ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ГОСТ Р 52633.4— 2011**

## Защита информации

# ТЕХНИКА ЗАЩИТЫ ИНФОРМАЦИИ

## Интерфейсы взаимодействия с нейросетевыми преобразователями биометрия — код доступа

Издание официальное

Москва
Стандартинформ
2012

## Предисловие

Цели и принципы стандартизации в Российской Федерации установлены Федеральным законом от 27 декабря 2002 г. № 184-ФЗ «О техническом регулировании», а правила применения национальных стандартов Российской Федерации — ГОСТ Р 1.0—2004 «Стандартизация в Российской Федерации. Основные положения».

**Сведения о стандарте**

1 РАЗРАБОТАН Федеральным государственным учреждением «Государственный научно-исследовательский испытательный институт проблем технической защиты информации Федеральной службы по техническому и экспортному контролю» (ФГУ «ГНИИИ ПТЗИ ФСТЭК России»), Федеральным государственным унитарным предприятием «Пензенский научно-исследовательский электротехнический институт» (ФГУП «ПНИЭИ»)

2 ВНЕСЕН Техническим комитетом по стандартизации «Защита информации» ТК 362

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 1 декабря 2011 г. № 682-ст

4 ВВЕДЕН ВПЕРВЫЕ

*Информация об изменениях к настоящему стандарту публикуется в ежегодно издаваемом информационном указателе «Национальные стандарты», а текст изменений и поправок — в ежемесячно издаваемых информационных указателях «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячно издаваемом информационном указателе «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет*

## Содержание

## Введение

Разработка настоящего стандарта связана с необходимостью спецификации взаимодействия биометрических систем с нейросетевыми преобразователями биометрия — код доступа, выполненными с учетом требований ГОСТ Р 52633.0.

Настоящий стандарт освещает три стороны взаимодействия биометрических систем с нейросетевыми преобразователями биометрия — код доступа:

1) специфицирует интерфейс взаимодействия с нейросетевыми преобразователями биометрия — код доступа различных производителей, использующих различные биометрические технологии;

2) указывает требования к параметрам биометрических образов, используемых для обучения, тестирования, извлечения выходного кода нейросетевых преобразователей биометрия — код доступа;

3) определяет формат представления нейросетевых биометрических контейнеров, используемых для обеспечения анонимности, конфиденциальности и обезличенности размещенных в них биометрических параметров.

Настоящий стандарт представляет собой спецификацию архитектуры (далее — модель) программного интерфейса нейросетевых преобразователей биометрия — код доступа и определяет модель нейросетевого преобразования биометрических данных человека в выходной код, пригодную для использования в любой биометрической технологии.

Настоящий стандарт является аналогом [1], но ориентирован на системы, для которых требование обеспечения конфиденциальности, анонимности и обезличенности персональных биометрических данных пользователя является обязательным.

В разделе 5 определена модель и архитектура нейросетевого преобразователя биометрия — код доступа.

В разделе 6 установлены требования к представлению биометрических параметров и других элементов нейросетевого преобразования биометрия — код доступа.

В разделе 7 определены модель и типовые схемы взаимодействия биометрического приложения с нейросетевым преобразователем биометрия — код доступа.

В разделе 8 определены основные макросы, типы данных и константы, используемые нейросетевыми преобразователями биометрия — код доступа.

В разделе 9 определены вызовы функций программных интерфейсов, инициируемые биометрическим приложением и обрабатываемые нейросетевыми преобразователями биометрия-код.

В разделе 10 установлены правила обработки биометрическим приложением событий, возникающих при работе с нейросетевым преобразователем биометрия — код доступа, входные и выходные параметры функции обработки событий в зависимости от типа события.

В разделе 11 установлены правила обработки ошибок, общие коды ошибок и коды ошибок программных интерфейсов компонентов.

В приложении А приведены форматы сетевого представления основных типов данных.

В приложении Б приведены типовые схемы организации параметров элементарных преобразователей.

В приложении В приведены требования к формату представления параметров нейросетевого преобразователя биометрия — код доступа.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Защита информации**

**ТЕХНИКА ЗАЩИТЫ ИНФОРМАЦИИ**

# Интерфейсы взаимодействия с нейросетевыми преобразователями биометрия — код доступа

Information protection. Information protection technology. Neural network biometric-code converters programming interfaces

**Дата введения — 2012—09—01**

## 1 Область применения

Настоящий стандарт устанавливает требования к программному интерфейсу взаимодействия с универсальным нейросетевым преобразователем биометрия — код доступа в качестве стандартного интерфейса биометрической системы, применяющей нейросетевое преобразование биометрия — код доступа (далее — нейросетевое преобразование) для выработки двоичного кода из персональных биометрических данных пользователя и использующей этот двоичный код для решения комплексных задач идентификации/аутентификации пользователей, защиты информации, ограничения доступа к ресурсам вычислительных систем.

Требования настоящего стандарта применяют к биометрическим системам, использующим различные биометрические технологии (динамика подсознательных движений, анализ отпечатка пальца, анализ особенностей голоса и других), имеющим разный масштаб развертывания (от персональных устройств идентификации/аутентификации и локальных систем обеспечения сетевой безопасности до дистанционных комплексных систем идентификации/аутентификации), произвольную архитектуру и вычислительную платформу.

Положения настоящего стандарта применяют к биометрическим системам, декларирующим обеспечение конфиденциальности, анонимности и обезличенности персональных биометрических данных пользователя, размещенных в нейросетевых биометрических контейнерах.

Настоящий стандарт устанавливает модель нейросетевого преобразователя биометрия — код доступа, в которой возможно:

- использование биометрическим приложением реализаций нейросетевого преобразователя биометрия — код доступа, выполненных различными изготовителями;
- динамическое подключение и отключение одного и более модулей нейросетевых преобразователей биометрия — код доступа;
- использование широкого набора универсальных правил проведения нейросетевого преобразования биометрия — код доступа;
- гибкий выбор формата представления входных и выходных данных преобразователя;
- использование встроенных средств тестирования качества обучения нейросетевого преобразователя биометрия — код доступа;
- безопасное удаление биометрических данных пользователя и выходного кода после настройки параметров нейросетевого преобразования биометрия — код доступа;
- экспорт и импорт параметров нейросетевого преобразователя биометрия — код доступа в виде нейросетевого биометрического контейнера.

**Издание официальное**

## 2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие стандарты:

ГОСТ Р 52633.0—2006 Защита информации. Техника защиты информации. Требования к средствам высоконадежной биометрической аутентификации.

ГОСТ Р 52633.1—2009 Защита информации. Требования к формированию баз естественных биометрических образов, предназначенных для тестирования средств высоконадежной биометрической аутентификации.

ГОСТ Р ИСО/МЭК 8824-1—2001 Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 1. Спецификация основной нотации.

ГОСТ Р ИСО/МЭК 8824-4—2003 Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 4. Параметризация спецификации АСН.1.

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодно издаваемому информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, или по соответствующим ежемесячно издаваемым информационным указателям, опубликованным в текущем году. Если ссылочный стандарт заменен (изменен), то при пользовании настоящим стандартом следует руководствоваться заменяющим (измененным) стандартом. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, применяется в части, не затрагивающей эту ссылку.

## 3 Термины и определения

В настоящем стандарте применены термины с соответствующими определениями:

3.1 **биометрические данные:** Данные с выходов первичных измерительных преобразователей физических величин, совокупность которых образует биометрический образ конкретного человека.
[ГОСТ Р 52633.0—2006, статья 3.5]

3.2 **биометрический образ:** Образ человека, полученный с выходов первичных измерительных преобразователей физических величин, подвергающийся далее масштабированию и иной первичной обработке с целью извлечения из него контролируемых биометрических параметров человека.
[ГОСТ Р 52633.0—2006, статья 3.7]

3.3 **биометрический образ «Свой»:** Биометрический образ легального пользователя.
[ГОСТ Р 52633.0—2006, статья 3.8]

3.4 **биометрический образ «Чужой»:** Биометрический образ злоумышленника, пытающегося преодолеть биометрическую защиту.
[ГОСТ Р 52633.0—2006, статья 3.9]

3.5 **биометрические образы «Все чужие»:** Совокупность множества биометрических образов «Чужой», верно отражающая статистику попыток подбора злоумышленниками образов «Свой».
[ГОСТ Р 52633.0—2006, статья 3.10]

3.6 **биометрические параметры:** Параметры, полученные после предварительной обработки биометрических данных.
[ГОСТ Р 52633.0—2006, статья 3.11]

3.7 **вероятность ошибки первого рода:** Вероятность ошибочного отказа «Своему» пользователю в биометрической аутентификации.
[ГОСТ Р 52633.0—2006, статья 3.12]

3.8 **вероятность ошибки второго рода:** Вероятность ошибочной аутентификации «Чужого» как «Своего» (ошибочная аутентификация).
[ГОСТ Р 52633.0—2006, статья 3.13]

3.9 **преобразователь биометрия-код:** Преобразователь, способный преобразовывать вектор нечетких, неоднозначных биометрических параметров «Свой» в четкий однозначный код ключа (пароля). Преобразователь, откликающийся случайным выходным кодом на воздействие случайного входного вектора, не принадлежащего множеству образов «Свой».

[ГОСТ Р 52633.0—2006, статья 3.18]

3.10 **нейросетевой преобразователь биометрия — код доступа;** НПБК: Преобразователь биометрия-код, использующий в своей работе искусственные нейронные сети или другие аналогичные им модели.

3.11 **дискретный биометрический параметр:** Биометрический параметр, значения которого составляют конечное множество.

3.12 **непрерывный биометрический параметр:** Биометрический параметр, значения которого составляют континуальное множество.

3.13 **вектор биометрических параметров;** ВБП: Нумерованный набор биометрических параметров или производных от них параметров, имеющих одну и ту же интерпретацию и формат представления.

3.14 **выходной код:** Код, получаемый на выходе преобразователя биометрия-код в качестве результата.

3.15 **модуль преобразователя биометрия-код:** Структурная единица вычислительной системы, отвечающая за хранение компонентов ПБК и обеспечение доступа к ним.

3.16 **компонент преобразователя биометрия-код:** Функциональная единица вычислительной системы, отвечающая за реализацию ПБК и предоставление доступа к его функциональности.

3.17 **программный интерфейс модуля;** ПИМ: Специальным образом объявленный набор функций, позволяющий организовать взаимодействие с модулем ПБК.

3.18 **программный интерфейс компонента;** ПИК: Специальным образом объявленный и структурированный набор функций, позволяющий организовать взаимодействие с компонентом ПБК.

3.19 **элементарный преобразователь:** Элемент преобразователя биометрия-код, предназначенный для выполнения узкоспециализированных операций (элементарных преобразований) над векторами биометрических параметров.

3.20 **схема преобразования:** Определенный порядок выполнения элементарных преобразований над векторами биометрических параметров.

3.21 **описатель программного интерфейса компонента;** ОПИК: Значение, однозначно определяющее ПИК конкретного экземпляра компонента ПБК при исполнении биометрического приложения.

3.22 **нейросетевой биометрический контейнер;** НБК: Структурированный блок данных, содержащий параметры обученного НПБК.

3.23 **защищенный нейросетевой биометрический контейнер:** Нейросетевой биометрический контейнер, в котором некоторые части скрыты от непосредственного изучения путем использования обратимого или необратимого преобразования.

3.24 **жесткая индикация соответствия биометрического образа:** Процедура бескомпроматного сравнения выходного кода (или некоторого промежуточного значения) с эталонным значением на предмет их полного совпадения.

3.25 **мягкая индикация соответствия биометрического образа:** Процедура бескомпроматного сравнения выходного кода (или некоторого промежуточного значения) с эталонным значением, оценивающая их различие.

3.26 **связывание нейросетевых биометрических контейнеров:** Процедура установления связи между параметрами нескольких НПБК посредством некоторого выходного кода.

3.27 **трансформация кода:** Процедура бескомпроматного расширения или сжатия выходного кода для приведения его к требуемому формату.

