

ВЫСОКОУРОВНЕВОЕ МОДЕЛИРОВАНИЕ ПРИ ОБУЧЕНИИ ОСНОВАМ ПРОГРАММИРОВАНИЯ¹

Хакимуллина Анжелика Альвертовна, Дацун Наталья Николаевна,

Пермский государственный национальный исследовательский университет, 614990, Россия,
г. Пермь, ул. Букирева, 15, anzhelika.khakimullina@psu.ru

В настоящей работе рассмотрена проблема написания грамотного и структурированного программного кода людьми, которые обучаются основам программирования. Для ее решения предложено создание программного средства для обучения, которое позволяет генерировать по высокоуровневой спецификации исходный код, необходимый для демонстрации, сравнения и анализа структуры программного кода. Приведено сравнение существующих инструментальных средств генерации исходного кода по следующим критериям: пользовательское представление, внутреннее представление, генерируемые языки, преимущества и недостатки. В ходе анализа сделан вывод о том, что необходимо использовать спецификацию IDEF0 в качестве представления пользователя, потому что она является интуитивно понятной и простой. В результате получен прототип исследуемого программного средства, сделан вывод о дальнейших перспективах данной работы.

Ключевые слова: спецификация IDEF0, генерация кода, обучение основам программирования.

С каждым годом все большее число людей приобщается к программированию. Их знакомство с написанием программного кода начинается с изучения конструкций языков программирования. Но знание лишь синтаксиса языка не гарантирует грамотное, структурированное написание программного кода. Программа может выполнять свою работу корректно, но её исходный код является сложным для восприятия и чтения человеком. Для того чтобы программный код приобрел хорошо отслеживаемую логику, прежде всего, нужна практика, которая требует времени. Зачастую в учебных учреждениях, где готовят ИТ-специалистов, не предусмотрено столько часов практики, сколько необходимо для приобретения навыка написания хорошо структурированных программ. В этом случае учебному процессу могут помочь различные средства обучения.

Обратившись к исследованиям П.Я. Гальперина, можно сделать вывод, что на этапе ознакомления с новым материалом могут быть выделены ориентировочные основы действий трех типов [1]:

- на основе образца действия;
- образцы действий и указания, как выполнять действия с новым заданием;
- обучение анализу новых заданий.

Целью данной работы является разработка программного средства обучения второго типа, которое позволяет генерировать по высокоуровневой спецификации исходный код, необходимый для демонстрации, сравнения и анализа структуры программного кода.

К разрабатываемому средству предъявляются следующие требования: пользовательское представление понятно человеку любого уровня компетенции в сфере программирования, наличие возможности генерации исходного кода по высокоуровневому представлению. На данный момент существует несколько инструментальных средств,

¹ © Дацун Н.Н., Хакимуллина А.А., 2018

которые позволяют на основе высокоуровневых спецификаций генерировать исходный код. Их сравнение представлено в Таблица:

Таблица. Инструментальные средства для генерации исходного кода

	MetaAuto[2]	Flexberry Designer[3]	QReal[4]	MetaLanguage [5]
Представление пользователя	Граф переходов	Диаграмма классов	Диаграмма активностей	Граф «сущность-связь»
Внутреннее представление	XML-формат	Объектная метамодель (C#), JSON (JavaScript)	Представление в виде графа	Объектная метамодель
Генерируемый язык	Любой	C# и JavaScript	C#	Любой
Преимущества	Возможность генерации в любой язык	Возможность увеличения числа языков генерации	Генерация кода программы, отражающего динамические аспекты системы	Возможность генерации в любой язык
Недостатки	Высокие требования к уровню компетенции пользователя	Коммерческий продукт	Неструктурированность диаграммы, неоднозначность генерации	Требуется эксперт для создания метаязыка

Ни одно из вышеуказанных пользовательских представлений не подходит для людей, которые только знакомятся с программированием. Поэтому в основу программного средства обучения была взята высокоуровневая спецификация IDEF0 методологии SADT, с помощью которой можно спроектировать процессы любой предметной области, примерами могут служить работы [6] и [7].

В основе методологии лежат четыре основных понятия: функциональный блок, интерфейсная дуга, декомпозиция, глоссарий [8]. В IDEF0 каждый функциональный блок имеет четыре дуги, которые означают вход, выход, управление и механизм (см. Рис.). В настоящей работе блок – это алгоритм, а его дуги означают входные и выходные параметры, условие и структура данных соответственно. С помощью интерфейсных дуг описывается взаимодействие между блоками, так выход одного блока может быть входом другого.

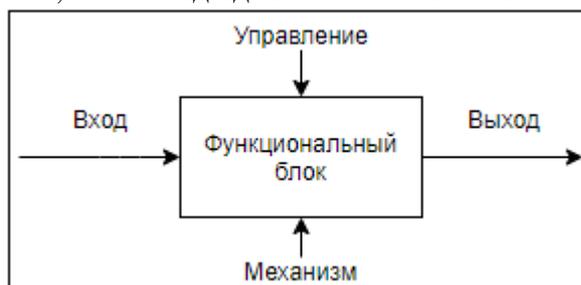


Рис. Функциональный блок

Для разрабатываемого средства были отобраны следующие фундаментальные алгоритмы: нахождение максимума, минимума, суммы, количества элементов, среднего арифметического, удаление элемента по значению; которые используются для базовых структур данных: массив, строка, список, текстовый и бинарный файл.

Для реализации данного средства использовался язык программирования C# в среде программирования Visual Studio. Пользовательским представлением является диаграмма с блоками, основанная на спецификации IDEF0. Внутренним представлением является

объектная метамодель. Средство предоставляет возможность генерации исходного кода в следующие языки: C и Pascal. В разработанном прототипе программного средства есть следующие возможности:

- выбор структуры данных;
- добавление блока на диаграмму (выбор алгоритма соответствующей структуры);
- связывание блоков по именам входных и выходных параметров;
- редактирование (изменение и удаление) функционального блока;
- генерация исходного кода в соответствии с диаграммой на выбранном языке.

Таким образом, данный прототип может служить как вспомогательное средство обучения на занятиях по основам программирования для демонстрации, сравнения и анализа структуры кода. В дальнейших реализациях планируется нивелировать зависимость внутреннего представления от генерируемого языка.

Библиографический список

1. Гальперин П.Я. Общая теория деятельностного подхода к обучению. М.: МГУ, 1968.
2. Канжелев С.Ю., Шалыто А.А. Автоматическая генерация программ с явным выделением состояний. // Журнал «Информационно-управляющие системы» №6. 2006. С. 60-63.
3. Flexberry Designer. [Электронный ресурс] URL: <http://wiki.flexberry.ru/FlexberryDesigner.ashx> (дата обращения: 5.03.2018).
4. QReal [Электронный ресурс] URL: <http://qreal.ru/> (дата обращения: 05.03.2018).
5. Сухов А.О. Разработка инструментальных средств создания визуальных предметно-ориентированных языков.: дис. канд. ф.-м. наук: 05.13.11: защищена 05.12.13: утв. 05.12.13. –М., 2013.
6. Руденко М.В., Апрощенко В.А., Багдасарян Р.Х., Бельченко В.Е., Бельченко И.В., Дьяченко Р.А., Пиотровский Д.Л. Исследование и разработка современной информационной системы оплаты коммунальных услуг на базе мобильных устройств связи.: коллективная монография. – Краснодар, 2017.
7. Григорьев В.П., Камышина Э.Н., Нестеров Ю.И., Никитин С.А. Применение методов искусственного интеллекта в САПР технологического проектирования производства электронной аппаратуры. – М.: МГТУ им. Н.Э. Баумана, 1998.
8. Марка Дэвид А., МакГоуэн Клемент Методология структурного анализа и проектирования SADT. – М: Изд. «Мета технология», 2008.

HIGH LEVEL MODELING IN THE PROGRAMMING TRAINING

Khakimullina Anzhelika A., Datsun Natalia N.

Perm State University, 15, Bukireva st., Perm, 614990, Russia,
anzhelika.khakimullina@psu.ru

It is considered the problem of creating literate and structured program code by people who are taught the basics of programming. It is proposed the creation of software for training, which allows generating the source code for demonstrating, comparing and analyzing the structure of the program code according to the high-level specification. The following criteria are researched for the comparison of the existing source code generation tools: user interface, internal representation, generated languages, advantages and disadvantages. It is concluded that it is necessary to use the IDEF0 specification as a visual base of user interface, because this specification is apprehensible As a result, it is created a prototype of the analyzed software and the future prospects of this work are researched.

Keywords: IDEF0 specification, code generation, learning the basics of programming.