www.volsu.ru

ИННОВАЦИИ В ИНФОРМАТИКЕ, ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ И УПРАВЛЕНИИ

DOI: https://doi.org/10.15688/NBIT.jvolsu.2021.4.1

УДК 004.052.4
ББК 30.14

МЕХАНИЗМЫ ОБЕСПЕЧЕНИЯ НАДЕЖНОСТИ ОБЪЕКТОВ ИНФОРМАЦИОННЫХ СИСТЕМ

Елизавета Евгеньевна Бандурова

Студент, кафедра информационной безопасности,
Волгоградский государственный университет
ibb-181_837244@volsu.ru
просп. Университетский, 100, 400062 г. Волгоград, Российская Федерация

Татьяна Александровна Омельченко

Старший преподаватель, кафедра информационной безопасности,
Волгоградский государственный университет
omelchenko.tatiana@volsu.ru
просп. Университетский, 100, 400062 г. Волгоград, Российская Федерация

Аннотация. В данной статье рассматриваются основные механизмы обеспечения надежности объектов информационных систем, выделяются критерии для их оценки. Проводится сравнительный анализ механизмов обеспечения надежности объектов информационных систем по выделенным критериям. Представлена формализованная модель, архитектура и интерфейс программного средства, проведена серия экспериментальных исследований.

Ключевые слова: надежность, информационная система, программное средство, предупреждение ошибок, обнаружение ошибок, исправление ошибок, устойчивость к ошибкам.

Очень трудно представить функционирование современного общества без информационных систем (ИС) различного уровня сложности. Контроль качества функционирования систем – одна из главных задач, на которую следует обратить пристальное внима-

ние. Одним из критериев качества информационной системы является надежность, то есть свойство объекта сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных режимах и условиях эксплуатации. Чем качественнее будет система, тем она надежнее, и наоборот. В связи с этим цель исследования – выбор наиболее подходящего механизма обеспечения надежности информационных систем.

Для того, чтобы определить механизмы обеспечения надежности ИС, сначала нужно выделить уязвимые объекты информационной системы [4], а затем сопоставить к ним угрозы, влияющие на надежность. По определению информационная система содержит следующие группы объектов: информацию, базы данных, информационные технологии и технические средства.

Чтобы сопоставить выделенным объектам угрозы, влияющие на надежность, необходимо проанализировать факторы, определяющие возникновение угроз на объектах ИС, угрозы безопасности, в том числе описанные в банке данных угроз безопасности информации ФСТЭК России [1] и базовой модели угроз (рис. 1).

Рассмотрим подробнее методы обеспечения надежности объектов информационных систем.

Предупреждение ошибок – метод позволяющий, при его использовании, предотвратить появление ошибок в готовой программе. Основан на принципах, согласно которым выдвигаются требования к проектированию и программированию, гарантирующие использование программы в точном соответствии со спецификациями, что не дает возможность использовать программу не по назначению [3].

Обнаружение ошибок – это метод, в котором особое внимание уделяется функции самого программного средства незамедлительно выявлять ошибки и сбои. Незамедлительное обнаружение многократно уменьшает влияние ошибок на работу программы и сокращает время на поиск и исправление ошибок так как осуществляется автоматически, а не вручную.