## 4 Обозначения и сокращения

В настоящем документе приняты следующие обозначения и сокращения:

ВБП — вектор биометрических параметров;

ПБК — преобразователь биометрия-код;

НПБК — нейросетевой преобразователь биометрия — код доступа;
ЭП — элементарный преобразователь;
НБК — нейросетевой биометрический контейнер;
ПИМ — программный интерфейс модуля;
ПИК — программный интерфейс компонента;
ОПИК — описатель программного интерфейса компонента;
УУИД — универсальный уникальный идентификатор;
ИНС — искусственная нейронная сеть;
АСН.1 — абстрактная синтаксическая нотация версии 1.0;
ОДЗ — область допустимых значений.

## 5 Архитектура нейросетевого преобразователя биометрия — код доступа

### 5.1 Нейросетевое преобразование биометрия — код доступа

5.1.1 Нейросетевое преобразование биометрия — код доступа описывается функцией $F_{НПБК}$ по формуле (1.1), которая строится согласно схеме преобразования $S$ из элементарных преобразований $F_{ЭП}$, описываемых по формуле (1.2):

$$(X_{код}, X_{спец.\ вых}) = F_{НПБК}(X_{вх.\ био}, X_{спец.\ вх}, B_{ЭП}, S), \quad (1.1)$$

$$(X_{вых}, X_{спец.\ вых}) = F_{ЭП}(X_{вх}, X_{спец.\ вх}, B_{ЭП}), \quad (1.2)$$

где $X_{вх.\ био}$ — входные биометрические параметры;
$X_{спец.\ вх}$ — специальные входные параметры;
$B_{ЭП}$ — конфигурации всех элементарных преобразователей;
$S$ — схема преобразования;
$X_{код}$ — выходной код;
$X_{спец.\ вых}$ — специальные выходные параметры;
$X_{вх}$ — входные параметры;
$B_{ЭП}$ — конфигурация элементарного преобразователя;
$X_{вых}$ — выходной параметр.

5.1.2 Нейросетевое преобразование биометрия — код доступа (далее — нейросетевое преобразование) реализуется НПБК.

5.1.3 Элементарное преобразование реализуется ЭП НПБК.

5.1.4 Входные биометрические параметры определяют примеры биометрических образов «Свой» и «Чужой» на входе НПБК.

5.1.5 Специальные входные параметры определяют дополнительные параметры нейросетевого преобразования или элементарного преобразования.

5.1.6 Конфигурация ЭП содержит параметры работы отдельного ЭП, настраивается во время обучения НПБК и хранится в НБК. Конфигурация ЭП должна быть определена для каждого ЭП, указанного в схеме преобразования.

5.1.7 Схема преобразования устанавливает последовательность вызовов элементарных преобразований, число параметров элементарных преобразований и их метаописание. Формат представления схемы преобразования должен соответствовать 6.1.

5.1.8 Выходной код определяет основной результат выполнения нейросетевого преобразования. Выходной код для биометрического образа «Свой» должен быть равен выходному коду, заданному во время обучения НПБК.

5.1.9 Специальные выходные параметры определяют дополнительные результаты выполнения нейросетевого преобразования или элементарного преобразования.

5.1.10 Входные параметры определяют основные обрабатываемые ЭП данные.

5.1.11 Выходной параметр определяет основной результат выполнения элементарного преобразования.

5.1.12 Входные, специальные входные, специальные выходные, выходные параметры нейросетевого преобразования представляются в формате матриц ВБП по 6.3.1.1.

### 5.2 Структура нейросетевого преобразователя биометрия — код доступа и его архитектурная модель

5.2.1 НПБК реализуют в виде компонента ПБК модуля ПБК системы.

Примечание — В зависимости от операционной системы и аппаратной платформы возможна реализация модуля НПБК как статической или динамической библиотеки, а также встраивание модуля НПБК непосредственно в операционную систему.

5.2.2 Компонент НПБК должен реализовывать функции ПИК НПБК в соответствии с требованиями раздела 7.

5.2.3 Модуль НПБК должен реализовывать функции ПИМ НПБК в соответствии с требованиями раздела 7.

5.2.4 Имя модуля НПБК, а также его место размещения выбирают произвольно.

Примечание — Правила передачи модуля НПБК в настоящем стандарте не специфицируются. В качестве варианта может рассматриваться передача модуля непосредственно от производителя НПБК разработчику биометрического приложения по договоренности.

## 6 Представление параметров нейросетевого преобразования

### 6.1 Представление схемы преобразования

6.1.1 Схема преобразования должна состоять из последовательности слотов четырех типов:

1) слот входного параметра, описывающий входной параметр НПБК;

2) слот специального входного параметра, описывающий специальный входной параметр НПБК;

3) слот специального выходного параметра, описывающий специальный выходной параметр НПБК;

4) слот ЭП, описывающий ЭП и выходной параметр ЭП.

Слоты первых трех типов должны содержать поля: «Номер слота» по 8.2.13, «Метаописание» по 8.2.22. Слот ЭП должен дополнительно содержать поля: «Флаги ЭП» по 8.2.1, «Состояние ЭП» по 8.2.4, «Число связанных слотов», «Список номеров связанных слотов».

6.1.2 Поле «Номер слота» задает уникальное в пределах одной схемы преобразования число, идентифицирующее слот. При заполнении поля выбирают в подполе «Индекс» поля «Номер слота» значение, наименьшее из возможных. Подполе «Тип» заполняют значением, соответствующим типу слота.

6.1.3 Поле «Метаописание» определяет число, тип, формат представления параметра, связанного со слотом. Поле заполняют по 6.3 и в соответствии с приложением Б.

6.1.4 Поля «Флаги ЭП», «Состояние ЭП» заполняют по 8.2.1, 8.2.3.

6.1.5 Поле «Число связанных слотов» содержит число связанных с ЭП слотов. Значение 0xFF используют, если ЭП имеет более 254 связанных слотов.

6.1.6 Поле «Список номеров связанных слотов» содержит номера слотов, параметры которых используются как входные, специальные входные или специальные выходные параметры для ЭП. При этом подполе «Тип» номера слота в поле изменяют в соответствии с типом параметра ЭП. Запрещается использовать в поле значений номера слотов, ранее не определенных в схеме преобразования.

6.1.7 Формат представления схемы преобразования в непрерывном блоке памяти определяют в соответствии с А.3 (приложение А).

6.1.8 Должен соблюдаться следующий порядок объявления слотов в схеме преобразования:

- слот, содержащий значения первых двух полей слота ЭП, выходной параметр которого будет интерпретироваться как выходной код;
- слоты входных параметров;
- слоты специальных входных/выходных параметров, слоты ЭП.

### 6.2 Представление биометрического параметра

#### 6.2.1 Общие положения

6.2.1.1 Представление биометрического параметра, получаемого из биометрического образа, и его интерпретация определяются форматом по 6.2.2 и типом по 6.2.3—6.2.6: непрерывный, дискретный, неизменяемый по области «Свой» непрерывный, неизменяемый по области «Свой» дискретный.

Примечание — Поддержка всех форматов и типов биометрических параметров не является обязательной. Поэтому должна осуществляться проверка поддержки конкретных форматов и типов на этапе перечисления поддерживаемых НПБК схем преобразования или на этапе установки схемы преобразования в НПБК.

6.2.1.2 Допустимо для каждого биометрического параметра дополнительно устанавливать ограничения по 8.2.5, определяющие особенности представления и интерпретации биометрического параметра.

Примечание — Поддержка всех типов ограничений, кроме nbCT_DATA, nbCT_DATA_OWN, nbCT_DATA_ALL, является необязательной.

**6.2.2 Формат биометрического параметра**

6.2.2.1 Формат биометрического параметра определяет его представление в памяти и правила кодирования множества его значений.

6.2.2.2 Допустимо кодировать биометрический параметр как целое, неотрицательное целое, вещественное число различной разрядности, а также как целое длиной 1, 2, 4 бита со знаком или без знака по 8.2.23.

6.2.2.3 Несколько биометрических параметров одного формата и типа организуют в ВБП по 8.2.20.

6.2.2.4 Размер ВБП в битах рассчитывают как произведение числа элементов на длину формата одного элемента в битах. Размер ВБП в байтах рассчитывают делением размера ВБП в битах на восемь и округлением до целого в большую сторону.

6.2.2.5 При выделении памяти для ВБП заполнители и промежутки между отдельными элементами ВБП не допускаются.

**6.2.3 Непрерывный биометрический параметр**

6.2.3.1 Непрерывным считается биометрический параметр, значения которого составляют континуальное множество и ограничены только точностью представления.

Примечание — Примерами непрерывного биометрического параметра могут служить: значение угла поворота отпечатка пальца, значение изменения давления электронного пера на поверхность ввода, координаты особенных точек лица человека.

6.2.3.2 Допустимо использовать любой формат представления непрерывного биометрического параметра.

Примечание — Для повышения совместимости НПБК рекомендуется поддерживать формат представления непрерывных биометрических параметров nbMF_R32.

6.2.3.3 Для непрерывных биометрических параметров допустимо устанавливать следующие ограничения: значения «Чужой», «Свой» и «Все чужие», ОДЗ, стандартное отклонение распределений значений «Все чужие» и «Свой», шаг дискретизации, законы распределения значений «Все чужие» и «Свои», смещение. Пример распределения значений «Все чужие» и «Свой» непрерывного биометрического параметра приведен на рисунке 1.

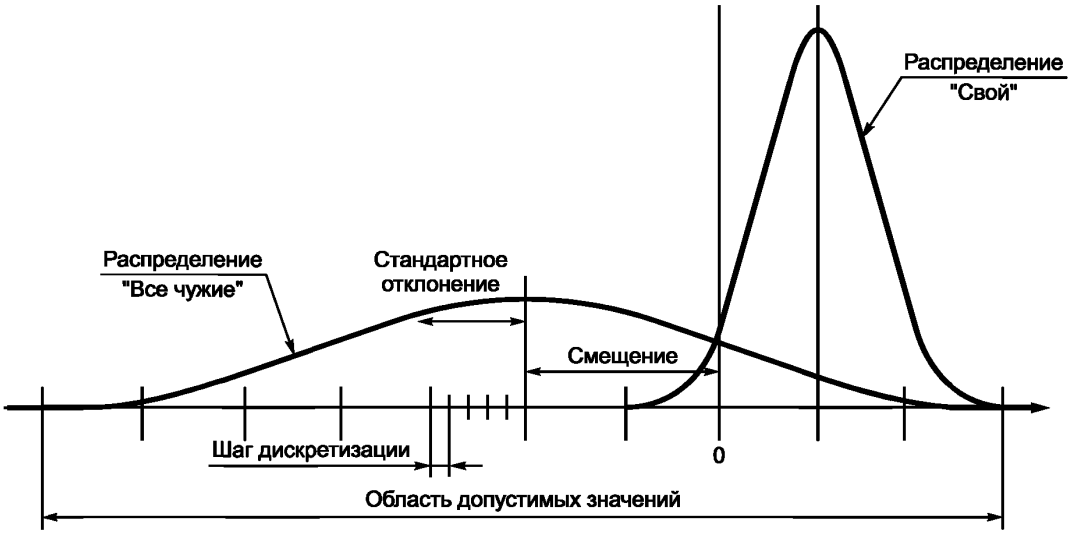


Рисунок 1 — Распределения «Все чужие» и «Свой» в области допустимых значений непрерывного биометрического параметра

6.2.3.4 ОДЗ по умолчанию должна быть ограничена форматом представления непрерывного биометрического параметра.

Примечание — В случае выхода в ходе вычислений значений непрерывного биометрического параметра за границы ОДЗ в качестве значения используется ближайшая граница ОДЗ.

6.2.3.5 ОДЗ должна определять границы множества фактически известных значений непрерывного биометрического параметра.

6.2.3.6 Центр распределения значений «Все чужие» должен совпадать с центром ОДЗ.

Примечания

1 Если закон распределения значений биометрических параметров «Все чужие» отличается от нормального, перед использованием следует центрировать значения так, чтобы минимизировать вероятность возникновения выхода значений за границы ОДЗ.

2 Рекомендуется выбирать ОДЗ так, чтобы его центр совпадал с 0.

6.2.3.7 Диапазон значений ОДЗ непрерывного биометрического параметра должен включать не меньше шести стандартных отклонений распределения «Все чужие».

