

УДК 004.052

В.Ю. ДУБНИЦКИЙ, А.М.КОБЫЛИН, О.А. КОБЫЛИН*Харьковский институт банковского дела УБД НБУ, Украина***СИСТЕМА ДИСТАНЦИОННОГО ОЦЕНИВАНИЯ ИНТЕРВАЛЬНОЙ НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПРЕДНАЗНАЧЕННОГО ДЛЯ ВЫПОЛНЕНИЯ ФИНАНСОВЫХ РАСЧЕТОВ**

Описана структура программной системы построенной по технологии ASP.NET, позволяющей динамически создавать документы на Web-сервере, когда они запрашиваются по протоколу HTTP. и предназначены для дистанционного оценивания интервальной надежности программного обеспечения в условиях нестochasticески неопределённых количественных значений параметров программной системы, определяющих её надёжность.

интервальные вычисления, линейные уравнения с интервальными коэффициентами, надежность программного обеспечения, Web-приложения, Web-сервер

Введение

Анализ литературы. В последние годы наметился переход от архитектуры «толстый клиент» к Web-приложениям. Интеграция серверных систем и прозрачное совместное использование данных, бывшие когда-то мечтой ИТ -отделов, уступили доводам, связанным с уменьшением совокупной стоимости владения, установки, не требующей места на локальном диске, и возможности запуска приложений из любого места, имеющего выход в Интернет.[1,2] Главная задача, стоящая сегодня перед разработчиками такова. Пусть эта программа работает в Web. К сожалению, программировать для Web с использованием старых технологий нелегко. Написание приложений типа Microsoft Excel и Microsoft Word – хорошо изученный вид программирования. Сегодня Web-разработка ведется с помощью инструментов и технологий первого поколения, больше похожих на BASIC, чем на современные платформы и среды, к которым привыкли разработчики. Ответ фирмы Microsoft на этот вызов времени – модель программирования второго поколения Web Forms. Эта модель часть системы ASP.NET, которая в свою очередь является частью Microsoft .NET Framework. Если простая в работе модель Active

Server Pages (ASP) для динамической генерации HTML на Web-серверах революционизировала Web-программирование, то ASP.NET идет дальше, вводя многократно применимые серверные элементы управления, формирующие HTML для клиентских браузеров и генерирующие события, которые могут быть обработаны сценариями на сервере. Суть Web-Forms – Web-страницы, содержащие элементы управления и обработки событий.

Термин Web-сервисы XML многократно встречается в любой книге, докладе или статье по Microsoft .NET. Web-сервисы XML, или, как их чаще называют. Web-сервисы, – краеугольный камень системы Microsoft .NET. Это ключ к Microsoft'овскому видению мира, в котором компьютеры общаются в Web, используя HTTP и другие повсеместно поддерживаемые протоколы. И именно они являются основной причиной, по которой Microsoft .NET Framework вообще существует: сделать все, чтобы максимально упростить создание Web-сервисов и Web-клиентов, Web-сервис – это особый тип Web-приложения. В отличие от традиционного Web-приложения у него нет пользовательского интерфейса. Вместо этого он предоставляет функции, более известные как Web-методы, которые могут быть вызваны по Интернету.

Принцип работы Web-приложений. Web-приложения используют архитектуру «клиент — сервер». Собственно, Web-приложение находится на сервере и обрабатывает запросы, которые передают ему через Интернет многочисленные клиенты. Схематически это показано на рис. 1. На стороне клиента Web-приложение работает в браузере, например в Internet Explorer или Netscape Navigator. Пользовательский интерфейс Web-приложения передается на клиентскую машину в виде страниц на языке HTML (Hypertext Markup Language), где браузер интерпретирует и отображает их.