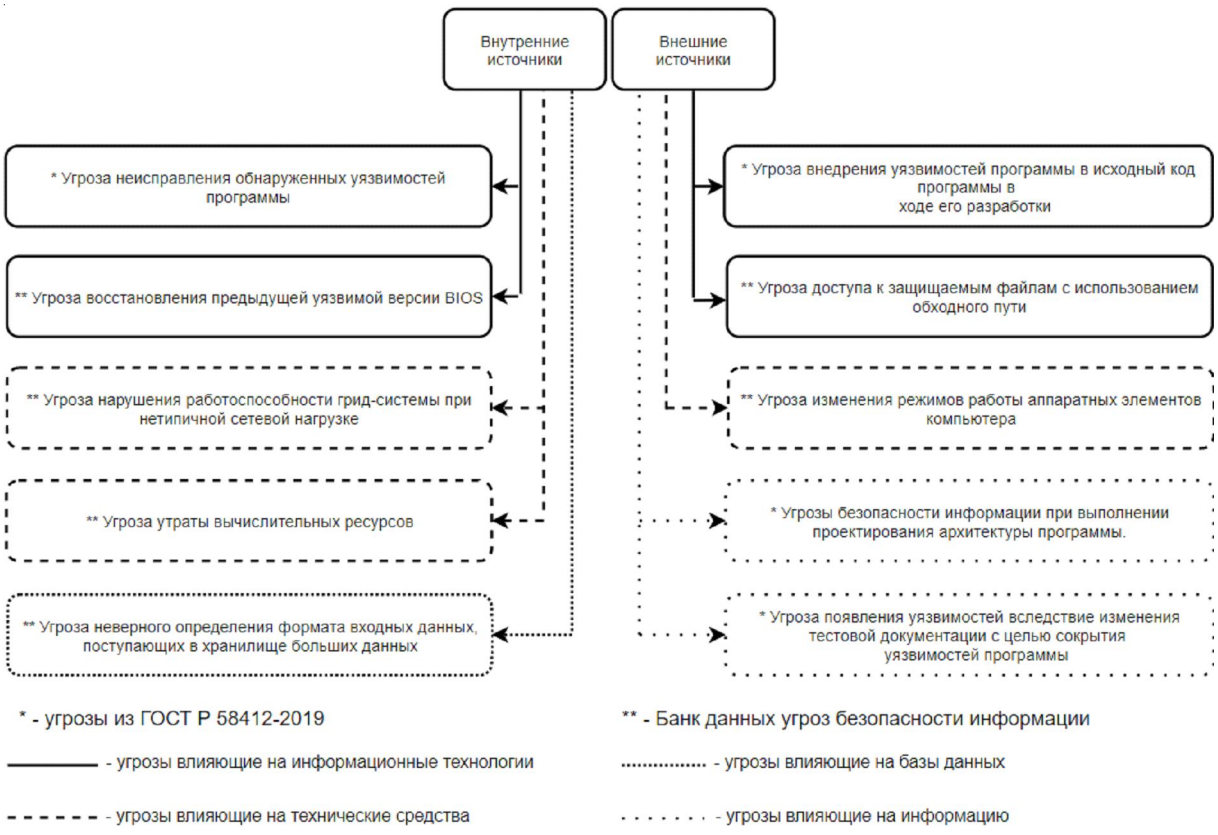

Рис. 1. Угрозы, влияющие на надежность объектов информационной системы

Исправление ошибок – следующий шаг после того, как ошибка была обнаружена. Обнаруженная ошибка и ее последствия должны быть исправлены автоматически самим программным средством, либо человеком вручную.

Как бы хорошо не была спроектирована программа и реализована в ней могут содержаться остаточные ошибки. Цель метода обеспечения устойчивости к ошибкам состоит в продолжении функционирования программной среды даже если в ней есть ошибки.

Определим критерии для оценки механизмов обеспечения надежности информационной системы и их возможные значения.

Критерий 1. Возможность восстановления исходного состояния системы. Данный критерий определяет возможно ли восстановить работоспособность системы путем отката – восстановлением предыдущего состояния системы. Может принимать значения: да; нет.

Критерий 2. Возможность выполнения резервирования. Данный критерий определяет предусмотрено ли в системе возможность резервирования и все ли технические средства способствуют этому. Может принимать значения: да; нет.

Критерий 3. Отказоустойчивость. Данный критерий определяет способна ли система продолжать полноценную работу при выходе из строя отдельных компонентов или программном сбое. Может принимать значения: нет; низкая; средняя; высокая.