**6.2.4 Дискретный биометрический параметр**

6.2.4.1 Дискретным считается такой биометрический параметр, значения которого составляют конечное множество.

Примечание — Примерами дискретного биометрического параметра могут служить: цвет глаз человека (карий, голубой, зеленый, серый и т. д.), тип контрольной точки отпечатка пальца (окончание гребня, бифуркация гребня, другая контрольная точка).

6.2.4.2 Разрешается использовать только целочисленные форматы представления дискретного биометрического параметра.

Примечание — Для повышения совместимости НПБК рекомендуется использовать формат представления дискретных биометрических параметров nbMF_I1.

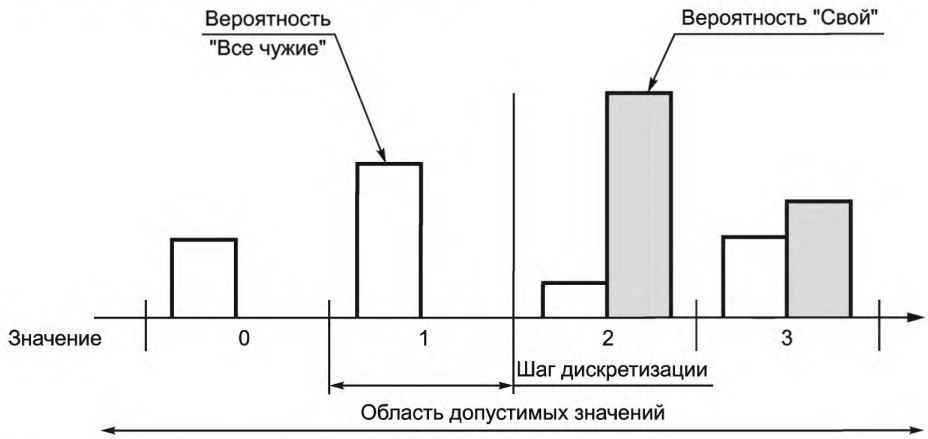


Рисунок 2 — Распределения «Все чужие» и «Свой» в области допустимых значений дискретного биометрического параметра

6.2.4.3 Для дискретного биометрического параметра допустимо устанавливать следующие ограничения: значения «Чужой», «Свой» и «Все чужие», ОДЗ, законы распределения значений «Все чужие» и «Свои», шаг дискретизации. Пример распределения значений «Все чужие» и «Свой» дискретного биометрического параметра приведен на рисунке 2.

6.2.4.4 ОДЗ по умолчанию должна быть ограничена форматом представления дискретного биометрического параметра.

6.2.4.5 Диапазон значений ОДЗ дискретного биометрического параметра должен включать диапазон наиболее вероятных значений распределения «Все чужие».

**6.2.5 Неизменяемый в области «Свой» непрерывный биометрический параметр**

6.2.5.1 Для неизменяемого в области «Свой» непрерывного биометрического параметра множество значений «Свой» должно содержать одно значение.

Примечание — В этом случае распределение «Свой» будет вырожденным со стандартным отклонением, равным 0.

6.2.5.2 Требования к представлению неизменяемого в области «Свой» непрерывного биометрического параметра определяют по 6.2.3.

**6.2.6 Неизменяемый в области «Свой» дискретный биометрический параметр**

6.2.6.1 Для неизменяемого в области «Свой» дискретного биометрического параметра множество значений «Свой» должно содержать одно значение.

6.2.6.2 Требования к представлению неизменяемого в области «Свой» дискретного биометрического параметра определяют по 6.2.4.

### 6.3 Представление параметров в виде матрицы векторов биометрических параметров

6.3.1 Параметры элементарных преобразований и нейросетевого преобразования (входные, выходные, специальные входные, специальные выходные) представляют в виде матриц ВБП согласно В.1 (приложение В).

6.3.2 При объединении нескольких параметров каждый параметр набора представляют по 6.3.1, а в поле nrows матрицы ВБП заносят общее число параметров.

6.3.3 Для представления ограничений используют представление данных в виде матриц ВБП согласно В.2 (приложение В).

6.3.4 Для представления результатов тестирования используют представление данных в виде матрицы ВБП согласно В.3 (приложение В).

6.3.5 Для представления параметров функций ПИК НПБК используют формат матриц ВБП, указанный в описании соответствующей функции.

6.3.6 Матрица ВБП шаблона параметра в общем случае должна содержать те же значения полей, что и матрица ВБП параметра. Но допускается устанавливать поле ncols матрицы ВБП шаблона параметра в 0, а поле data — в nbNULL.

6.3.7 Допускается использовать формат представления отдельных параметров, отличный от рекомендуемого, при реализации НПБК на устройствах с ограниченными вычислительными возможностями.

### 6.4 Представление нейросетевого биометрического контейнера и его блоков

6.4.1 НБК представляют в виде последовательности блоков НБК в соответствии с правилами А.5 (приложение А).

6.4.2 Блок заголовка НБК (nbBT_NBC) размещают первым.

6.4.3 Блок схемы преобразования (nbBT_CSCHEME) размещают перед блоками ЭП, соответствующими слотам ЭП в схеме преобразования.

6.4.4 Блоки ЭП, используемые для хранения конфигураций ЭП, размещают в соответствии с порядком определения соответствующих слотов ЭП в схеме преобразования.

6.4.5 Информационные блоки, используемые разработчиком биометрического приложения для хранения вспомогательной информации, размещают в конце НБК.

6.4.6 При анализе НБК информационные блоки неизвестного типа пропускают.

## 7 Программные интерфейсы взаимодействия биометрического приложения с нейросетевым биометрическим преобразователем биометрия — код доступа

### 7.1 Модель взаимодействия нейросетевого преобразователя с биометрическим приложением

7.1.1 Модель взаимодействия биометрического приложения и НПБК приведена на рисунке 3.

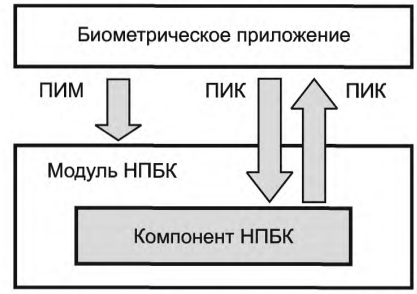


Рисунок 3 — Модель взаимодействия ПИМ/ПИК НПБК

7.1.2 ПИМ определяет интерфейс взаимодействия модуля ПБК с биометрическим приложением. ПИМ реализуют по 7.2.

7.1.3 ПИК определяет интерфейс взаимодействия компонента ПБК с биометрическим приложением. ПИК реализуют по 7.3.

7.1.4 Модуль ПБК должен предоставлять доступ к размещенным в нем компонентам ПБК.

7.1.5 Компонент ПБК должен предоставлять доступ к поддерживаемым им ПИМ.

7.1.6 Каждому компоненту ПБК присваивают УУИД компонента. УУИД выбирают произвольно.

Примечание — Уникальный идентификатор компонента должен быть сообщен разработчику биометрического приложения разработчиком НПБК.

7.1.7 Доступ к ПИК компонента ПБК осуществляют через ОПИК.

7.1.8 Для получения доступа к компоненту ПБК модуль НПБК загружают по 7.4.1, а компонент — создают по 7.4.3. После завершения работы с компонентом его освобождают по 7.4.5, а модуль выгружают по 7.4.2.

7.1.9 С помощью ПИК НПБК выполняют следующие типовые операции: запрос к произвольному ПИК компонента по 7.4.3, предотвращение освобождения компонента по 7.4.4, освобождение компонента по 7.4.5, экспорт параметров НПБК по 7.4.6, импорт параметров НПБК по 7.4.7, обучение НПБК по 7.4.8, дообучение ЭП НПБК по 7.4.9, выполнение нейросетевого преобразования биометрия — код доступа по 7.4.10, выбор и установка начальной схемы преобразования по 7.4.11, тестирование обученного НПБК по 7.4.12.

### 7.2 Программный интерфейс модуля

7.2.1 ПИМ представляют как набор функций, экспортируемых модулем ПБК и реализуемых согласно 9.2.

7.2.2 Разработчик ПБК должен обеспечивать соответствие имен функций, порядка передачи параметров аргумента и правил возврата из них требованиям 9.1, кодов ошибок — требованиям 11.1.

### 7.3 Программный интерфейс компонента

7.3.1 ПИК представляют в виде особым образом оформленной таблицы функций, с помощью вызовов которых осуществляется взаимодействие с компонентом. Организация ПИК приведена на рисунке 4.

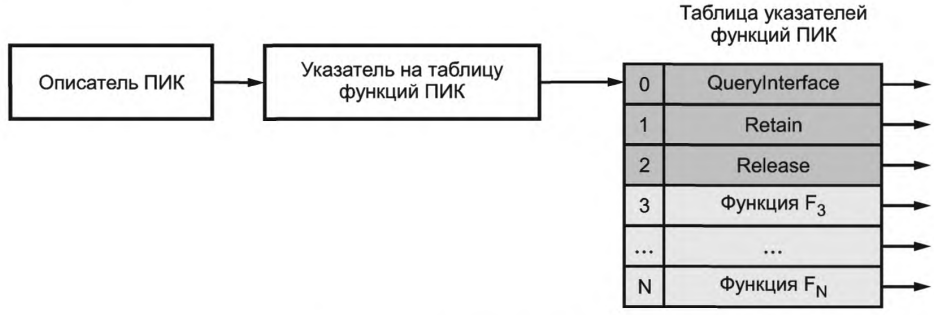


Рисунок 4 — Организация ПИК

7.3.2 Обеспечивают соответствие имен функций, порядка передачи параметров и правил возврата из них по 9.1, соответствие выбранного варианта ПИК его объявлению по 9.3—9.5, а также соответствие кодов ошибок по 11.1.

7.3.3 Каждому ПИК присваивают собственный УУИД ПИК, однозначно идентифицирующий набор функций ПИК. Набор УУИД для специфицируемых в настоящем стандарте ПИК определяют по 8.3.

7.3.4 Описатель ПИК представляют в виде указателя на область памяти, содержащей указатель на таблицу функций ПИК. Тип ОПИК определяют по 8.2.4.

7.3.5 В таблице указателей функций ПИК размещают указатели на точки входа функций компонента в порядке объявления функций в ПИК. В таблице не допускают значений, не являющихся указателями на реализованные функции. Нумерацию функций начинают с 0.

7.3.6 Первыми тремя функциями любого ПИК устанавливают QueryInterface по 9.3.1, Retain по 9.3.2, Release по 9.3.3.

7.3.7 Компонент ПБК может реализовывать несколько ПИК.

7.3.8 Экземпляр компонента ПБК должен возвращать одинаковый ОПИК для одного и того же ПИК.

Примечание — Значения ОПИК разных ПИК одного и того же компонента могут отличаться.

7.3.9 Проверку соответствия ОПИК одному и тому же компоненту ПБК проводят сравнением ОПИК «Неизвестный», полученных с помощью исходных ОПИК.

7.3.10 Компонент ПБК считают активным, если запрошен хотя бы один его ОПИК.

7.3.11 Компонент ПБК считают неактивным, если освобождены все его ОПИК.

Примечание — Компонент ПБК в неактивном состоянии должен быть удален соответствующим ему модулем ПБК при первой возможности.

7.3.12 Управление состоянием компонента (активное/неактивное) осуществляют с помощью типовых вызовов: запрос описателя ПИК — по 7.4.3, предотвращение освобождения компонента — по 7.4.4, освобождение компонента — по 7.4.5.

### 7.4 Типовые схемы взаимодействия биометрического приложения с нейросетевым преобразователем биометрия — код доступа

#### 7.4.1 Загрузка модуля

Загрузку модуля ПБК проводят с помощью штатных средств операционной системы.

#### 7.4.2 Освобождение модуля

Освобождение модуля ПБК проводят только после освобождения всех активных компонентов модуля с помощью штатных средств операционной системы.

#### 7.4.3 Запрос описателя ПИК и создание компонента

Запрос ОПИК осуществляют одним из двух способов:

1) вызывают функцию NbQueryComponent ПИМ НПБК по 9.2.1 с указанием УУИД ПИК и УУИД требуемого компонента.