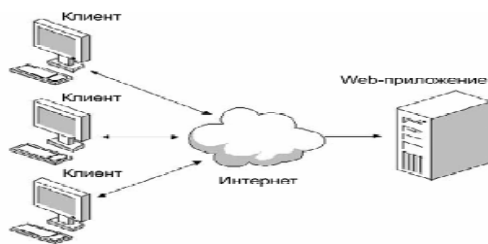


Рис. 1. Архитектура Web-приложений

На стороне сервера Web-приложение работает под управлением IIS (Internet Information Services). IIS управляет работой приложения, передает ему клиентские запросы и возвращает клиентам результаты исполнения их запросов. Запросы и результаты их исполнения передаются через Интернет по протоколу HTTP (Hypertext Transport Protocol). Протокол реализуется через среду, такую, как Интернет. На рис. 2 показано, как клиент и сервер взаимодействуют через Интернет.

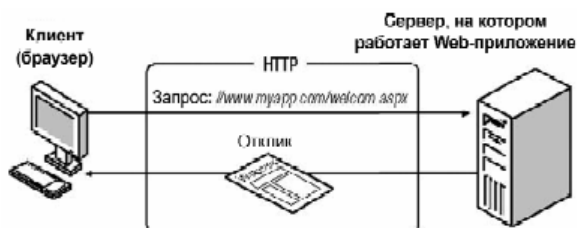


Рис. 2. Взаимодействие клиента и сервера при работе Web-приложения

Web-приложение компонует отклик из серверных ресурсов, к которым относятся исполняемый код, работающий на сервере (то, что традиционно

считается «приложением» в Windows-программировании), Web-формы, HTML-страницы, графические файлы и иное содержимое, составляющее информационное наполнение приложений. Web-приложения во многом напоминают традиционные Web-сайты, но в отличие от них отображают пользователю динамическое содержимое, генерируемое исполняемым кодом приложения, а не статические страницы, хранящиеся на сервере в готовом виде. На рис. 3 показано, как Web-приложение компонует HTML-страницы, возвращаемые пользователям.



Рис. 3. Web-приложение генерирует отклик из серверных ресурсов

Исполняемая часть Web-приложения способна делать многое из того, чего не могут статические Web-сайты, а именно:

- принимать данные от пользователя и сохранять их на сервере;
- выполнять для пользователя различные действия: размещать заказы, делать сложные вычисления и извлекать информацию из баз данных (БД);
- опознавать пользователя и отображать интерфейс, настроенный в соответствии с его предпочтениями;
- отображать постоянно меняющееся содержимое, например инвентарные списки, обрабатываемые заказы и сведения об отгружаемых товарах.

Этот перечень далеко не полон. В принципе, Web-приложения способны решить любые воображаемые задачи, доступные и клиент-серверному приложению. Особенность Web-приложений в том, что взаимодействие между клиентом и сервером осуществляется через Интернет. Следует отметить, что при построении рис. 1 – рис. 3 была использована работа [1].

Ранее в работе [6] для решения задач интервального оценивания надежности программ была использована предложенная в [4] система аксиом A_1 , в соответствии с которой действия с интервальными числами выполнялись так:

$$[A] + [B] = (a_1, a_2) + (b_1, b_2) = [(a_1 + b_1), (a_2 + b_2)] \quad (1)$$

$$[A] - [B] = (a_1, a_2) - (b_1, b_2) = [(a_1 - b_2), (a_2 - b_1)] \quad (2)$$

$$[A] \cdot [B] = (\min U, \max U) \quad U = (a_1 b_1, a_1 b_2, a_2 b_1, a_2 b_2) \quad (3)$$

$$[A] / [B] = (a_1, a_2) \cdot \left(\frac{1}{b_2}, \frac{1}{b_1} \right), \quad 0 \in [b_1, b_2]. \quad (4)$$

В связи с необходимостью решения обратной задачи – подбора параметров программы, гарантирующих ее необходимую надежность, в работе [4] использовали приведенную в работе [5] систему аксиом A_2 .

Пусть операция

$$\circ \in \{+, -, \cdot, / \}. \quad (5)$$

Тогда для двух интервальных чисел $[A], [A_2]$ справедливо условие:

$$2.1. \quad [A_1] \circ [A_2] = (\min U, \max U), \quad (6)$$

где

$$U = ((a_1, a_2) \circ (b_1, b_2)) = (a_1 \circ b_1), (a_1 \circ b_2), (a_2 \circ b_1), (a_2 \circ b_2). \quad (7)$$

Сравним выполнение операции (-) в этих системах аксиом, используя геометрическую иллюстрацию.