Критерий 4. Тестирование. Критерий измерения качества, определения корректности и реальной надежности функционирования программ, систем на любых этапах разработки. Может принимать значения: да; нет.

Критерий 5. Сложность реализации. Данный критерий определяет степень сложности реализации выбранных методов на информационной системе. Может принимать значения: низкая; средняя; высокая.

Критерий 6. Возможность устранения ошибок на этапе проектирования системы: да; нет.

Проведем сравнительный анализ методов обеспечения надежности объектов информационных систем (см. таблицу).

Сравнительный анализ проводился при помощи расчета расстояния Евклида. Механизм обеспечения устойчивости к ошибкам получил наименьшую результирующую оценку, и следовательно, наилучшую оценку по результатам сравнения всех механизмов по всем критериям. Тем не менее, он является сложно реализуемым. Поэтому принято решение более подробно рассмотреть механизмы обнаружения и исправления ошибок, имеющих следующую по величине оценку.

Основной целью программного средства является реализация механизмов обнаружения и исправления ошибок в передаваемом сообщении. Достижением поставленной цели будет считаться соответствие предаваемого сообщения полученному в результате обнаружения ошибки и ее исправления. В реализации механизмов помог код Хэмминга.

В качестве входных данных будет число для передачи, а результатом работы программного средства – полученное число и информация о наличии ошибки.

Формализованная модель процесса обнаружения и исправления ошибки будет представлена ниже.

Код Хэмминга – корректирующий код, применяемый для контроля целостности пе-

Сравнительный анализ механизмов обеспечения надежности объектов ИС

Критерии	Механизмы обеспечения надежности объектов информационных систем			
	Предупреждение ошибок	Обнаружение ошибок	Исправление ошибок	Обеспечение устойчивости к ошибкам
К1	Нет (0)	Нет (0)	Нет (0)	Да (1)
К2	Да (1)	Нет (0)	Да (1)	Да (1)
К3	Нет (0)	Средняя (0,667)	Средняя (0,667)	Высокая (1)
К4	Да (1)	Да (1)	Да (1)	Да (1)
К5	Средняя (0,5)	Низкая (1)	Низкая (1)	Высокая (0)
К6	Да (1)	Да (1)	Нет (0)	Да (1)
Результат оценки метода	1,5	1,4	1,4	1

редачи данных [2]. Его корректирующая способность связана с таким понятием как – кодовое расстояние. Кодовым расстоянием между кодовыми комбинациями A и B является число позиций, в которых элементы этих комбинаций не совпадают. Расстояние между A и B, равно весу некой комбинации C, вес $W(C)$ кодовой комбинации C – количество содержащихся в ней двоичных единиц. Комбинация C получается путем поразрядного сложения комбинаций A и B в соответствии со следующей формулой:

$$W(C) = W(A \oplus B) = \sum (a_i \oplus b_i).$$

Данный код имеет такой значительный параметр как $d_{\min} = 3$ – минимальное расстояние Хэмминга. Введенный параметр определяет такую же важную характеристику, как корректирующую способность – максимальное количество исправлений:

$$t = \left[\frac{d_{\min} - 1}{2} \right],$$

то есть код Хэмминга способен гарантированно исправить только одну ошибку.

N-значный код Хэмминга имеет m-информационных разрядов и k-проверочных разрядов (контрольных бит). Число проверочных разрядов должно удовлетворять соотношению:

$$k >= \log_2 (n+1),$$

значения контрольных бит, формируются путем подсчета четности суммы единиц для определенных групп информационных разрядов.

Когда контрольным битам заданы значения, переходим ко второй комбинации и также расставляем в ней контрольные биты. Затем сравниваем получившиеся значения. Если есть отличия, то складываем номера позиций проверочных разрядов, в которых они отличаются, и тем самым получаем позицию, в которой произошла ошибка, затем конвертируем значение в полученной позиции – таким образом исправляя ошибку.