Примечание — В этом случае одновременно с предоставлением ОПИК компонента выполняется создание компонента ПБК;

2) для имеющегося ОПИК вызывают функцию QueryInterface с указанием УУИД ПИК требуемого интерфейса.

Примечание — В этом случае возвращаемый ОПИК определяет тот же экземпляр компонента ПБК, что и имеющийся ОПИК.

#### 7.4.4 Предотвращение освобождения компонента

Для предотвращения освобождения компонента блокируют ОПИК компонента ПБК путем вызова функции Retain этого ОПИК.

Примечания

1 Блокировка компонента необходима, если ОПИК передается в другую часть биометрического приложения, функционирование которой продолжается независимо. В этом случае может возникнуть ситуация, когда обе части биометрического приложения попытаются освободить компонент. Тогда один из ОПИК станет недействительным, и может возникнуть ошибка доступа к компоненту.

2 Рекомендуется выполнять блокировку компонента всякий раз, когда выполняется дублирование ОПИК. Освобождение компонента по 7.4.5 необходимо выполнять всякий раз при удалении ОПИК.

#### 7.4.5 Освобождение компонента

Запрос на освобождение компонента, заданного ОПИК, проводят вызовом функции Release этого ОПИК. Последующее использование этого ОПИК запрещается.

#### 7.4.6 Экспорт параметров нейросетевого преобразователя биометрия — код доступа

7.4.6.1 Экспорт параметров НПБК в формате НБК проводят вызовом функции ExportNbc ПИК НПБК по 9.4.16.

Примечание — Вызов функции экспорта не должен приводить к изменению состояния НПБК.

7.4.6.2 Экспорт параметров НПБК в формате НБК допустим только для обученного НПБК.

#### 7.4.7 Импорт параметров нейросетевого преобразователя биометрия — код доступа

7.4.7.1 Импорт параметров НПБК из НБК проводят вызовом функции ImportNbc ПИК НПБК по 9.4.17.

7.4.7.2 Загрузку НБК неизвестного для НПБК типа не проводят.

7.4.7.3 Загрузку блоков ЭП НБК неизвестных для НПБК типов не проводят.

7.4.7.4 Разработчик НПБК должен поддерживать загрузку всех информационных блоков НБК для последующего экспорта.

**7.4.8 Обучение нейросетевого преобразователя биометрия — код доступа**

7.4.8.1 Обучение НПБК проводят однократно для начальной настройки НПБК.

7.4.8.2 Перед обучением разработчик биометрического приложения должен:

- выбрать и установить начальную схему преобразования по 7.4.11;
- при необходимости установить специальные входные и выходные параметры, а также ограничения вызовом функции SetConstraint ПИК НПБК по 9.4.8;
- при необходимости получения уведомлений во время обучения установить обработчик событий вызовом функции SetEventHandler ПИК НПБК по 9.4.4;
- при необходимости проверить наличие грубых отклонений во входных биометрических параметрах вызовом функции IndicateGrossErrors ПИК НПБК по 9.4.13 и исключить из входных примеров «Свой» примеры с большим значением отклонений.

7.4.8.3 Обучение запускают вызовом функции Train ПИК НПБК по 9.4.11.

Примечание — Биометрические параметры примеров «Все чужие», используемые во время обучения, должны быть получены из сбалансированной базы биометрических образов «Все чужие», созданной по ГОСТ Р 52633.1 и учитывающей особенности конкретной реализации биометрической технологии, в том числе и особенности устройств их ввода.

7.4.8.4 После завершения обучения проводят тестирование НПБК по 7.4.12.

**7.4.9 Дообучение нейросетевого преобразователя биометрия — код доступа**

7.4.9.1 Дообучение НПБК проводят для конкретного ЭП с целью дополнительной настройки его параметров. Дообучение проводят только после обучения НПБК по 7.4.8 или загрузки НБК, содержащего параметры обученного НПБК, по 7.4.7.

7.4.9.2 Перед дообучением разработчик биометрического приложения должен:

- при необходимости установить специальные входные, выходные параметры и ограничения для обучаемого слота вызовом функции SetConstraint ПИК НПБК по 9.4.8;
- при необходимости получения уведомлений во время дообучения установить обработчик событий вызовом функции SetEventHandler ПИК НПБК по 9.4.4.

7.4.9.3 Дообучение запускают вызовом функции PostTrain ПИК НПБК по 9.4.12.

**7.4.10 Выполнение нейросетевого преобразования биометрия — код доступа**

7.4.10.1 Нейросетевое преобразование проводят только для обученного НПБК, у которого установлены все требуемые специальные входные параметры.

7.4.10.2 Нейросетевое преобразование запускают вызовом функции Extract ПИК НПБК по 9.4.10.

**7.4.11 Выбор и установка начальной схемы преобразования**

7.4.11.1 Начальную схему преобразования формируют одним из способов:

- заполнением блока данных схемы преобразования по 6.1;
- выбором одной из поддерживаемых НПБК схем преобразования вызовом EnumConvScheme ПИК НПБК по 9.4.6.

7.4.11.2 Установку начальной схемы преобразования проводят вызовом функции SetConvScheme ПИК НПБК по 9.4.7.

**7.4.12 Тестирование обученного нейросетевого преобразователя биометрия — код доступа**

7.4.12.1 Тестирование проводят только для обученного НПБК.

7.4.12.2 НПБК должен поддерживать режимы тестирования НПБК по 8.2.14.

Примечание — Режимы тестирования соответствуют характеристикам НПБК, получаемым по ГОСТ Р 52633.0 во время тестирования.

7.4.12.3 Тестирование запускают вызовом функции Test ПИК НПБК по 9.4.14.

## 8 Типы данных, макросы и константы

### 8.1 Макросы

#### 8.1.1 Макрос nbCALL

Задает соглашение о вызовах функций ПИМ и ПИК.

```
#if defined(_WIN32)
#define nbCALL    __stdcall
#else
#define nbCALL
#endif
```

Примечание — Взаимодействие программных продуктов разных изготовителей в общем случае зависит от выбранного (иногда в соответствии с установками в заголовочном файле языка Си) представления в памяти (например, заполнение промежутков между элементами данных, механизм передачи параметров и использование регистров или стеков). Многие операционные системы выбирают один из вариантов по умолчанию, который должен использоваться. Если не существует выбранных установок по умолчанию, необходимо использовать опцию представления структур данных без заполнения промежутков между элементами и выравнивания структур данных по границам.

### 8.1.2 Макрос nbRefUuid

Определяет неизменяемую ссылку на УУИД. Используется для передачи идентификаторов компонентов, ПИК и ПИМ в функции.

```
#define nbRefUuid    const nbUuid * const
```

### 8.1.3 Макрос nbNULL

Определяет нулевое (неопределенное) значение для ОПИК, указателей, целых и вещественных значений. Может использоваться как значение по умолчанию для параметров, являющихся указателем, если это явно оговорено в описании функции.

```
#define nbNULL       0
```

## 8.2 Типы данных

В настоящем стандарте используются встроенные типы языка Си:
int8_t, int16_t, int32_t, int64_t, uint8_t, uint16_t, uint32_t, uint64_t, void, float,
а также вводятся собственные типы.

### 8.2.1 Тип nbBlockFlags

8.2.1.1 Определяет флаги состояния ЭП в схеме преобразования. Допускается совместное использование отдельных флагов.

8.2.1.2 Объявление

```
typedef uint16_t      nbBlockFlags;
```

8.2.1.3 Значения

```
#define nbBF_NONE                                    (0x0000)
```

Все флаги сброшены.

```
#define nbBF_EXCLUDE_UNUSED                          (0x0001)
```

Флаг исключения неиспользуемого ЭП из схемы преобразования.

Примечание — Наличие флага означает, что выходные и специальные выходные параметры слота во время выполнения нейросетевого преобразования не вычисляются. Разработчик биометрического приложения должен самостоятельно устанавливать значения этих параметров, если они необходимы для получения результата при выполнении нейросетевого преобразования.

```
#define nbBF_PROTECTED                               (0x0002)
```

Флаг защищенного ЭП.

Примечание — Наличие флага определяет состояние ЭП, в котором дополнительно проводится модификация параметров преобразования ЭП, препятствующая их изучению.

```
#define nbBF_HIDDEN                                  (0x0004)
```

Флаг скрытого ЭП.

Примечание — Используется для запрета обучения, дообучения и переобучения ЭП. Выходные и специальные параметры слота вычисляются в ходе нейросетевого преобразования или устанавливаются разработчиком биометрического приложения заранее, до установки флага.

```
#define nbBF_THROUGH                                 (0x0010)
```

Флаг идентичности входных и выходных параметров.

Примечание — Наличие флага означает, что данные выходных параметров равны входным.

```
#define nbBF_GENERATOR                               (0x0020)
```

Флаг генерации выходных параметров.

Примечание — Если флаг установлен, то значения выходных параметров выбираются разработчиком НПБК случайным образом, иначе во время обучения необходимо устанавливать их эталонные значения. Если используется совместно с флагом nbBF_THROUGH, выходные параметры вычисляются как производные от входных параметров, и их эталонные значения не требуются.

```
#define nbBF_BACKWARD_TRAIN                          (0x0040)
```

Флаг обратного обучения.

Примечание — Если флаг установлен, используется обратное направление обучения (от выходных параметров к входным) с соответствующим изменением интерпретации параметров для флагов nbBF_THROUGH и nbBF_GENERATOR.

#define nbBF_ITERATIVE_TRAIN (0x0080)
Флаг итерационного обучения.

Примечание — Используется в качестве индикатора необходимости продолжения обучения в цикле по всем слотам, связанным с ЭП. Устанавливается перед обучением НПБК.

### 8.2.2 Тип nbBlockHeader

8.2.2.1 Определяет заголовок блока НБК.

8.2.2.2 Объявление

```
typedef struct nbBlockHeader_t {
 uint16_t        sizeLo;
 uint8_t         sizeHi;
 nbBlockType     type;
} nbBlockHeader;
```

8.2.2.3 Параметры

sizeLo — младшие 16 бит длины блока НБК без заголовка блока НБК.
sizeHi — старшие 8 бит длины блока НБК без заголовка блока НБК.
type — тип блока НБК (тип ЭП или тип информационного блока).

Примечание — Максимальная длина данных блока НБК, считая без заголовка блока НБК, не может превышать $2^{24} - 1$ байт.

### 8.2.3 Тип nbBlockState

8.2.3.1 Определяет состояние элементарного преобразователя в схеме преобразования.

8.2.3.2 Объявление

```
typedef uint8_t        nbBlockState;
```

8.2.3.3 Значения

Значения в диапазоне [0…127] зарезервированы для предопределенных состояний ЭП разного типа. Значения в диапазоне [128…255] должны использоваться разработчиком НПБК для собственных состояний ЭП.

#define nbBS_UNTRAINED (0)
Элементарный преобразователь не обучен.

Примечание — Другие значения характеризуют обученный ЭП.

#define nbBS_TRAINED (1)
Элементарный преобразователь обучен.

### 8.2.4 Тип nbBlockType

8.2.4.1 Определяет тип элементарного преобразователя или информационного блока НПБК.

8.2.4.2 Объявление

```
typedef uint8_t        nbBlockType;
```

8.2.4.3 Значения

Значения из диапазона [1…127] определяют типы ЭП. Значения из диапазона [128…254] определяют типы информационных блоков. Значения из диапазона [1…63] могут использоваться разработчиком НПБК для определения собственных типов ЭП. Значения из диапазона [128…191] могут использоваться разработчиком биометрического приложения для определения собственных типов информационных блоков.

#define nbBT_NBC (0)
Нейросетевой биометрический контейнер.

Примечание — Данное значение задает тип блока заголовка нейросетевого биометрического контейнера.

#define nbBT_NEURAL_NET_CONVERTER (127)
Нейросетевой ЭП.
#define nbBT_FUZZY_CONVERTER (126)
Нечеткий ЭП.
#define nbBT_CRYPTO_CONVERTER (125)
ЭП, использующий криптографическое преобразование.
#define nbBT_HARD_INDICATOR (124)
ЭП, выполняющий жесткую индикацию.