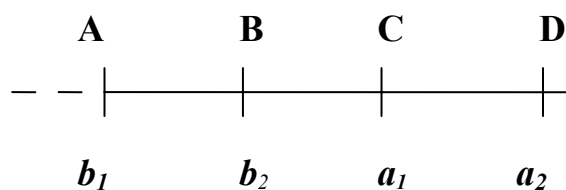


Рис. 4. Геометрическая иллюстрация выполнения операции (-)

Очевидно, что:

$$DA = a_2 - b_1; \quad CA = a_1 - b_1; \quad DB = a_2 - b_2;$$

$$CB = a_1 - b_2.$$

Тогда отрезком наибольшей длины будет отрезок DA, наименьшей – отрезок CB. Таким образом показано совпадение этих двух систем аксиом потому, что условия (1), (3), (4) совпадают с условием (2.1).

Следуя работе [5] назовем интервалы $[A] = (a_1, a_2)$ и $[\bar{A}] = (a_2, a_1)$ сопряженными.

В этой же работе показано, что решения алгебраических уравнений, коэффициенты которых есть интервальные числа могут быть получены в следующем виде.

Уравнение I типа:

$$[A] + [X] = [B], \quad (8)$$

тогда

$$[X] = [B] - [A]. \quad (9)$$

Уравнение II типа:

$$[A] \cdot [X] = [B], \quad 0 \notin [A], \quad (10)$$

тогда

$$[X] = \frac{[B]}{[A]}. \quad (11)$$

Уравнение III типа:

$$[A] [X] + [B] = [C], \quad (12)$$

тогда

$$[X] = \frac{[C] - [\bar{B}]}{[A]}, \quad (13)$$

при условии, что $0 \notin [A]$.

Приведем численные примеры.

Рассмотрим решение уравнения типа I.

$$[X] + [5; 7] = [14; 18]$$

$$[X] = [14; 18] - [7; 5]$$

$$U = (7; 11; 9; 13)$$

$$\min U = 7, \quad \max U = 13$$

$$[Y] = [7; 13].$$

Рассмотрим решение уравнения типа II.

$$[X] [5; 7] = [14; 18].$$

$$[X] = \frac{[14; 18]}{[7; 5]};$$

$$U = \left(2; \frac{14}{5}; \frac{18}{7}; \frac{18}{5}\right), \min U = 2; \max U = \frac{18}{5};$$

$$[Y] = \left[2; \frac{18}{5}\right].$$

Рассмотрим решение уравнения типа III.

Пусть $[A]=[2; 5]$; $[B]=[4; 9]$; $[C]=[15; 23]$.

Тогда:

$$[2; 5] * [X] + [4; 9] = [15; 23]$$

В соответствии с (13)

$$[X] = \frac{[15; 23] - [9; 4]}{[5; 2]} = \left[\frac{6}{5}; \frac{19}{2}\right].$$

Следовательно, при решении прямых задач применение аксиом A_1 и A_2 приводит к одинаковым результатам, при решении обратных задачи следует использовать систему аксиом A_2 .

Цель работы: разработка алгоритмов и создание программного обеспечения для удалённого доступа к специализированному калькулятору, пред-назначенному для выполнения расчётов с числами, заданными в интервальном виде.

Изложение результатов

Предложенные методы и технологии программирования были использованы при выполнении финансовых расчётов.

Рассмотрим пример использования Web-приложения для расчета месячных платежей по займу.

Расчет срочной годовой уплаты производится по формуле

$$Y = D \frac{i(1+i)^n}{(1+i)^n - 1}, \quad (14)$$

где: i – процентная ставка; n – срок кредита; D – величина кредита. Вид формы веб-приложения приведен на рис. 5.

Web-форма расчета месячных платежей для обычных вычислений и интервального калькулятора основных арифметических операций приведена на рис.6.

