

## РЕАЛИЗАЦИЯ ДРЕВОВИДНЫХ СТРУКТУР ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ШАБЛОНОВ

Белорусский государственный университет информатики и радиоэлектроники  
Минск, Республика Беларусь

И. В. Савченко

Г. В. Давыдов – к. т. н., доцент

Древовидная структура – одна из наиболее широко распространенных структур данных в информатике, эмулирующая древовидную структуру в виде набора связанных узлов. Является связанным графом, не содержащим циклы

Любая древовидная структура состоит из связанных узлов. Узел является экземпляром одного из двух типов элементов графа, соответствующим объекту некоторой фиксированной природы. Узел может содержать значение, состояние или представление отдельной информационной структуры или самого дерева. Каждый узел имеет ноль или более узлов-потомков, которые располагаются ниже по древовидной структуре. Узел, имеющий потомка, называется узлом-родителем относительно своего потомка. Каждый узел имеет не больше одного предка. Самый верхний узел называется корневым узлом. В диаграммах корневой узел обычно изображается на самой вершине. В некоторых деревьях, например кучах, корневой узел обладает особыми свойствами. Каждый узел древовидной структуры можно рассматривать как корневой узел подструктуры, начинающейся в этом узле [1].

В процессе реализации программного средства для автоматизированной генерации речеподобных сигналов было доказана возможность статической генерации древовидной структуры с использованием шаблонов.

При таком подходе логическая часть структуры данных строится статически. Таким образом, задается тип данных «дерево», характерный для определенной задачи. Этот тип данных формируется на этапе компиляции. Переменная заданного типа хранит указатель на корневой узел. На корневом элементе начинаются все операции над древовидной структурой.

В то же время параметры дерева задаются динамически: число дочерних узлов для каждого узла, свойства каждого из узлов, информация соответствующая каждому из узлов.

В данном случае шаблон определяет узел. Такой шаблон принимает два параметра:

- 1) тип данных, который определяет информацию, содержащуюся в данном узле
- 2) тип данных, который задает дочерние элементы данного узла.

Желательно, чтобы все типы данных, которые определяют информацию, содержащуюся в узле, были унаследованы от одного и того же класса. Данный класс может быть чисто виртуальным. В таком случае суперкласс выполняет роль интерфейса. Такой подход позволяет унифицировать доступ к данным каждого узла, значительно упростить обход дерева и при необходимости балансировку дерева.[2]

Обходом дерева называется пошаговый перебор элементов по связям между узлами-предками и узлами-потомками. При использовании описанного выше унифицированного способа доступа к информации элементов структуры данных, операция обхода может быть выполнена перемещением указателя по отдельным узлам.

Существуют различные способы обхода: предупорядоченный, поступорядоченный, симметричный, обход в ширину. Унифицированный способ доступа к информации, хранящейся в узлах, позволяет реализовать все описанные типы обхода дерева, не учитывая внутреннюю реализацию каждого элемента.

Основными преимуществами использования шаблонов для задачи построения структуры данных являются:

- позволяют избежать дублирования кода при задании каждого из узлов;
- позволяют избавиться от динамической диспетчеризации и повысить скорость работы приложения;
- позволяют отследить большую часть ошибок на этапе компиляции;
- использование стратегий позволяет не писать сложные классы с нуля, а собирать их из множества меньших.

Основными недостатками использования шаблонов для задачи построения дерева являются:

- недостаточная гибкость динамической конфигурации системы;
- построение структур данных на этапе компиляции.

В разработке программного средства задача построения структуры данных была сведена к разработке шаблона проектирования.

Шаблон проектирования Паттерн это повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках заданного часто возникающего контекста. Шаблон не является законченным образцом [3].

В рамках данной задачи был использован объектно-ориентированный шаблон, который задает отношения и взаимодействия между объектами, являющимися элементами дерева.

Элементами древовидной структуры данных являются на каждом из уровней: фонетический период, синтагма, слово, слог. Набор элементов нижнего уровня с учетом пауз между элементами каждого из уровней образует последовательность пригодную для генерации речеподобных помех. Элементы, хранящие информацию о слогах, не имеют потомков. Слоги в данном случае выступают наименьшей единицей информации.

Список использованных источников

1. Кнут, Д. Э. Искусство программирования / Д. Э. Кнут. – М. : Вильямс, 2000. – 832 с.
2. Страуструп, Б. Язык программирования С++ / Б. Страуструп. – М. : Бином, 2008. – 386 с.
3. Страуструп, Б. Программирование. Принципы и практика использования С++ / Б. Страуструп. – М. : Вильямс, 2011. – 298 с.