#define nbBT_SOFT_INDICATOR (123)
ЭП, выполняющий мягкую индикацию.
#define nbBT_ERROR_DETECTOR (122)
ЭП для обнаружения и исправления ошибок.
#define nbBT_CONNECTOR (121)
ЭП для связывания по данным и размножения ошибок.
#define nbBT_BIO_AUDITOR (120)
ЭП для проведения биометрического аудита.
#define nbBT_FUZZY_ADDRESSER (119)
ЭП нечеткой адресации.
#define nbBT_CODE_TRANSFORMER (118)
ЭП трансформации (выходного) кода.
#define nbBT_SECURITY (117)
ЭП для обеспечения безопасности (блок безопасности).
#define nbBT_NBC_ID (254)
Информационный блок идентификатора НБК.
#define nbBT_USER_ID (253)
Информационный блок идентификатора пользователя.
#define nbBT_DATE (252)
Информационный блок дат.
#define nbBT_BIO_TECH (251)
Информационный блок описания используемой биометрической технологии.
#define nbBT_CSCHEME (250)
Информационный блок схемы преобразования.
#define nbBT_TEST_RESPONSE (249)
Информационный блок откликов НПБК.
#define nbBT_EXTENSION (255)
Расширение предыдущего блока конфигурации ЭП или информационного блока, если его данные не умещаются в 1 блок.

Примечание — Разработчики биометрических приложений должны корректно обрабатывать возможное появление блоков типа nbBT_EXTENSION.

### 8.2.5 Тип nbConstraintType

8.2.5.1 Определяет тип ограничения для параметров ЭП.

8.2.5.2 Объявление

typedef uint16_t nbConstraintType;

8.2.5.3 Значения

#define nbCT_DATA (1)
Примеры «Чужой» или «Свой» для преобразования.
#define nbCT_DATA_OWN (2)
Примеры «Свой» для обучения и тестирования.
#define nbCT_DATA_ALL (3)
Примеры «Все чужие» для обучения и тестирования.
#define nbCT_SIGMA (4)
Стандартное отклонение распределений биометрического параметра.
#define nbCT_INTERVAL (5)
Область допустимых значений биометрического параметра.
#define nbCT_DISPLACEMENT (6)
Смещение распределения «Все чужие» относительно центра ОДЗ.
#define nbCT_DISCRETE_STEP (7)
Шаг дискретизации в ОДЗ.
#define nbCT_RANGE_LIMIT (8)
Максимальное по модулю учитываемое значение дискретного биометрического параметра.
#define nbCT_OWN_LAW (9)
Закон распределения биометрических параметров примеров «Свои».
#define nbCT_ALL_LAW (10)
Закон распределения биометрических параметров примеров «Все чужие».

#define nbCT_ALGORITHM (11)
Основной алгоритм, используемый ЭП.
#define nbCT_SALT (12)
Соль (случайное число).
#define nbCT_ERROR_DETECTION_RATE (13)
Число обнаруживаемых ошибок.
#define nbCT_ERROR_CORRECTION_RATE (14)
Число исправляемых ошибок.
#define nbCT_CRYPTO_STRENGTH_REDUCTION (15)
Величина снижения стойкости.
#define nbCT_CONN_ID (16)
Идентификатор соединения, связанный с параметром.

**8.2.6 Тип nbData**

8.2.6.1 Определяет указатель на блок данных, оформленный особым образом.

8.2.6.2 Объявление

typedef uint8_t* nbData;

8.2.6.3 Структура блока данных представлена на рисунке 5.

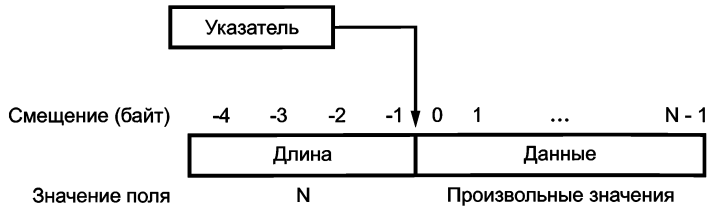


Рисунок 5 — Структура блока данных в памяти

Указатель — определяет указатель на блок данных.

Поле «Длина» (4 байта) — задает длину поля «Данные» в байтах. Поле должно располагаться по смещению минус 4 байта относительно значения указателя на блок данных.

Поле «Данные» — содержит данные блока. Интерпретация содержимого поля должна зависеть от контекста его использования и должна быть явно оговорена.

Примечание — Тип используется для хранения сложно структурированных данных в непрерывном блоке памяти без необходимости спецификации этой структуры. Например, блок данных может содержать строковое значение в кодировке UTF-8. В этом случае поле «Длина» определяет максимальный размер блока данных, который может занять строка.

8.2.6.4 Формат хранения блока данных определяется в соответствии с А.6 (приложение А).

**8.2.7 Тип nbEvent**

8.2.7.1 Определяет событие обработчика событий.

8.2.7.2 Объявление

typedef int32_t nbEvent;

8.2.7.3 Значение события должно интерпретироваться в зависимости от компонента, который его инициирует. Допустимые значения событий определены в разделе 10. Значения из диапазона [0x00000000…0x00007FFF] и диапазона [0x80000000…0xFFFFFFFF] зарезервированы.

**8.2.8 Тип nbHandle**

8.2.8.1 Определяет описатель программного интерфейса компонента.

8.2.8.2 Объявление

typedef void* nbHandle;

8.2.8.3 Значение описателя несуществующего компонента равно nbNULL.

8.2.8.4 Используется для идентификации созданных компонентов и доступа к ПИК компонентов.

Примечание — ОПИК представляет собой указатель на область памяти, содержащую указатель на таблицу функций ПИК в порядке их объявления. Обращение к функциям ПИК должно осуществляться после приведения описателя к описателю соответствующего ПИК. Нельзя использовать полученный ОПИК в качестве первого параметра (параметра this) функции ПИК другого компонента. Описатели разных ПИК одного и того же компонента могут отличаться.

**8.2.9 Тип nbNbccState**

8.2.9.1 Определяет состояния НПБК. Общее состояние НПБК описывается комбинацией состояний.

8.2.9.2 Объявление

typedef uint16_t nbNbccState;

8.2.9.3 Значения

#define nbNS_INITIAL (0x0000)

Начальное состояние НПБК.

#define nbNS_TRAINED (0x0001)

Флаг состояния «I». НПБК обучен.

#define nbNS_CSCHEME (0x0002)

Флаг состояния «II». Схема преобразования установлена.

#define nbNS_SPEC_IN (0x0004)

Флаг состояния «III». Специальные входные параметры и их ограничения заданы.

#define nbNS_SPEC_OUT (0x0008)

Флаг состояния «IV». Специальные выходные параметры, а также результаты промежуточных вычислений получены.

#define nbNS_CRITICAL_IN (0x0010)

Флаг состояния «V». Входные параметры для обучения, дообучения и тестирования, заданы.

#define nbNS_CRITICAL_OUT (0x0020)

Флаг состояния «VI». Выходные параметры для обучения, дообучения и тестирования заданы.

**8.2.10 Тип nbNbcBody**

8.2.10.1 Определяет заголовок НБК.

8.2.10.2 Объявление

```
typedef struct nbNbcBody_t {
 uint32_t        size;
 nbUuid          nbcType;
} nbNbcBody;
```

8.2.10.3 Параметры

size — полная длина НБК в байтах с заголовком НБК.

nbcType — УУИД типа нейросетевого биометрического контейнера.

Примечание — УУИД типа нейросетевого контейнера определяет разработчик НПБК. Идентификатор должен полностью определять формат хранения блоков НБК.

**8.2.11 Тип nbPurpose**

8.2.11.1 Определяет признаки и цели использования биометрических образов или производных от них биометрических параметров. Допустимо значения различных целей и признаков использовать совместно.

8.2.11.2 Объявление

typedef uint8_t nbPurpose;

8.2.11.3 Значения

#define nbPURPOSE_UNKNOWN (0x00)

Цель неизвестна.

#define nbPURPOSE_AUTHENTICATE (0x01)

Аутентификация (сопоставление 1:1).

#define nbPURPOSE_ENTER (0x02)

Вхождение в группу (поиск в базе 1:∞ ).

#define nbPURPOSE_IDENTIFY (0x04)

Идентификация для обучения.

#define nbPURPOSE_AUDIT (0x08)

Для аудита или тестирования.

#define nbPURPOSE_MULTI_USER (0x10)

Содержит биометрические образы разных пользователей.

#define nbPURPOSE_MULTI_IMAGE (0x20)

Содержит несколько биометрических образов.

#define nbPURPOSE_BALANCED (0x40)

Содержит сбалансированную базу биометрических образов.

#define nbPURPOSE_SYNTHETIC (0x80)

Содержит синтетические образы.

**8.2.12 Тип nbResult**

8.2.12.1 Определяет значение результата вызова функции ПИК или ПИМ. Допустимые значения результата включают в себя:

- значения успешного вызова;
- значения ошибок.

8.2.12.2 Объявление

typedef int32_t nbResult;

8.2.12.3 Интерпретация значения результата определяется значением старшего бита. Если старший бит сброшен в «0», значение результата должно интерпретироваться как значение успешного вызова (см. рисунок 6), иначе значение результата должно интерпретироваться как значение ошибки (см. рисунок 7).

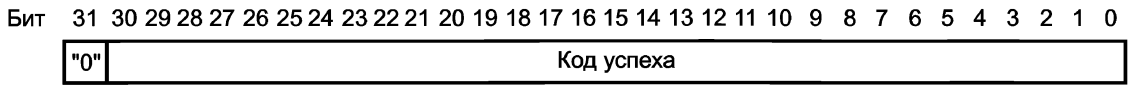


Рисунок 6 — Значение успешного вызова

Бит 31 — сброшен в «0».

Поле «Код успеха» (биты 0—30) — содержит значение, специфичное для конкретной функции. Допускается использование любого значения кода успеха.

Типичные значения успешного вызова определены согласно 8.2.12.4.

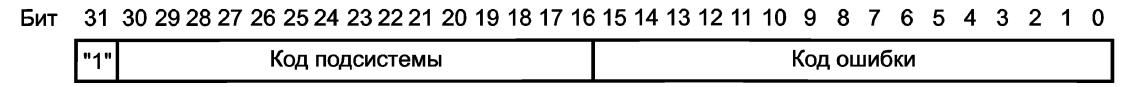


Рисунок 7 — Значение ошибки

Бит 31 — установлен в «1».

Поле «Код подсистемы» (биты 16—30) — определяет подсистему, для которой интерпретируется код ошибки. Перечень допустимых подсистем определен по 11.2.

Поле «Код ошибки» (биты 0—15) — определяет код ошибки. Коды ошибок должны определяться разработчиками компонентов в соответствии с особенностями реализации ПИК или быть предопределенными. Перечень предопределенных кодов ошибок определен по 11.3.

8.2.12.4 Значения успешного вызова функции

#define nbS_OK (0x00000000)

Успешно.

#define nbS_CANCEL (0x00000001)

Отменено.

#define nbS_SKIP (0x00000002)

Пропущено.

**8.2.13 Тип nbSlotID**

8.2.13.1 Определяет номер слота НПБК, уникальный в пределах одной схемы преобразования. В зависимости от диапазона значений позволяет адресовать в рамках НПБК:

- слот входного, специального входного или специального выходного параметра;
- слот элементарного преобразователя.

8.2.13.2 Объявление

typedef uint16_t nbSlotID;

8.2.13.3 Структура типа показана на рисунке 8.

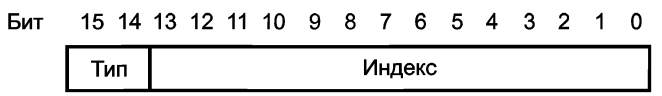


Рисунок 8 — Структура номера слота в памяти

Поле «Тип» (биты 14—15) — определяет тип слота.

Поле «Индекс» (биты 0—14) — определяет уникальный индекс слота в пределах схемы преобразования. Если поле «Тип» определяет слот ЭП, то старшие 7 бит поля «Индекс» соответствуют типам

ЭП со значениями из диапазона [1…127], а младшие 7 бит его номеру в схеме преобразования из диапазона [1…127].

8.2.13.4 Значения типа слота

#define nbSLOT_TYPE_IN (0)

Слот параметра, содержащий входной параметр.

#define nbSLOT_TYPE_SPEC_IN (1)

Слот параметра, содержащий специальный входной параметр.

#define nbSLOT_TYPE_SPEC_OUT (2)

Слот параметра, содержащий специальный выходной параметр.

#define nbSLOT_TYPE_CONV (3)

Слот ЭП, содержащий выходной параметр.

8.2.13.5 Макросы формирования номера слота

#define nbSLOT_OUT (nbSlotID)(0x0000)

Задает номер слота выходного кода НПБК.

#define nbSLOT_IN(i) (nbSlotID)(0x3FFF&i)

Задает номер слота входного параметра с индексом i. Значение i должно принадлежать множеству [0x0001…0x3FFF].

#define nbSLOT_SPEC_IN(i) (nbSlotID)(0x4000|(0x3FFF&i))

Задает номер слота специального входного параметра с индексом i. Значение индекса должно принадлежать множеству [0x0001…0x3FFF].

#define nbSLOT_SPEC_OUT(o) (nbSlotID)(0x8000|(0x3FFF&o))

Задает номер слота специального выходного параметра с индексом. Значение индекса из множества [0x0001…0x3FFF].

#define nbSLOT_CONV (bt,i) (nbSlotID)(0xC000|(0x3F80&(bt<<7))|(0x007F&i))

Задает номер слота ЭП с индексом i и типом bt. Значение индекса должно принадлежать множеству [1…127]. Значение типа ЭП должно принадлежать множеству [1…127].

8.2.13.6 Используется для идентификации параметров нейросетевого преобразования и элементарных преобразователей в схеме преобразования, а также во время импорта/экспорта НБК.

**8.2.14 Тип nbTestMode**

8.2.14.1 Определяет режим тестирования НПБК.

8.2.14.2 Объявление

typedef uint16_t nbTestMode;

8.2.14.3 Значения

#define nbTEST_MODE_E1_BIO (1)

Оценка вероятности ошибки первого рода (отказ по примеру образа «Свой»).

#define nbTEST_MODE_E2_BIO (2)

Оценка вероятности ошибки второго рода (пропуск по примеру образа «Чужой»).

#define nbTEST_MODE_E2_BIO_HUMAN_COMPROMISED (3)

Оценка вероятности ошибки второго рода при скомпрометированном биометрическом образе (известна косвенная информация о правилах его построения).

#define nbTEST_MODE_E2_BIO_COMPROMISED (4)

Оценка вероятности ошибки второго рода при скомпрометированном биометрическом образе (известен одиночный пример биометрического образа «Свой»).

#define nbTEST_MODE_E2_CODE (5)

Оценка эффективной длины ключа в битах.

#define nbTEST_MODE_E2_BIO_WHITE_NOISE (6)

Оценка вероятности ошибки второго рода при подборе биометрического образа с помощью атаки белым шумом.

#define nbTEST_MODE_E2_BIO_CORR_NOISE (7)

Оценка вероятности ошибки второго рода при подборе биометрического образа с помощью атаки коррелированным шумом, учитывающим особенности распределения биометрических образов «Все чужие».

#define nbTEST_MODE_E2_CONV (8)

Оценка вероятности ошибки при атаке подбора с известным НБК.

8.2.14.4 Используется во время тестирования НПБК.

**8.2.15 Тип nbTime**

8.2.15.1 Определяет дату и время. Хранит число секунд, прошедших с 00:00:00 1 января 1970. Отрицательное значение задает дату и время до 1 января 1970.

8.2.15.2 Объявление

typedef int64_t nbTime;

8.2.15.3 Используется для представления дат и времени в НПБК.

**8.2.16 Тип nbTimeout**

8.2.16.1 Определяет время ожидания в микросекундах ($10^{-6}$ с).

Примечание — Максимальное время ожидания составляет примерно 4295 с, поэтому разработчик должен корректно обрабатывать ситуацию истечения времени, например, предлагая вариант повторной инициализации счетчика времени ожидания.

8.2.16.2 Объявление

typedef uint32_t nbTimeout;

8.2.16.3 Специальные значения

#define nbTIMEOUT_INFINITE ((uint32_t)(-1))

Ожидать бесконечно.

#define nbTIMEOUT_NO_WAIT (0)

Без ожидания.

8.2.16.4 Используется для задания максимального времени проведения длительной операции, например, создания компонента или обучения НПБК, а также времени ожидания события.

**8.2.17 Тип nbTimeType**

8.2.17.1 Определяет вид даты и времени.

8.2.17.2 Объявление

typedef uint8_t nbTimeType;

8.2.17.3 Значения

#define nbTIME_CREATE (1)

Дата и время создания.

#define nbTIME_LAST_MODIFY (2)

Дата и время модификации.

#define nbTIME_LAST_ACCESS (3)

Дата и время последнего использования.

#define nbTIME_EXPIRITE (4)

Дата и время истечения срока действия.

#define nbTIME_ERROR_I (10)

Дата и время последней ошибки первого рода.

#define nbTIME_ERROR_II (11)

Дата и время последней ошибки второго рода.

**8.2.18 Тип nbTrainMode**

8.2.18.1 Определяет режим обучения (дообучения) ЭП. Определяется как комбинация отдельных режимов обучения.

8.2.18.2 Объявление

typedef uint16_t nbTrainMode;

8.2.18.3 Значения

#define nbTRAIN_MODE_DEFAULT (0x0000)

Нормальный режим обучения.

#define nbTRAIN_MODE_WITHOUT_FLAGS (0x1000)

Без изменения флагов ЭП.

#define nbTRAIN_MODE_WITHOUT_STATE (0x2000)

Без изменения состояния ЭП.

#define nbTRAIN_MODE_CHECK_ONLY (0x8000)

Проверка параметров обучения без выполнения обучения (переобучения).

**8.2.19 Тип nbUuid**

8.2.19.1 Определяет универсальный уникальный идентификатор, используемый для идентификации ПИК, типа НБК, компонентов НПБК.

8.2.19.2 Объявление

```
typedef struct nbUuid_t {
  uint32_t    d1;
  uint16_t    d2;
  uint16_t    d3;
  uint8_t     d4[8];
} nbUuid;
```

8.2.19.3 Предопределенное значение, в котором все биты УУИД сброшены в «0», обозначает «пустой» идентификатор.

```
const nbUuid nbUUID_NIL =
{0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}};
```

**8.2.20 Тип nbVbp**

8.2.20.1 Определяет вектор биометрических параметров. Формат представления отдельных биометрических параметров определяется метаописанием ВБП.

8.2.20.2 Объявление

```
typedef void*          nbVbp;
```

8.2.20.3 Используется для представления в памяти характеристик биометрических образов, выходного кода и параметров НПБК.

**8.2.21 Тип nbMatrix**

8.2.21.1 Определяет матрицу ВБП.

8.2.21.2 Объявление

```
typedef struct nbMatrix_t {
 uint32_t        nrows;
 uint32_t        ncols;
 nbMeta          *meta;
 nbVbp           **data;
} nbMatrix;
```

8.2.21.3 Параметры

nrows — число частей в каждом примере.

ncols — число примеров.

meta — указатель на вектор метаописаний представления ВБП соответствующей части примера.

data — двойной указатель на ВБП примеров.

Примечание — Указатели поля «data» должны быть установлены так, чтобы по первому измерению осуществлять доступ к частям (nrows) примеров, по второму — к конкретным ВБП (ncols). Т. е. доступ ВБП j-й части i-го примера должен осуществляться с помощью оператора data[j][i].

8.2.21.4 В зависимости от комбинации значений отдельных полей матрица ВБП может быть отнесена к одному из типов:

а) «пустая». Не содержит данных и метаописаний. Допустимые значения полей:

nrows = 0; ncols = 0; meta = nbNULL; data = nbNULL;

б) «шаблон». Определяет матрицу ВБП, у которой определено только метаописание. Допустимые значения полей:

nrows $\in$ [0x00000001…0x0000FFFF]; ncols = 0; meta = указатель на вектор метаописаний; data = nbNULL;

в) «заполненная». Определяет матрицу ВБП, которая содержит данные. Допустимые значения полей:

nrows $\in$ [0x00000001…0x0000FFFF]; ncols $\in$ [0x00000001…0xFFFFFFFF]; meta = указатель на вектор метаописаний; data = двойной указатель на ВБП;

г) «глобальный идентификатор». Определяет глобальный уникальный идентификатор базы данных, содержащей примеры матрицы ВБП.

nrows = значения бит 0—31 УУИД, кроме значений из диапазона [0x00000001…0x0000FFFF]; ncols = значения бит 32—63 УУИД; meta = значения бит 64—95 УУИД; data = значения бит 96—127 УУИД.

Примечание — Если УУИД в младших 32 битах содержит значение из диапазона [0x00000001…0x0000FFFF], он должен быть сгенерирован заново;

д) «описатель». Определяет ОПИК доступа к базе данных, содержащей примеры матрицы ВБП.

nrows = 0; ncols = 0; meta = указатель на ОПИК; data = указатель на ОПИК.

Примечание — Настоящий стандарт не специфицирует операции с матрицами ВБП перечисленных типов г) и д). Однако НПБК должен корректно распознавать матрицы ВБП типов г) и д) и выдавать код ошибки nbECODE_MATRIX_NOTATION.

8.2.21.5 Используется для передачи/получения параметров и ограничений нейросетевого преобразования в НПБК.

8.2.21.6 Формат хранения матрицы ВБП в непрерывном блоке определяется по А.4 (приложение А).

**8.2.22 Тип nbMeta**

8.2.22.1 Определяет метаописание ВБП.

8.2.22.2 Объявление

```
typedef struct nbMeta_t {
 uint32_t          count;
 nbMetaFormat      format;
 nbMetaType        type;
} nbMeta;
```

8.2.22.3 Параметры

count — число биометрических параметров в ВБП.

format — формат биометрических параметров.

type — тип биометрических параметров.

**8.2.23 Тип nbMetaFormat**

8.2.23.1 Определяет формат представления биометрического параметра в памяти.

8.2.23.2 Объявление

```
typedef uint8_t        nbMetaFormat;
```

8.2.23.3 Значения

#define nbMF_ANY (0)

Любой числовой тип.

Примечание — Используется для метаописания любого допустимого входного формата параметров в схеме преобразования. На момент фактической передачи значений параметров это значение использовать недопустимо.

#define nbMF_I1 (0+ 1)

Знак. Область значений: [– 1,+ 1].

#define nbMF_I2 (0 + 2)

Целое число длиной 2 бит. Область значений: [– 2…1].

#define nbMF_I4 (0 + 3)

Целое число длиной 4 бит. Область значений: [– 8…7].

#define nbMF_I8 (0 + 4)

Целое число длиной 1 байт. Соответствует типу int8_t.

#define nbMF_I16 (0 + 5)

Целое число длиной 2 байт. Соответствует типу int16_t.

#define nbMF_I32 (0 + 6)

Целое число длиной 4 байт. Соответствует типу int32_t.

#define nbMF_I64 (0 + 7)

Целое число длиной 8 байт. Соответствует типу int64_t.

#define nbMF_U1 (16 + 1)

Бит. Область значений: [0,1].

#define nbMF_U2 (16 + 2)

Неотрицательное целое число длиной 2 бит. Область значений: [0…3].

#define nbMF_U4 (16 + 3)

Неотрицательное целое число длиной 4 бит. Область значений: [0…15].

#define nbMF_U8 (16 + 4)

Неотрицательное целое число длиной 1 байт. Соответствует типу uint8_t.

#define nbMF_U16 (16 + 5)

Неотрицательное целое число длиной 8 байт. Соответствует типу uint16_t.

#define nbMF_U32 (16 + 6)

Неотрицательное целое число длиной 8 байт. Соответствует типу uint32_t.

#define nbMF_U64 (16 + 7)

Неотрицательное целое число длиной 8 байт. Соответствует типу uint64_t.

#define nbMF_R32 (32 + 6)

Вещественное число одинарной точности. Соответствует типу float.