Формула (14) для расчета месячных платежей по займу в интервальном виде будет иметь вид:

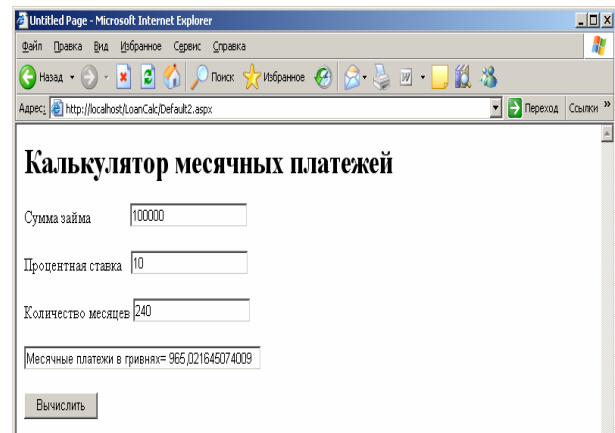


Рис. 5. Web-форма расчета месячных платежей

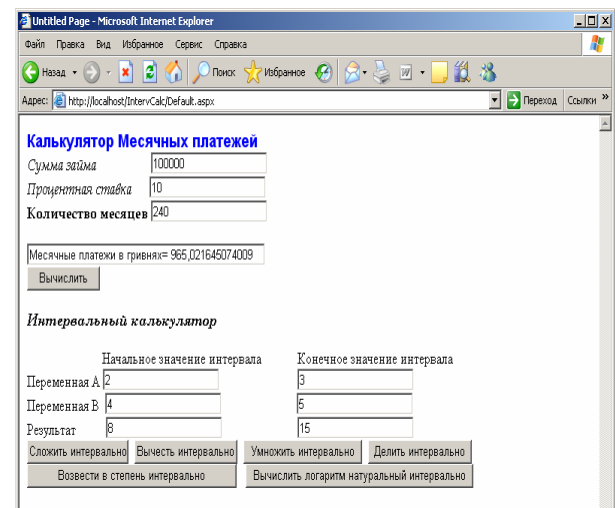


Рис. 6. Web-форма расчета месячных платежей и интервального калькулятора

$$[Y_n; Y_n] = [D_n; D_n] \frac{[i_n; i_n] (1 + [i_n; i_n])^{[n_n; n_n]}}{(1 + [i_n; i_n])^{[n_n; n_n]}}, \quad (15)$$

где $[i_n; i_n]$ – интервальное значение процентной ставки;

$[n_n; n_n]$ – интервальное значение срока кредита;

$[D_n; D_n]$ – интервальное значение величины кредита;

$[Y_n; Y_n]$ – интервальное значение месячных платежей.

Web-форма расчета месячных платежей в интервальном представлении исходных данных приведена на рис. 7 и 8.

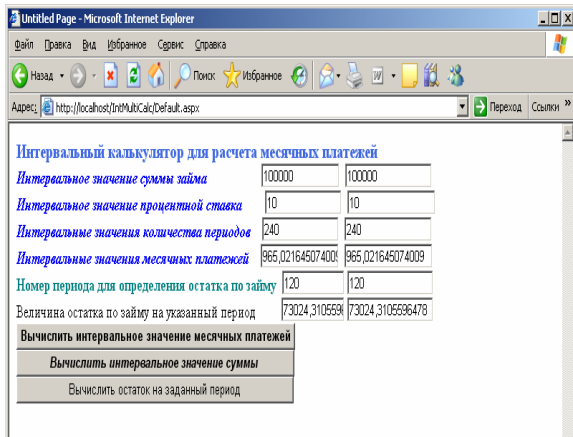


Рис. 7. Web-форма расчета месячных платежей в интервальном представлении (значения верхнего и нижнего интервалов совпадают)

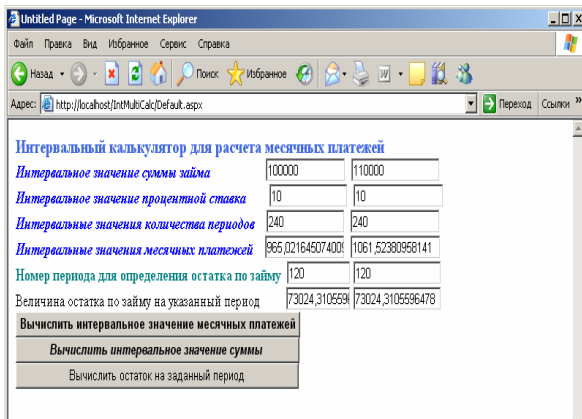


Рис. 8. Web-форма расчета месячных платежей в интервальном представлении (значения верхнего и нижнего интервалов не равные)

Web-форма решения обратной задачи расчета месячных платежей в интервальном представлении исходных данных приведена на рис.9.

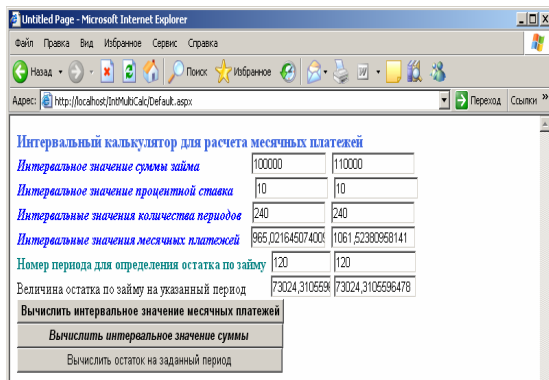


Рис. 9. Web-форма решения обратной задачи расчета месячных платежей в интервальном представлении

При разработке программного обеспечения для выполнения подобных расчётов возникла потребность в формулировании в техническом задании на программирование требований к структуре ПО, обеспечивающей необходимую надёжность процесса вычислений.

Далее рассмотрены различные модели оценки надежности программного обеспечения с использованием аппарата интервальных вычислений.

Модель Холстеда

$$N_{\text{ошибок}} = \frac{V}{E_{\text{критическое}}}, \quad (16)$$

где V – объем программы;

$E_{\text{критическое}}$ – эмпирическая постоянная $E = 2 \cdot 10^3 \dots 5 \cdot 10^3$;

$N_{\text{ошибок}}$ – количество ошибок в программе.

В интервальном виде эта модель имеет вид:

$$[N_{\text{ошибок}}] = [A] [V], \quad (17)$$

где

$$[N_{\text{ошибок}}] = [2 \cdot 10^{-4}; 5 \cdot 10^{-4}] [V], \quad (18)$$

то есть получено уравнение вида I.

На рис. 10 Web-форма решения обратной задачи оценки надежности программного обеспечения с использованием интервального калькулятора модели Холстеда.

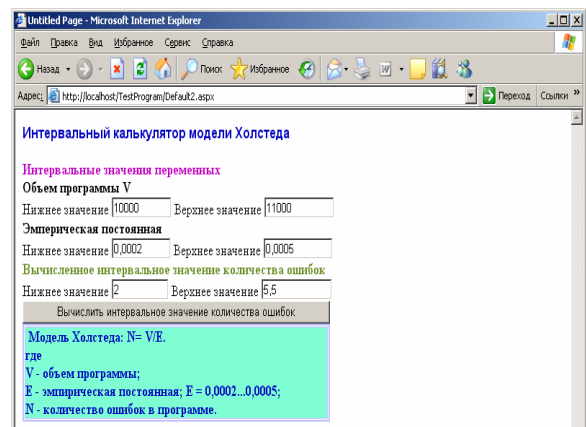


Рис. 10. Web-форма решения обратной задачи оценки надежности программного обеспечения с использованием интервального калькулятора модели Холстеда

Модель Миллса

$$N_{\text{ош}} = \frac{N_{\text{отм}} - N_{\text{внес}}}{N_{\text{внесв}}} \quad (19)$$

где $N_{\text{ош}}$ – количество собственных дефектов в программе;

$N_{\text{отм}}$ – количество выявленных собственных дефектов в программе;

$N_{\text{внес}}$ – количество внесенных дефектов;

$N_{\text{внесв}}$ – количество выявленных внесенных дефектов.

Для решения обратной задачи возможны три варианта: решение относительно $N_{\text{отм}}$, относительно $N_{\text{внес}}$, относительно $N_{\text{внесв}}$. Однако, в любом случае задача сводится к уравнению вида II. Его варианты приведены в табл 1.

