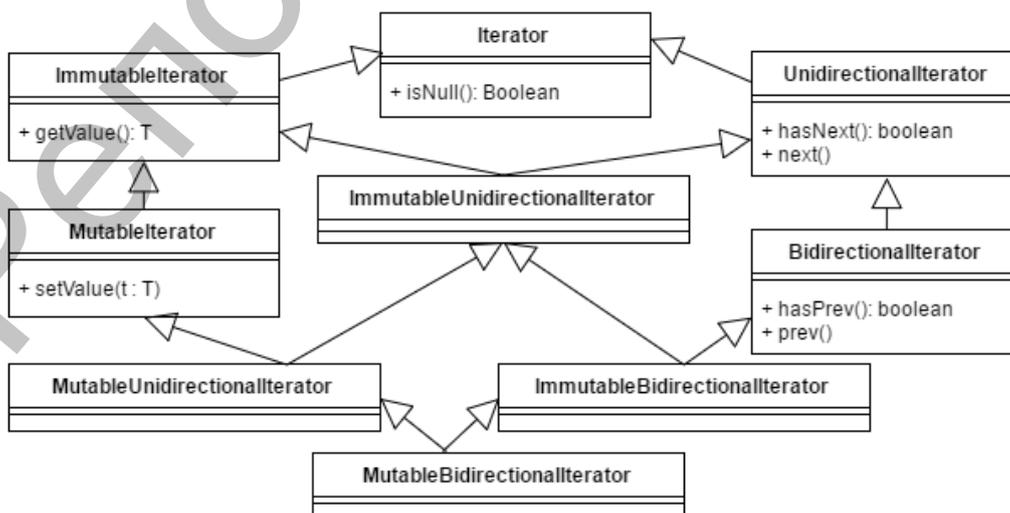


Рисунок 2 – Иерархия итераторов для линейных структур данных

Так как фактические объекты, сохраняющие в себе значения элементов и связи между ними, являются деталями реализации структур данных, то на уровне абстракций информация о положении элемента в структуре данных инкапсулируется в объект, называемый итератором.

Итератор – объект, предоставляющий доступ к каждому элементу набора в некотором порядке [3].

Итераторы линейных структур данных можно классифицировать по двум независимым критериям (см. рисунок 2):

– предоставляет ли итератор возможность изменять элемент, позицию которого он содержит: не изменяющий (ImmutableIterator) или изменяющий итератор (MutableIterator);

– количество направлений, по которым итератор позволяет осуществлять переход: односторонний (UnidirectionalIterator) или двусторонний итератор (BidirectionalIterator).

Итераторы линейных структур данных поддерживают такие базовые операции как проверка наличия элемента в конкретной позиции (isNull), переход к предыдущему (prev) или следующему (next) элементам, получение (getValue) и установка (setValue) значения элемента.

Предлагаемый подход отличается от используемого в стандартных библиотеках языков программирования, например, Java и C++ иной иерархией классов и отсутствием у структур данных не характерных для них операций.

Заключение. Использование понятия АД способствует осознанному следованию принципам объектно-ориентированного проектирования при разработке приложений, использующих линейные структуры данных. Не имея ясного представления об АД, студенты зачастую создают классы, которые только называются «классами», будучи на самом деле лишь наборами плохо согласующихся друг с другом данных и методов. Понимание АД облегчает создание классов и их последующее изменение.

Предложенная архитектура интерфейсов позволяет увеличить наглядность при работе с абстрактными типами данных, а также обеспечить более углубленное освоение линейных структур данных, а также объектно-ориентированного подхода к разработке приложений.

Список литературы

1. Образовательный стандарт Республики Беларусь. Высшее образование. Первая ступень. Специальность 1-31 03 03 Прикладная математика (по направлениям): ОСВО 1-31 03 03-2013. Введ. – 01.09.2013. – Минск : Министерство образования Республики Беларусь: РИВШ, 2013. – III, 32 с.
2. Образовательный стандарт высшего образования. Высшее образование. Первая ступень. Специальность 1-31 03 07 Прикладная информатика (по направлениям): ОСВО 1-31 03 07-2013. Введ. – 01.09.2013. – Минск : Министерство образования Республики Беларусь: РИВШ, 2013. – III, 45 с.
3. Макконнелл, С. Совершенный код. Мастер-класс / С. Макконнелл. – М. : Издательство «Русская редакция», 2010. – 896 с.
4. Объектно-ориентированный анализ и проектирование с примерами приложений / Г. Буч [и др.]. – 3-е изд. – М. : И. Д. Вильямс, 2008. – 720 с.

О ГРАНИЧНЫХ ЗАДАЧАХ С ПОГРАНИЧНЫМ СЛОЕМ

*И.Ф. Соловьева
Минск, БГТУ*

Актуальность рассматриваемой тематики, по мнению автора, может быть объяснена в основном следующими причинами: в связи с учетом новых физических эффектов, с внедрением технических новинок, использующих теплопередачу, диффузию и т. д., возникают новые классы граничных задач, прикладное значение которых широко известно. Они распространены в механике, акустике, динамике жидкостей, физике и других областях науки и техники.

Граничные задачи с пограничным слоем являются математическими моделями, решения которых отличаются сложным характером поведения, в частности, им присуще развитие пограничных слоев с большими градиентами решений, что значительно усложняет решение этих задач.

Причина трудности решений задач с пограничным слоем заключается в неустойчивости численного процесса [1]. Для решения названных выше задач предлагается метод дифференциальной ортогональной прогонки с введением в зонах пограничных слоев регулирующих множителей.

Метод решения. Рассмотрим двухточечные граничные задачи для обыкновенных дифференциальных уравнений второго порядка вида: