

и

$$b_i^k = \varepsilon_k(x), i = \overline{1, n}. \quad (31)$$

Как и раньше, систему уравнений (29) будем записывать в векторном виде $\Lambda a_k = b_k$.

Заметим, что в качестве приближения к искомому решению можно брать любую из функций $u_k(x), \tilde{y}_k(x), \tilde{z}_k(x)$.

Следует обратить внимание на тот факт, что на основе анализа формул (22)–(27) при $S_0(x) = 0$ приближение $\tilde{z}_1(x)$ совпадает с приближением, построенным по колокационному методу, приближение $u_1(x)$ – с приближением, построенным по улучшенному колокационному методу, а при $\omega_k(x) = 0, k = 1, 2, 3, \dots$, приближение $u_k(x)$ находится по нестационарному методу последовательных приближений.

Алгоритм (22)–(27) можно свести к нестационарному колокационно-итеративному методу решения интегро-функционального уравнения (1). В самом деле, формулы (22), (23), (24) и (27) можно записать в виде

$$(S\tilde{y}_{k-1})(x) = (P_{k-1}S_{k-1})(x),$$

$$(Sy_k)(x) = f(x) + (K\tilde{y}_{k-1})(x) + (K\omega_k)(x),$$

$$\tilde{r}_k = f(x) + (K\tilde{y}_{k-1})(x) + (K\omega_k)(x) - (S\tilde{y}_{k-1})(x) - (S\omega_k)(x).$$

С помощью этих формул условие (26) легко преобразуется к виду

$$(Sy_k - S\tilde{y}_{k-1} - S\omega_k)(x_i) = 0, i = \overline{1, n}.$$

Теперь положим

$$u_k(x) = s_k(x) = (Sy_k)(x), \alpha_k(x) = (S\omega_k)(x), \quad (32)$$

откуда следует

$$y_k(x) = (S^{-1}u_k)(x), \omega_k(x) = (S^{-1}\alpha_k)(x). \quad (33)$$

На основании формул (32), (33) и соотношения (28), получаем

$$u_k(x) = f(x) + (TP_{k-1}u_{k-1})(x) + (T\alpha_k)(x); \quad (34)$$

$$\alpha_k(x) = \sum_{j=1}^n a_j^k \varphi_j(x), k = 1, 2, 3, \dots, \quad (35)$$

$$\int_a^b \{u_k(x) - (P_{k-1}u_{k-1})(x) - \alpha_k(x)\} \varphi_j(x) dx = 0, i = \overline{1, n}. \quad (36)$$

Теперь видно, что соотношения (34)–(36) – это нестационарный колокационно-итеративный метод решения интегро-функционального уравнения (1). В работах [1, 3] установлены различные достаточные условия сходимости и оценки погрешности этого метода для интегральных уравнений Фредгольма второго рода. Из изложенных там результатов, в частности, следует утверждение.

Теорема 2. Если единица не является точкой спектра интегрального оператора T , при $k \rightarrow \infty$, и система функций $\{\varphi_j(x)\}$

полна в пространстве $L_2^k(a; b)$, то существует такой номер n , при котором нестационарный колокационно-итеративный метод (22)–(27) сходится, причем быстрота сходимости увеличивается с ростом n .

Заключение. В статье рассмотрены приближенные методы решения интегро-функциональных уравнений. Исследован колокационно-итеративный (стационарный и нестационарный) методы нахождения приближенных решений интегро-функциональных уравнений.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Вайнико, Г.М. О сходимости и устойчивости метода колокации / Г.М. Вайнико // Дифференц. уравнения. – 1965. – № 2. – С. 244–254.
2. Лучка, А.Ю. Проекционно-итеративные методы решения линейных дифференциальных и интегральных уравнений / А.Ю. Лучка. – Киев: Наук. думка, 1980. – 264 с.
3. Лучка, А.Ю. Быстрота сходимости одного проекционно-итеративного метода, основанного на тригонометрической колокации, для линейных интегральных уравнений / А.Ю. Лучка, Е.М. Луцев // Докл. АН УССР. Сер. А. – 1984. – № 3. – С. 10–13.

Материал поступил в редакцию 10.10.2017

GESELEVA K.G. Kolokatsionny and kolokatsionno-iterative methods solutions of the integro-functional equations

Approximate methods for solving integro-functional equations are considered in the article. Investigated collocation and collocation-iterative (stationary and nonstationary) methods for finding approximate solutions of integro-functional equations.

УДК 004.4

Мухов С.В., Муравьев Г.Л., Савицкий Ю.В., Ашаев Ю.П.

ИСПОЛЬЗОВАНИЕ ТИПИЗАЦИИ ДАННЫХ И ЭЛЕМЕНТОВ ПРОГРАММНЫХ СИСТЕМ ПРИ ОБУЧЕНИИ ПЕРСОНАЛА, СВЯЗАННОГО С ИХ РАЗРАБОТКОЙ И ЭКСПЛУАТАЦИЕЙ

Введение. При изучении технологий обработки данных актуальна максимальная типизация получаемых навыков в силу того, что за счет этого может быть обеспечена высокая надежность функционирования программных систем. При этом необходимо в качестве ра-

бочего материала при обучении использовать системы с перспективой использования полученных навыков в будущем в зависимости от места и целевого предназначения в обработке данных конкретного персонала. При этом типизация существенно определяет качество

Мухов Сергей Владимирович, к.т.н., доцент кафедры информатики и прикладной математики Брестского государственного технического университета.