Таблица 1

Условия задачи по определению параметров модели Миллса

[A]	[X]	[B]
$[N_{\text{отм}}] / [N_{\text{внесв}}]$	$[N_{\text{внес}}]$	$[N_{\text{ош}}]$
$[N_{\text{внес}}] / [N_{\text{внесв}}]$	$[N_{\text{отм}}]$	$[N_{\text{ош}}]$
$[N_{\text{отм}}] / [N_{\text{внес}}]$	$[X] = [1 \ ; \ 1] [N_{\text{внес}}]$ $[N_{\text{внес}}] = [X]^{-1}$	$[N_{\text{ош}}]$

На рис. 11 показана Web-форма решения прямой и обратной задачи оценки надежности программного обеспечения с использованием интервального калькулятора модели Милса

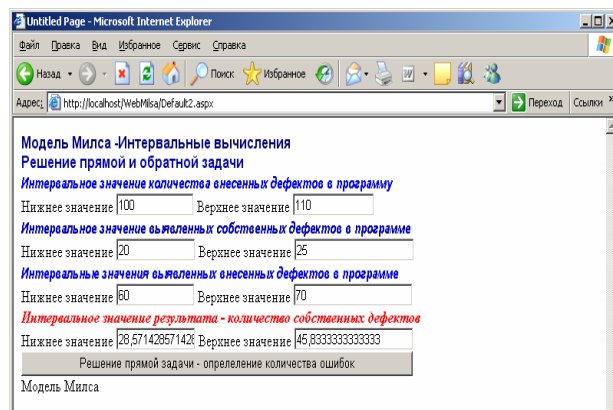


Рис. 11. Web-форма решения прямой и обратной задачи оценки надежности программного обеспечения с использованием интервального калькулятора модели Милса

Модель фирмы IBM имеет вид:

$$N = \alpha \cdot \text{MUM} + \gamma \cdot \text{UM} \quad (20)$$

где α, γ – эмпирические постоянные, $\alpha=23; \gamma=2$.

MUM – количество многократно исправляемых модулей (более десяти раз);

UM – количество исправляемых модулей (не более десяти раз).

В свою очередь

$$\text{UM} = 0,9 \text{NM} + 0,15 \text{CUM} \quad (21)$$

$$\text{MUM} = 0,15 \text{UM} + 0,06 \text{CUM}; \quad (22)$$

где NM – количество модулей, CUM – количество старых исправляемых модулей.

Подставив (21) и (22) в (20) в интервальном виде получим уравнение

$$N = [3,45; 3,45] * [\text{MUM}] + [1,8; 1,8] * [\text{NM}] + [1,68; 1,68] * [\text{CUM}]. \quad (23)$$

Пусть

$$[3,45; 3,45] = [F]; [1,8; 1,8] = [G]; [1,68; 1,68] = [Q].$$

Тогда

$$[N] = [F] * [\text{NUM}] + [G] * [\text{NM}] + [Q] * [\text{CUM}]. \quad (24)$$

При решении этого уравнения относительно одного из параметров получим уравнение типа III.

Варианты постановки задачи приведены в табл. 2.

Таблица 2

Условия задачи по определению параметров модели фирмы IBM¹

[A]	[X]	[B]	[C]
[A]	[M]	$[B] * [\text{NM}] + [C] * [\text{CUM}]$	[N]
[B]	[NM]	$[A] * [M] + [C] * [\text{CUM}]$	[N]
[C]	[CUM]	$[A] * [M] + [B] * [\text{NM}]$	[N]

$$^1 \text{Модель вида } [A] \cdot [X] + [B] = [\bar{C}]$$

Следующие примеры демонстрируют применение Web-сервисов для выполнения расчетов с применением интервального калькулятора. На рис. 12 - 15 приведены интервальный калькулятор в Internet Explorer, тестовая страница для одной из операций интервального калькулятора и XML, возвращаемый этим методом.