Архитектура программного средства состоит из следующих модулей:

1. Модуль ввода данных и их кодирования.
2. Модуль декодирования данных.
3. Модуль вывода результатов.
4. Пользовательский интерфейс.

Модуль ввода данных и их кодирования отвечает за кодирование введенных данных в двоичную систему счисления, а также за расставление в получившейся комбинации контрольных бит. Затем передает сообщение в закодированном виде в модуль декодирования данных.

В модуль декодирования данных поступает закодированное сообщение, в котором также расставляются контрольные биты. В этом же модуле происходит сравнение комбинаций с контрольными битами, полученными в модуле ввода и кодирования данных, с модулем декодирования данных. В зависимости от результата сравнения, комбинация, полученная в модуле декодирования данных – декодируется и передается в модуль вывода результатов.

Модуль вывода результатов отображает на пользовательском интерфейсе результат передачи сообщения и выводит сообщение об ошибках.

Программное средство состоит из двух пользовательских интерфейсов: пользовательского интерфейса для кодирования (см. рис. 2) и пользовательского интерфейса для декодирования данных (см. рис. 3).

Пользовательский интерфейс для кодирования данных состоит из: поля для ввода сообщения для передачи; кнопки, по нажатию которой открывается второй пользовательский интерфейс, а также заполняется поле для вывода закодированного сообщения и поле для вывода закодированного сообщения с контрольными битами.

Пользовательский интерфейс для декодирования данных состоит из: поля для полученного сообщения из пользовательского интерфейса для кодирования; блока для расставления контрольных бит в полученном сообщении; блока для декодирования.

Для проверки достижения поставленной цели была проведена серия из трех экспериментов. Данные для передачи в пользовательский интерфейс для декодирования во всех экспериментах будут одинаковые, они представлены на рисунке 4.

Задача первого эксперимента заключается в том, чтобы передать в пользовательский интерфейс для декодирования сообщение в закодированном виде без ошибок (см. рис. 5).

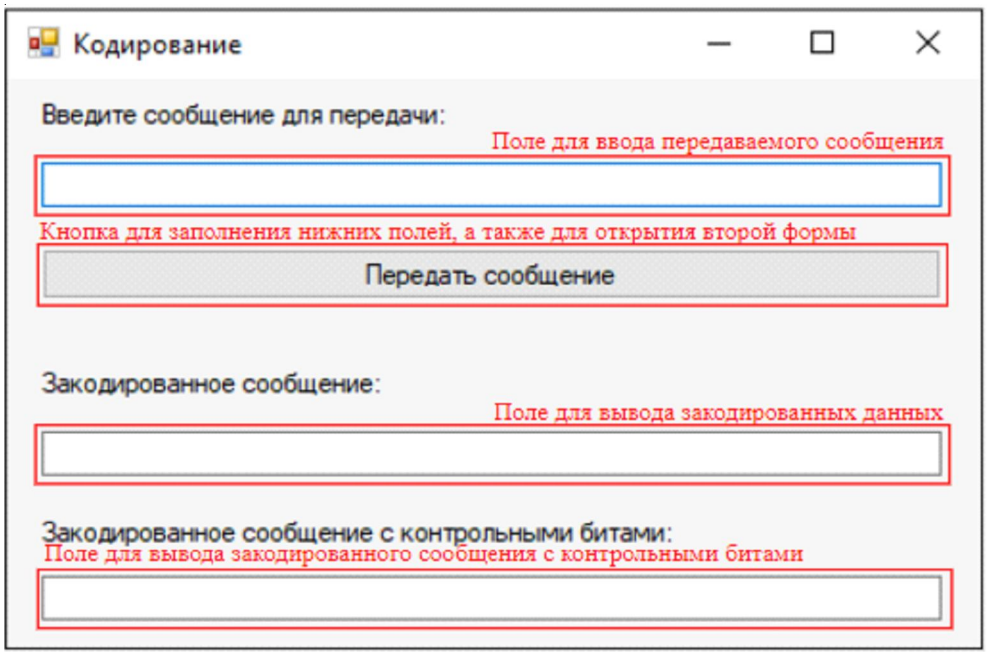

Рис. 2. Пользовательский интерфейс для кодирования данных

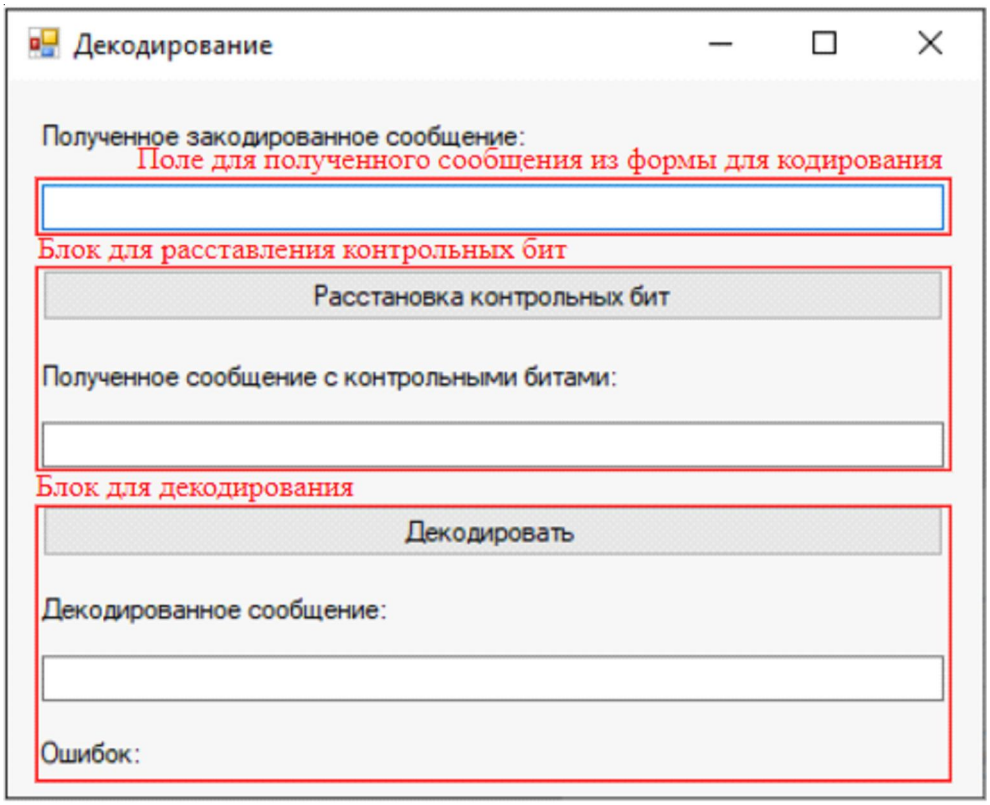

Рис. 3. Пользовательский интерфейс для декодирования данных

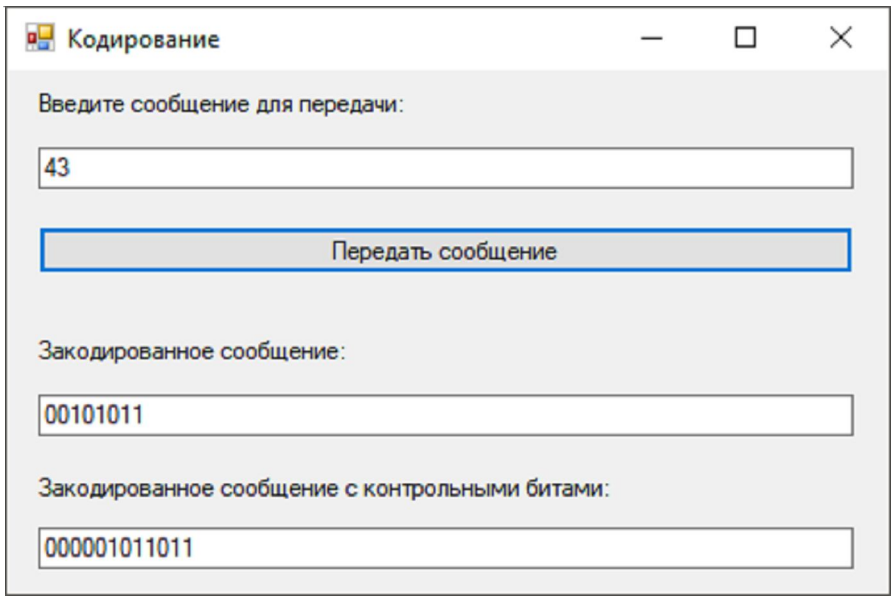

Рис. 4. Данные для передачи в пользовательский интерфейс

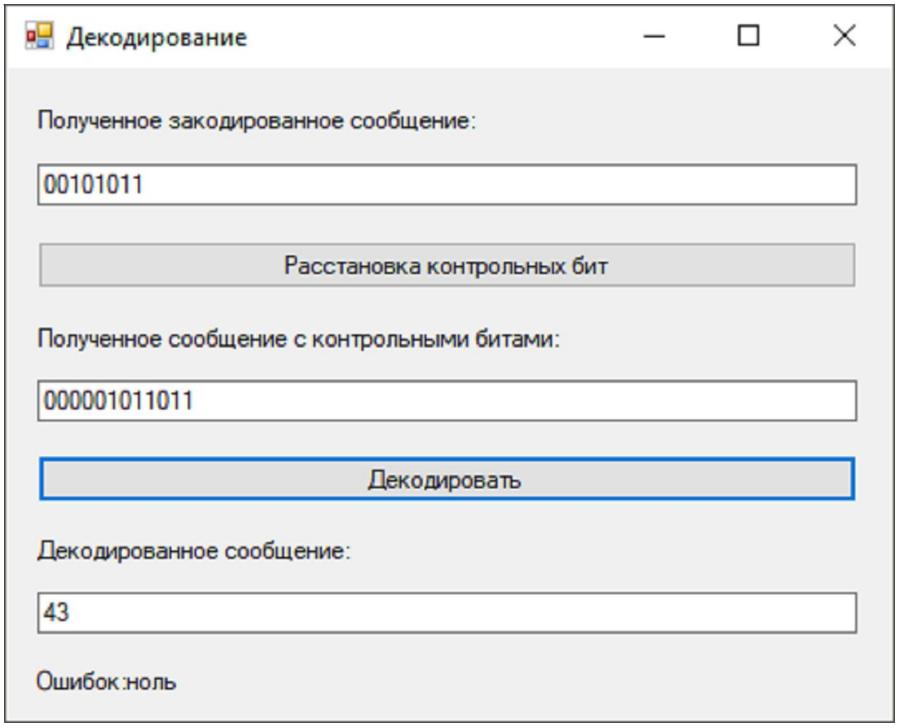

Рис. 5. Результат первого эксперимента

Задача второго эксперимента заключается в том, чтобы передать в пользовательский интерфейс для декодирования сообщение в закодированном виде с одной ошибкой (см. рис. 6).

Задача третьего эксперимента заключается в том, чтобы передать в пользовательский интерфейс для декодирования сообщение в закодированном виде более чем с одной ошибкой (см. рис. 7).

Анализ результатов экспериментальных исследований показал, что разработанное программное средство позволяет обнаружить и исправить ошибки в передаваемых данных.

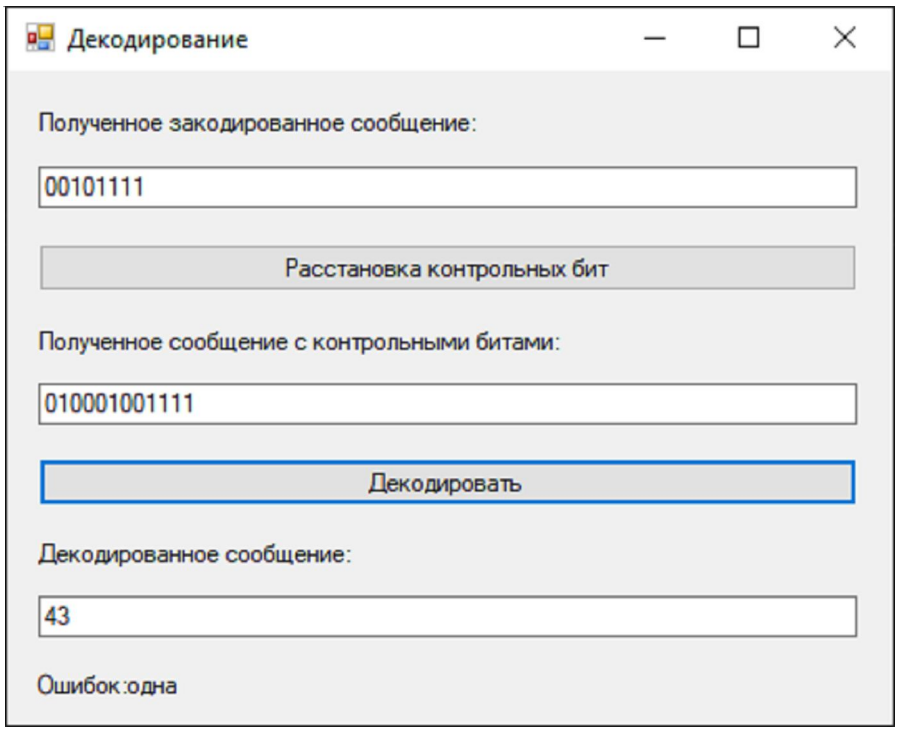

Рис. 6. Результат второго эксперимента

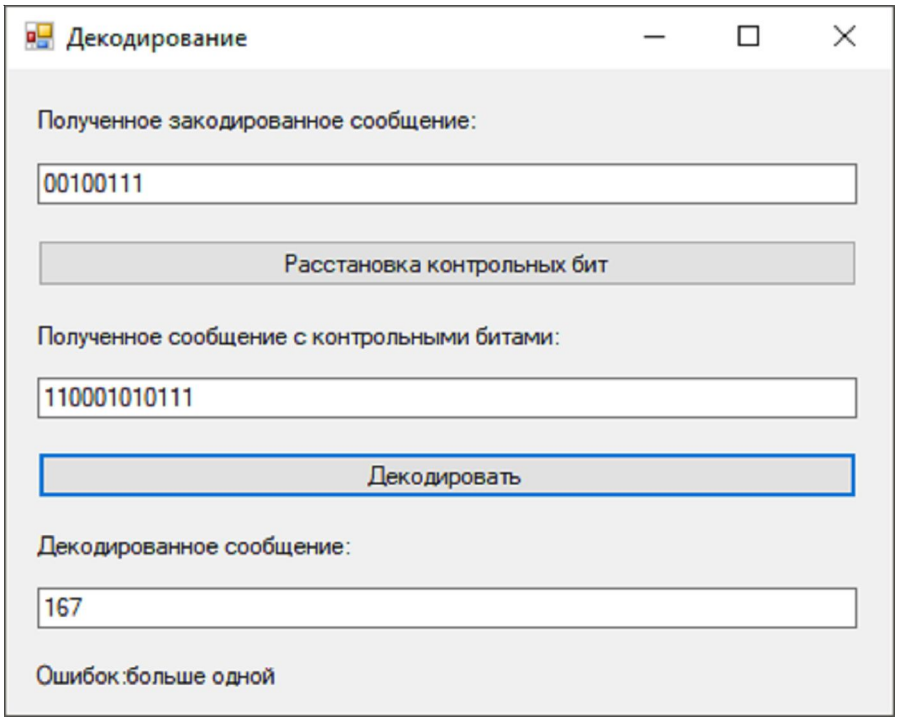

Рис. 7. Результат третьего эксперимента

СПИСОК ЛИТЕРАТУРЫ

1. Банк данных угроз безопасности информации : сайт. – Электрон. дан. – Режим доступа: https://bdu.fstec.ru/threat (дата обращения: 25.11.2021). – Загл. с экрана.