Муравьев Геннадий Леонидович, к.т.н., доцент, профессор кафедры интеллектуальных информационных технологий Брестского государственного технического университета.

Савицкий Юрий Викторович, к.т.н., доцент кафедры интеллектуальных информационных технологий Брестского государственного технического университета.

Ашаев Юрий Павлович, к.т.н., доцент кафедры информатики и прикладной математики Брестского государственного технического университета.

Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.

системы и стоимость ее эксплуатации и сопровождения. Весьма удачным примером набора типизированных шаблонов является реализация классической модели обработки экономических данных. Можно отметить, что использование различных инструментальных сред в рамках данной модели позволяет проводить достаточно хорошо их сравнительный анализ.

Типизация элементов в упрощенной модели обработки данных, используемой в системах экономического назначения. Функциональная схема обработки данных в рамках модели классической обработки экономических данных рассматривалась авторами ранее в работах [1, 2] в контексте описания ее функционала и интерфейса пользователя.

Отметим, что в данной модели имеет место классическая реализация данных в программных системах, а именно:

- вход системы, реализуемый в виде картотек первичного ввода;
- промежуточные данные типа журналов, которые позволяют реализовать двухуровневую обработку первичных данных;
- промежуточные данные, используемые для формирования выходных отчетных форм;
- справочники, используемые для формирования входного потока данных, выполнения промежуточных расчетных операций и формирования выходных форм;
- выход системы в виде модифицированных справочников и выходных отчетных форм;
- картотеки настройки системы, определяющие функциональные возможности системы.

Рассмотрим более детально типизацию вышеуказанных структур данных и типизацию функциональной части их обработки.

В рамках предлагаемой модели можно говорить о наличии следующих типизированных элементов данных, а именно:

- картотеки первичного ввода со следующим типизированным набором операций сопровождения картотеки, а именно – операций добавления, удаления, редактирования, позиционирования карточки и также ее разноски в некоторые промежуточные картотеки. Отметим, что редактирование полей может выполняться как непосредственно с помощью ручного ввода, так и с помощью типизированной операции выборки данных из некоторой картотеки-справочника;
- картотеки-справочники со следующим типизированным набором операций сопровождения картотеки, а именно – операций добавления, удаления, редактирования и позиционирования карточки. Отметим, что редактирование полей может выполняться как непосредственно с помощью ручного ввода, так и с помощью выборки данных из некоторой картотеки-справочника;
- промежуточные картотеки со следующим типизированным набором операций сопровождения картотеки, а именно – операций просмотра и позиционирования карточки. Среди промежуточных картотек можно выделить журналы операций, картотеки текущего состояния и расчетные картотеки. Журналы операций – это картотеки, использование которых позволяет упростить и типизировать обработку данных для картотек первичного ввода с единообразным функционалом, например, для множества накладных, связанных с перемещением объектов, принято использовать один журнал операций движение объектов. Картотеки текущего состояния объектов системы формируются на базе картотек первичного ввода и служат для оперативного отслеживания наличия и состояния объектов системы и использования этих данных при формировании последующего первичного ввода, связанного с этими объектами. Расчетные картотеки – это картотеки, формируемые некоторым алгоритмическим образом программно на базе других картотек. Как правило, можно говорить в этом случае о достаточно сложных алгоритмах формирования расчетной картотеки;
- картотека настройки системы, которая содержит данные, относящиеся ко всем элементам системы, или данные, определяющие режимы функционирования системы. Отметим, что при обработке системы в учебных целях выгодно использовать в каче-

стве картотеки настройки системы набор данных, который состоит из одной записи. Такая реализация картотеки настройки системы позволяет очень просто связывать данные с помощью SQL запросов при формировании печатных форм.

При реализации экранных форм, используемых при сопровождении картотек, можно выделить следующий минимальный и достаточный набор типизированных программных элементов обработки данных, а именно:

- ручное редактирование поля;
- редактирование поля с использованием справочника;
- позиционирование карточки (назад, вперед, выборка из списка);
- добавить карточку;
- удалить карточку;
- печать карточки;
- печать реестра картотеки;
- разноска текущей карточки в другие картотеки;
- выход из редактирования карточки.

При реализации экранных форм типа меню можно выделить следующий минимальный и достаточный набор типизированных программных элементов обработки данных, а именно:

- вызов экранной формы сопровождения картотеки. Можно выделить функционально различающиеся вызовы экранных форм для картотек первичного ввода, картотек-справочников и картотеки настройка системы;
- вызов программного формирования промежуточных картотек;
- вызов формирования печатной формы;
- выход из системы.