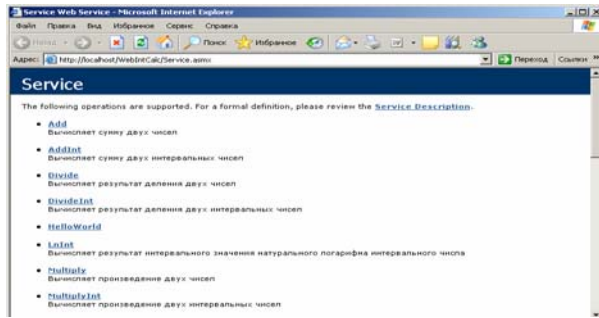


Рис. 12. Интервальный калькулятор в Internet Explorer

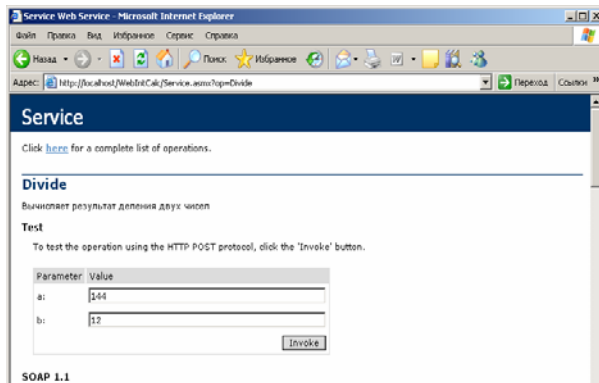


Рис. 13. Тестовая страница для метода Divide

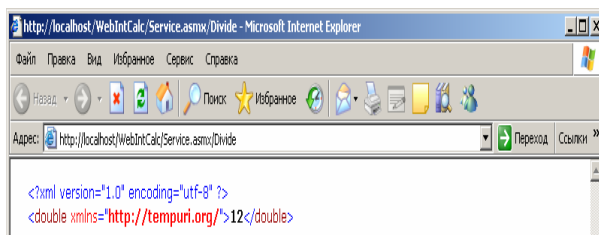


Рис. 14. XML, возвращаемый методом Divide

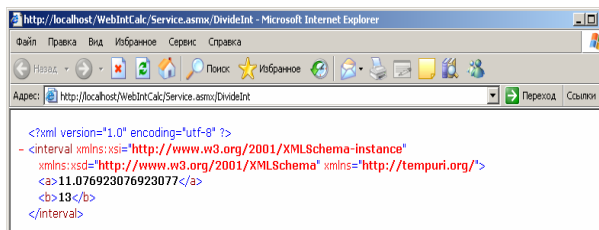


Рис. 15. XML, возвращаемый интервальной операцией

Выводы

1. Для дистанционного оценивания интервальной надежности программного обеспечения, предназначенного для выполнения финансовых расчетов предложена система, основанная на

решении линейных уравнений, коэффициенты которого есть интервальные числа.

2. Решение этих уравнений проведено с использованием технологии ASP.NET. Приведены примеры решения задач по интервальной оценке параметров, определяющих надёжность программного обеспечения, выполненные на специализированном программном калькуляторе.

Литература

1. Просис Дж. Программирование для Microsoft .NET. Пер. с англ. – М.: Издательско-торговый дом Русская Редакция. – 307 с.
2. Экспозито Д. Microsoft ASP.NET 2.0. Базовый курс. Мастер-класс / Пер. с англ. – М.: Издательство Русская редакция; СПб.: Питер, 2007. – 688 с.
3. Дубницкий В.Ю., Кобылин А.М., Супрун И.А. Интервальное вычисление эффективности тестирования компьютерных программ // Системы обработки информации. Х.:ХВУ, 2006. – Вып 6 (55). – С. 66-71.
4. Алефельд Г., Херцбергер Ю. Введение в интервальные вычисления. – М.: Мир, 1987. – 250 с.
5. Алтунин А.Е., Семухин М.В., Модели и алгоритмы принятия решений в нечетких условиях. Тюмень: ТГУ, – 2000. – 352 с.
6. Дубницкий В.Ю, Кобылин А.М. Интервальное решение обратной задачи оценки эффективности тестирования компьютерных программ // Радиоелектронні і комп'ютерні системи. – 2007. – № 8 (27). – С. 181-183.

Поступила в редакцию 28.01.2008

Рецензент: канд. техн. наук, доц. В.В. Скляр, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.