2. Баранова, Е. К. Методы и средства защиты компьютерной информации: методические указания к выполнению лабораторной работы / Е .К. Баранова. – М. : Российский государственный социальный университет, 2007. – 14 с.

3. Шубинский, И. Б. Методы обеспечения функциональной надежности программ / И. Б. Шубинский. – Электрон. текстовые дан. – Режим доступа: http://www.ibtrans.ru/upload/iblock/c44/c44620b0cb7906c2abb378a9ae9e7d6c.pdf (дата обращения: 25.11.2021). – Загл. с экрана.

4. Шубинский, И. Б. Структурная надежность информационных систем: методы анализа / И. Б. Шубинский. – М. : Журнал «Надежность», 2012. – 216 с.

REFERENCES

1. *Bank dannyh ugroz bezopasnosti informacii: sait* [Information Security Threat Data Bank. Website]. URL: https://bdu.fstec.ru/threat (accessed 25 November 2021).

2. Baranova E.K. *Metody i sredstva zashhity komp'juternoj informacii: metodicheskie ukazanija k vypolneniju laboratornoj raboty* [Methods and Means of Computer Information Protection: Methodological Instructions for Laboratory Work]. Moscow, Rossiyskiy gosudarstvennyy sotsial'nyy universitet, 2007. 14 p.

3. Shubinskij I.B. *Metody obespechenija funkcional'noj nadezhnosti program* [Methods to Ensure Functional Reliability of Programs]. URL: http://www.ibtrans.ru/upload/iblock/c44/c44620b0cb7906c2abb378a9ae9e7d6c.pdf (accessed 25 November 2021).

4. Shubinskij I.B. *Strukturnaja nadezhnost' informacionnyh sistem: metody analiza* [Structural Reliability of Information Systems. Methods of Analysis]. Moscow, Zhurnal «Nadezhnost'», 2012. 216 p.

MECHANISM FOR ENSURING THE RELIABILITY OF INFORMATION SYSTEM OBJECTS

Elizaveta E. Bandurova

Student, Department of Information Security,
Volgograd State University
ibb-181_837244@volsu.ru
Prosp. Universitetsky, 100, 400062 Volgograd, Russian Federation

Tatiana A. Omelchenko

Senior Lecturer, Department of Information Security,
Volgograd State University
omelchenko.tatiana@volsu.ru
Prosp. Universitetsky, 100, 400062 Volgograd, Russian Federation

Abstract. It is difficult to imagine the functioning of modern society without information systems of various levels of complexity. Quality control of the functioning of systems is one of the main tasks that should be paid close attention to. One of the criteria for the quality of an information system is reliability, the better the system is, the more reliable it is, and vice versa. In this regard, the purpose of the study is to choose the most appropriate mechanism for ensuring the reliability of information systems. This article discusses the main mechanisms for ensuring the reliability of information system objects such as: error prevention, error detection, error correction, error tolerance; highlighted criteria for their evaluation such as: the ability to restore the initial state of the system, the ability to perform redundancy, fault tolerance, testing, complexity of implementation, the ability to eliminate errors at the stage of system design. The factors determining the occurrence of threats to the objects of information systems and security threats are analyzed. A comparative analysis of the mechanisms for ensuring the reliability of information system objects according to the selected criteria and analyzed factors is carried out. The formalized model, architecture and interface of the software are presented. A series of experimental studies has been carried out

Key words: reliability, information system, software tool, error prevention, error detection, error correction, error tolerance.