Отметим, что при программном формировании промежуточных картотек можно выделить следующие типовые алгоритмы, а именно:

- формирование выходного набора данных методом увеличения реквизитов входного формирования выходного набора данных методом увеличения реквизитов входного набора данных и формирование этих реквизитов при помощи справочников и (или) использования некоторых специфических алгоритмов пользователя;
- формирование выходного набора данных из журналов операций методом двойной записи, что позволяет в дальнейшем использовать для формирования печатных форм классических генераторов отчетов.

При реализации классических печатных форм можно выделить следующий минимальный и достаточный набор типизированных программных элементов обработки данных, а именно:

- формирование промежуточного набора данных при помощи выборки записей согласно некоторым условиям. Как правило, в экономических системах это выборки по интервальному заданию времени или выборки на равенство по некоторому коду объекта;
- формирование из ранее выбранных данных промежуточного набора данных при помощи сортировки согласно ранее указанным уровням формируемой отчетности. Отметим, что процедуры выборки и сортировки достаточно удачно могут быть реализованы при помощи классических SQL запросов;
- формирование из ранее выбранных и отсортированных записей требуемой печатной формы.

Отметим, что для печатных форм можно выделить следующие типизированные виды отчетных форм, а именно:

- классический реестр картотеки;
- классическая списковая отчетная форма с выводом информации по каждой обрабатываемой записи и формированием итоговых сумм по уровням отчета;
- классическая итоговая отчетная форма с формированием итоговых сумм по уровням отчета.

Отметим, что процедуры выборки и сортировки достаточно удачно могут быть реализованы при помощи классических SQL запросов.

В силу достаточно сильной типизации процедур в случае использования при разработке системы СУБД за счет вынесения в

отдельные структуры определения данных весьма эффективно применение классических генераторов экранных и печатных форм.

Так как при реализации систем используется одна модель данных в рамках нескольких альтернативных инструментальных сред у обучаемых формируются навыки сравнительного качественного анализа конкретной инструментальной среды. Оценивание инструментальной среды можно проводить по следующим критериям:

- относительная сложность исходного кода, которая непосредственно связана с качеством и надежностью сопровождения;
- сложность и надежность инструментария;
- сложность и надежность сопровождения данных;
- сложность и надежность восстановления данных;
- относительная сложность и трудозатраты при использовании шаблонов для реализации типовых элементов.

Вышеуказанный набор типизированных программных элементов является минимальным и достаточным для создания программных систем универсального назначения практически любой информационной емкости и сложности обработки данных с использованием любых реляционных баз данных или реляционной модели в случае использования табличных редакторов типа MS Excel. Отметим, что типизация данных и методов их обработки позволяет эффективно реализовывать взаимодействие разработчиков и эксплуатационных служб за счет однозначного и типизированного подхода персонала к элементам программной системы.

Предложенная модель обработки данных с соответствующей типизацией методов их обработки была достаточно удачно отработана в процессе обучения на различных инструментальных средах, а именно: MS Access, MS Excel, FoxPro, CBuilder, «чистый» C GCC в совокупности с поддержкой графического интерфейса GTK+ и обеспечением доступа к данным с помощью СУБД MySQL. Отметим, что для специалистов в области информационных технологий представ-

ляет интерес обработка навыков и методик переноса программного продукта, а также совместимости разработок для различных инструментальных сред.

Заключение. Использование вышеуказанной модели обработки данных позволяет отработать при преподавании дисциплин, связанных с изучением информационных технологий, навыки использования шаблонов при проектировании, разработке, сопровождении и эксплуатации программных систем. В силу этого можно говорить об обеспечении более качественного уровня подготовки специалистов за счет типизации навыков оценки качества, надежности, стоимости и затрат на сопровождение программных систем.

Массовое использование элементов вышеуказанной модели при обработке данных экономического назначения позволяет говорить о том, что опыт, который получен в процессе ее программной реализации в различных инструментальных средах, будет полезен при подготовке специалистов, как связанных с разработкой программных средств, так и специалистов, умеющих грамотно эксплуатировать эти программные средства.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Мухов, С.В. Модель бухгалтерского учета как основа систематизации компьютерной бухгалтерии / С.В. Мухов, Г.Л. Муравьев / Бухгалтерский учет и анализ: научно-практический журнал. – Минск. – 2010. – № 46 (162). – С. 40–42.
2. Мухов, С.В. Сравнительный анализ использования различных инструментальных средств при разработке систем обработки данных экономического назначения / С.В. Мухов, Г.Л. Муравьев, С.И. Парфомук // Вестник Брестского государственного технического университета. – Брест: УО БрГТУ, 2015. – № 5 (95). – С. 61–64.

Материал поступил в редакцию 29.03.2018

MUHOV S.V., MURAVJEV G.L., SAVITSKIY Yu.V., ASHAEV Yu.P. Use of typification of data and elements of program systems when training the personnel connected with their development and operation

When training students of the information technologies specializing in area within disciplines of information profile it is offered to use classical model of data processing in economic systems for working off of skills of development of systems on the basis of the typified templates.

In work the standard model of data processing is offered and also the minimum and sufficient set of standard modules and elements of the developed systems is allocated. System development within this model will allow students further on the basis of the realized educational project to develop the systems practically of any volume and level of complexity with minimization of costs of development.