#define nbMF_R64 (32 + 7)

Вещественное число с двойной точностью. Соответствует типу double.

**8.2.24 Тип nbMetaType**

8.2.24.1 Определяет качественную характеристику (тип) биометрического параметра.

8.2.24.2 Объявление
typedef uint8_t nbMetaType;
8.2.24.3 Значения
#define nbMT_ANY (0)
Любой тип.

Примечание — Используется для метаописания любого допустимого входного типа параметров в схеме преобразования. На момент фактической передачи параметров это значение использовать недопустимо.

#define nbMT_CONTINUOUS (1)
Непрерывный биометрический параметр.
#define nbMT_DISCRETE (3)
Дискретный биометрический параметр.
#define nbMT_CONTINUOUS_OWN (5)
Неизменяемый по области «Свой» непрерывный биометрический параметр.
#define nbMT_DISCRETE_OWN (7)
Неизменяемый по области «Свой» дискретный биометрический параметр.

### 8.3 Константы идентификаторов программных интерфейсов компонентов

#### 8.3.1 Константа nbUUID_IUNKNOWN

static const nbUuid nbUUID_IUNKNOWN =
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46};
Определяет УУИД ПИК Неизвестный.

#### 8.3.2 Константа nbUUID_INBCC

static const nbUuid nbUUID_INBCC =
{0x60,0x3e,0x21,0xa3,0xb3,0x57,0x43,0x46,0x84,0xe4,0x11,0xca,0xb9,0xe8,0x1f,0xfe};
Определяет УУИД ПИК НПБК.

#### 8.3.3 Константа nbUUID_IEVENT_HANDLER

static const nbUuid nbUUID_IEVENT_HANDLER =
{0x37,0x92,0x5a,0x58,0x9d,0x8d,0xdb,0x4e,0xab,0x24,0x8f,0x41,0xc9,0xa2,0x2a,0xc6};
Определяет УУИД ПИК Обработчик событий.

## 9 Функции

### 9.1 Общие положения

9.1.1 Используется соглашение о вызовах функции __stdcall.

9.1.2 Порядок объявления функций, размещаемых в таблицах функций, должен соответствовать порядку их описания в настоящем стандарте. Первая описываемая функция каждого ПИК имеет номер 0.

9.1.3 Память под параметры функций должна выделяться вызывающей стороной, если явно не оговорено иное.

9.1.4 Результаты работы функции должны возвращаться только через ее параметры или в качестве значения результата.

### 9.2 Программный интерфейс модуля

Функции этого раздела реализуются разработчиком НПБК и экспортируются модулем НПБК.

#### 9.2.1 Функция NbQueryComponent

nbResult nbCALL NbQueryComponent(nbRefUuid cid, nbRefUuid iid, nbHandle *icd);

9.2.1.1 Функция запрашивает или создает компонент с заданным идентификатором и возвращает запрашиваемый описатель программного интерфейса компонента.

9.2.1.2 Параметры

cid (входной) — указатель на УУИД запрашиваемого компонента, определенный разработчиком компонента. Передача значения nbNULL приводит к созданию компонента, определенного разработчиком модуля НПБК в качестве создаваемого по умолчанию.

iid (входной) — указатель на УУИД запрашиваемого ПИК.

icd (выходной) — указатель ОПИК для запрашиваемого и создаваемого компонента.

9.2.1.3 Возвращает nbS_OK в случае успешного завершения работы функции. Другие значения содержат код ошибки.

9.2.1.4 Коды ошибок
nbECODE_NO_COMPONENT
nbECODE_NO_INTERFACE
nbECODE_INTEGRITY_FAIL

### 9.3 Программный интерфейс компонента Неизвестный (nbIUnknown)

Доступ к функциям этого раздела осуществляется с помощью ОПИК компонента.

**9.3.1 Функция QueryInterface**

nbResult QueryInterface(nbHandle this, nbRefUuid iid, nbHandle *icd);

9.3.1.1 Функция выдает новый описатель программного интерфейса компонента для уже созданного компонента.

9.3.1.2 Параметры
this (входной) — ОПИК.
iid (входной) — указатель на УУИД запрашиваемого ПИК.
icd (выходной) — указатель на ОПИК компонента.

9.3.1.3 Возвращает nbS_OK в случае успешного завершения работы функции. Другие значения содержат код ошибки.

9.3.1.4 Коды ошибок
nbECODE_NO_INTERFACE

**9.3.2 Функция Retain**

nbResult Retain(nbHandle this);

9.3.2.1 Функция увеличивает внутренний счетчик использования компонента на единицу.

9.3.2.2 Параметры
this (входной) — ОПИК.

9.3.2.3 Возвращает положительное значение, которое интерпретируется как текущее значение счетчика использования компонента.

**9.3.3 Функция Release**

nbResult Release(nbHandle this);

9.3.3.1 Функция уменьшает значение счетчика использования компонента на единицу.

9.3.3.2 Параметры
this (входной) — ОПИК.

9.3.3.3 Возвращает неотрицательное значение, которое интерпретируется как текущее значение счетчика использования компонента. Значение «0» означает, что компонент больше не используется и должен быть освобожден.

### 9.4 Программный интерфейс компонента Нейросетевой преобразователь биометрия — код доступа (nbINbcc)

Функции этого раздела реализуются разработчиком НПБК для компонента НПБК. Доступ к функциям осуществляется с помощью ОПИК компонента.

**9.4.1 Функция QueryInterface**

Соответствует функции QueryInterface ПИК Неизвестный по 9.3.1.

**9.4.2 Функция Retain**

Соответствует функции Retain ПИК Неизвестный по 9.3.2.

**9.4.3 Функция Release**

Соответствует функции Release ПИК Неизвестный по 9.3.3.

**9.4.4 Функция SetEventHandler**

nbResult SetEventHandler(nbHandle this, nbIEventHandler *handler);

9.4.4.1 Функцию устанавливает обработчик событий НПБК.

9.4.4.2 Параметры
this (входной) — ОПИК НПБК.
handler (входной) — ОПИК Обработчик событий.

9.4.4.3 Возвращает nbS_OK, если обработчик установлен. Другие значения содержат код ошибки.

9.4.4.4 Коды ошибок
nbECODE_POINTER
nbECODE_UNSUPPORTED_FUNCTION

**9.4.5 Функция SetConvScheme**

nbResult SetConvScheme(nbHandle this, const nbData scheme);

9.4.5.1 Функция устанавливает начальную схему преобразования НПБК.

9.4.5.2 Параметры

this (входной) — ОПИК НПБК.

scheme (входной) — указатель на блок данных с начальной схемой преобразования. Значение указателя nbNULL задает схему преобразования, выбираемую разработчиком НПБК по умолчанию.

9.4.5.3 Возвращает nbS_OK, если схема успешно установлена. Другие значения содержат код ошибки.

9.4.5.4 Коды ошибок

nbECODE_OUT_OF_MEMORY
nbECODE_CONSISTENCY_FAIL
nbECODE_UNSUPPORTED_CSCHEME
nbECODE_UNSUPPORTED_BLOCK_TYPE
nbECODE_META
nbECODE_PARAM_IN_NUMBER
nbECODE_PARAM_SPEC_IN_NUMBER
nbECODE_PARAM_SPEC_OUT_NUMBER
nbECODE_BLOCK_FLAG
nbECODE_BLOCK_STATE
nbECODE_NUMERATION_RULE

### 9.4.6 Функция EnumConvScheme

nbResult EnumConvScheme(nbHandle this, nbRefUuid csid, const nbMatrix *mtxAny, const nbMatrix *mtxCode, nbData scheme);

9.4.6.1 Функция возвращает информацию о поддерживаемых схемах преобразования или типах данных.

9.4.6.2 Параметры

this (входной) — ОПИК НПБК.

csid (входной) — указатель на УУИД схемы преобразования или типа данных. Значения УУИД в диапазоне [0x00000001...0x0000FFFF] определяют порядковый номер схемы, известной НПБК. Номера должны располагаться по порядку без пропусков. Значение указателя nbNULL запрашивает схему преобразования по умолчанию.

mtxAny (входной/необязательный) — указатель на матрицу ВБП, представляющую шаблон входных параметров. Значение указателя nbNULL определяет шаблон на основе известных НПБК схем преобразования. Формат представления параметра определяется по 6.3.1.6.

mtxCode (входной/необязательный) — указатель на матрицу ВБП, представляющую шаблон выходного кода. Значение указателя nbNULL определяет шаблон на основе известных НПБК схем преобразования. Формат представления параметра определяется по 6.3.1.6.

scheme (выходной) — указатель на блок данных, хранящий начальную схему преобразования или описание типа данных. Если размер блока данных недостаточен, в поле «Длина» блока данных записывается значение требуемой длины блока и возвращается код ошибки nbECODE_INSUFFICIENT_BUFFER.

9.4.6.3 Возвращает значение nbS_OK, если блок данных сформирован правильно. Другие значения содержат код ошибки.

9.4.6.4 Коды ошибок

nbECODE_IDENTIFIER
nbECODE_UNBOUND_INDEX
nbECODE_INSUFFICIENT_BUFFER
nbECODE_META
nbECODE_PARAM_IN_NUMBER
nbECODE_PARAM_OUT_NUMBER

Примечание — Код ошибки nbECODE_IDENTIFICATOR должен возвращаться, если запрошена схема преобразования с несуществующим номером.

### 9.4.7 Функция GetSlotDescr

nbResult GetSlotDescr(nbHandle this, nbSlotID id, nbData descr);

9.4.7.1 Функция считывает описание слота с учетом его фактического состояния.

9.4.7.2 Параметры

this (входной) — ОПИК НПБК.

id (входной) — номер слота, для которого делается запрос.

descr (выходной/необязательный) — указатель на блок данных, в который записываются текущие данные слота параметров или ЭП. Если размер блока данных недостаточен, в поле «Длина» блока данных записывается значение требуемой длины блока и возвращается код ошибки nbECODE_INSUFFICIENT_BUFFER.

9.4.7.3 Возвращает номер слота, следующего за слотом с номером id, по схеме преобразования. Другие значения содержат код ошибки.

Примечание — В случае успеха при вызове функции для последнего слота должно возвращаться значение nbSLOT_OUT.

9.4.7.4 Коды ошибок

nbECODE_NBCC_STATE

nbECODE_SLOT_ID

nbECODE_INSUFFICIENT_BUFFER

**9.4.8 Функция SetConstraint**

nbResult SetConstraint (nbHandle this, nbSlotID id, nbConstraintType conType, const nbMatrix *mtxCon);

9.4.8.1 Функция устанавливает ограничение для слота схемы преобразования, а также биометрические параметры примеров «Чужой», «Свой» или «Все чужие».

9.4.8.2 Параметры

this (входной) — ОПИК НПБК.

id (входной) — номер слота, у которого устанавливается ограничение.

conType (входной) — тип устанавливаемого ограничения.

mtxCon (входной) — указатель на матрицу ВБП, содержащую данные ограничения. Формат представления ограничения определяется по 6.3.1.3. Передача nbNULL отменяет установленное ограничение. При установке ограничений типа nbCT_DATA, nbCT_DATA_OWN, nbCT_DATA_ALL должен передаваться указатель на матрицу ВБП, содержащую, соответственно, примеры «Свой»/«Чужой», «Свой», «Все чужие».

9.4.8.3 Возвращает nbS_OK, если данные ограничения установлены. Другие значения содержат код ошибки.

9.4.8.4 Коды ошибок

nbECODE_ACCESS_DENIED

nbECODE_INVALID_NBCC_STATE

nbECODE_INVALID_SLOT

nbECODE_UNSUPPORTED_CONSTRAINT

nbECODE_UNSUPPORTED_PARAM

nbECODE_NO_DATA

nbECODE_UNBOUND_DATA

nbECODE_MATRIX_NOTATION

nbECODE_META

**9.4.9 Функция GetConstraint**

nbResult GetConstraint(nbHandle this, nbSlotID id, nbConstraintType conType, nbMatrix *mtxCon);

9.4.9.1 Функция считывает ограничение для слота схемы преобразования.

9.4.9.2 Параметры

this (входной) — ОПИК НПБК.

id (входной) — номер слота, у которого считывается ограничение.

conType (входной) — тип считываемого ограничения.

mtxCon (выходной) — указатель на матрицу ВБП, в которую будут записаны данные ограничения. Формат представления параметра определяется по 6.3.1.3.

9.4.9.3 Возвращает nbS_OK, если ограничение считано. Другие значения содержат код ошибки.

9.4.9.4 Коды ошибок

Коды ошибок включают коды ошибок функции SetConstraint по 9.4.8.

**9.4.10 Функция Extract**

nbResult Extract(nbHandle this, const nbMatrix *mtxAny, nbMatrix *mtxCode, const nbSlotID *idCode);

9.4.10.1 Функция выполняет нейросетевое преобразование, преобразуя биометрические параметры в выходной код.

9.4.10.2 Параметры

this (входной) — ОПИК НПБК.

mtxAny (входной) — указатель на матрицу ВБП, содержащую преобразуемые примеры «Свой»/«Чужой» для всех входных параметров схемы преобразования в порядке их объявления. Формат представления отдельных входных параметров определяется по 6.3.1.1.

Примечание — Разработчик НПБК должен устанавливать переданные данные в качестве ограничений типа nbCT_DATA соответствующих слотов входных параметров.

mtxCode (выходной) — указатель на матрицу ВБП, содержащую результаты нейросетевого преобразования для слотов из списка idCode. Формат представления параметров определяется по 6.3.1.1.

Примечание — Разработчик НПБК должен считывать данные из ограничений типа nbCT_DATA соответствующих слотов входных и специальных входных параметров без изменений, а слотов ЭП и специальных выходных параметров — по результатам нейросетевого преобразования.

idCode (входной/необязательный) — указатель на список номеров слотов, для которых вычисляются результаты нейросетевого преобразования. Значение nbNULL предполагает вычисление значения только для слота выходного кода (nbSLOT_OUT) или отождествленного с ним.

9.4.10.3 Возвращает nbS_OK в случае успешного выполнения НПБК. Другие значения содержат код ошибки.

9.4.10.4 Коды ошибок

Коды ошибок включают в себя коды ошибок функции SetConstraint по 9.4.8.

**9.4.11 Функция Train**

nbResult Train(nbHandle this, const nbMatrix *mtxOwn, const nbMatrix *mtxAll, const nbMatrix *mtxCode, nbTimeout timeout);

9.4.11.1 Функция выполняет обучение НПБК, т. е. настройку параметров его работы и конфигураций ЭП.

Примечание — Функция должна вызываться только после установки начальной схемы преобразования НПБК.

9.4.11.2 Параметры

this (входной) — ОПИК НПБК.

mtxOwn (входной) — указатель на матрицу ВБП, содержащую примеры «Свой» для всех входных параметров схемы преобразования в порядке их объявления. Формат представления параметра определяется по 6.3.1.1.

Примечание — Разработчик НПБК должен устанавливать переданные данные в качестве ограничений типа nbCT_DATA_OWN соответствующих слотов входных параметров.

mtxAll (входной/необязательный) — указатель на матрицу ВБП, содержащую базу примеров «Все чужие» для всех входных параметров схемы преобразования в порядке их объявления. Значение указателя nbNULL определяет требование использовать значения по умолчанию. Формат представления параметра определяется по 6.3.1.1.

Примечание — Разработчик НПБК должен устанавливать переданные данные в качестве ограничений типа nbCT_DATA_ALL соответствующих слотов входных параметров.

mtxCode (входной) — указатель на матрицу ВБП, содержащий выходной код. Формат представления параметра определяется по 6.3.1.1.

timeout (входной) — максимально допустимое время обучения.

Примечание — По истечении максимально допустимого времени, отведенного на обучение, функция должна завершить работу с кодом ошибки nbECODE_TIMEOUT_EXPIRED. Состояние НПБК в этом случае не определено.

9.4.11.3 Возвращает nbS_OK, если обучение завершено успешно. Другие значения содержат код ошибки.

9.4.11.4 Коды ошибок

Коды ошибок включают коды ошибок функции SetConstraint по 9.4.8.

Коды ошибок включают коды ошибок функции PostTrain по 9.4.12.

**9.4.12 Функция PostTrain**

nbResult PostTrain(nbHandle this, nbSlotID id, nbTrainMode trainMode, nbBlockState blockState, nbBlockFlags blockFlags, nbTimeout timeout);

9.4.12.1 Функция выполняет дообучение или индикацию возможности обучения заданного ЭП.

Примечание — Функция должна вызываться только после успешного обучения НПБК или загрузки НБК.

9.4.12.2 Параметры

this (входной) — ОПИК НПБК.

id (входной) — номер слота дообучаемого ЭП.

trainMode (входной) — режим обучения.

blockState (входной/необязательный) — состояние ЭП.

blockFlags (входной/необязательный) — набор устанавливаемых/сбрасываемых флагов ЭП.

timeout (входной) — максимально допустимое время обучения.

Примечания

1 По истечении максимально допустимого времени обучения функция должна завершить работу с кодом ошибки nbECODE_TIMEOUT_EXPIRED. Состояние НПБК в этом случае не определено.

2 В зависимости от выбранных функции дообучения может потребоваться установка дополнительных параметров (ограничений). Для этого следует использовать функцию SetConstraint перед вызовом функции дообучения.

9.4.12.3 Возвращает nbS_OK, если дообучение проведено успешно или возможность его проведения подтверждена. Другие значения содержат код ошибки.

9.4.12.4 Коды ошибок

nbECODE_TIMEOUT_EXPIRED
nbECODE_CANCELLED
nbECODE_ACCESS_DENIED
nbECODE_NBCC_STATE
nbECODE_SLOT_ID
nbECODE_BLOCK_STATE
nbECODE_BLOCK_FLAG
nbECODE_UNATTAINABLE_BLOCK_STATE
nbECODE_UNSUPPORTED_TRAIN_MODE
nbECODE_NO_DATA
nbECODE_NO_DATA_OWN
nbECODE_NO_DATA_ALL
nbECODE_POOR_DIVERGENCE
nbECODE_POOR_QUALITY
nbECODE_POOR_STABILITY
nbECODE_POOR_ORIGINALITY
nbECODE_TOO_MANY_GROSS_ERRORS

**9.4.13 Функция IndicateGrossErrors**

nbResult IndicateGrossErrors(nbHandle this, const nbMatrix *mtxOwn, const nbMatrix *mtxAll, nbMatrix *mtxGe);

9.4.13.1 Функция выполняет индикацию сильных отклонений примеров «Свой» по отношению к некоторому среднему ожидаемому значению, что позволяет исключить до начала обучения биометрические образы, ослабляющие обучающую выборку.

Примечание — Функция должна вызываться только после установки начальной схемы преобразования НПБК.

9.4.13.2 Параметры

this (входной) — ОПИК НПБК.

mtxOwn (входной) — указатель на матрицу ВБП, содержащую примеры «Свой» для всех входных параметров схемы преобразования в порядке их объявления. Формат представления параметра определяется по 6.3.1.1.

mtxAll (входной) — указатель на матрицу ВБП, содержащую базу примеров «Все чужие» для всех входных параметров схемы преобразования в порядке их объявления. Формат представления параметра определяется по 6.3.1.1.

mtxGe (выходной) — указатель на матрицу ВБП, содержащую вектор индикаторов, показывающих отклонение отдельных примеров «Свой» от среднего ожидаемого значения. Большее значение соответствует большему отклонению. Формат представления параметра определяется по В.4 (приложение В).

9.4.13.3 Возвращает nbS_OK, если функция успешно завершила свою работу. Другие значения содержат код ошибки.

9.4.13.4 Коды ошибок
nbECODE_NBCC_STATE
nbECODE_NO_DATA
nbECODE_NO_DATA_OWN
nbECODE_NO_DATA_ALL
nbECODE_UNBOUND_DATA
nbECODE_TOO_MANY_GROSS_ERRORS

**9.4.14 Функция Test**

nbResult Test(nbHandle this, nbTestMode testMode, nbMatrix *mtxEst, nbTimeout timeout);

9.4.14.1 Функция выполняет тестирование обученного НПБК.

Примечание — Функция должна вызываться только для обученного НПБК непосредственно после обучения или дообучения.

9.4.14.2 Параметры
this (входной) — ОПИК НПБК.
testMode (входной) — режим тестирования.
mtxEst (выходной) — указатель на матрицу ВБП, содержащую значение оцениваемого параметра. Формат представления параметра определяется по 6.3.1.4.
timeout (входной) — максимальное время проведения тестирования.

Примечание — В зависимости от выбранного режима тестирования может потребоваться установка дополнительных параметров (ограничений). Для этого следует использовать функцию SetConstraint перед вызовом функции тестирования.

9.4.14.3 Возвращает nbS_OK, если тестирование проведено успешно. Другие значения содержат код ошибки.
9.4.14.4 Коды ошибок
nbECODE_NBCC_STATE
nbECODE_UNSUPPORTED_TEST_MODE
nbECODE_TIMEOUT_EXPIRED
nbECODE_CANCELLED
nbECODE_PARAM
Другие коды ошибок, зависящие от режима тестирования.

**9.4.15 Функция Reset**

nbResult Reset(nbHandle this, nbNbccState nbccState);

9.4.15.1 Функция осуществляет сброс отдельных или группы состояний НПБК.
9.4.15.2 Параметры
this (входной) — ОПИК НПБК.
nbccState (входной) — комбинация сбрасываемых состояний НПБК. Значение nbNS_INITIAL не изменяет состояние НПБК.
9.4.15.3 Возвращает фактическое состояние НПБК на момент завершения вызова. Другие значения содержат код ошибки.
9.4.15.4 Коды ошибок
nbECODE_FAIL
nbECODE_NBCC_STATE

**9.4.16 Функция ExportNbc**

nbResult ExportNbc(nbHandle this, nbData nbc, nbRefUuid nbcType);

9.4.16.1 Функция осуществляет экспорт конфигураций ЭП и параметров работы НПБК в форме НБК.
9.4.16.2 Параметры
this (входной) — ОПИК НПБК.
nbc (выходной) — указатель на блок данных, в который будут записаны данные НБК. Формат параметра определяется в соответствии с 6.4. Если размер блока данных недостаточен, в поле «Длина» записывается значение требуемой длины блока и возвращается код ошибки nbECODE_INSUFFICIENT_BUFFER.
nbcType (входной) — указатель на тип НБК, в формате которого должен быть проведен экспорт. Значение nbNULL определяет тип НБК по умолчанию.
9.4.16.3 Возвращает nbS_OK, если данные НПБК успешно экспортированы. Другие значения содержат код ошибки.

9.4.16.4 Коды ошибок
nbECODE_NBCC_STATE
nbECODE_INCORRECT_VERSION
nbECODE_INSUFFICIENT_BUFFER

**9.4.17 Функция ImportNbc**

nbResult ImportNbc(nbHandle this, const nbData nbc);

9.4.17.1 Функция осуществляет импорт конфигураций ЭП и параметров работы НПБК из НБК.

9.4.17.2 Параметры

this (входной) — ОПИК НПБК.

nbc (входной) — указатель на блок данных, содержащий НБК. Формат параметра определяется по 6.4.

9.4.17.3 Возвращает nbS_OK, если конфигурации ЭП и параметры НПБК успешно импортированы. Другие значения содержат код ошибки.

9.4.17.4 Коды ошибок
nbECODE_OUT_OF_MEMORY
nbECODE_INCORRECT_VERSION
nbECODE_CONSISTENCY_FAIL
nbECODE_PARAM

### 9.5 Программный интерфейс компонента Обработчик событий (nbIEventHandler)

Функции этого раздела реализуются разработчиком биометрического приложения для некоторого компонента модуля с целью его совместного использования с НПБК.

**9.5.1 Функция QueryInterface**

Соответствует функции QueryInterface ПИК Неизвестный по 9.3.1.

**9.5.2 Функция Retain**

Соответствует функции Retain ПИК Неизвестный по 9.3.2.

**9.5.3 Функция Release**

Соответствует функции Release ПИК Неизвестный по 9.3.3.

**9.5.4 Функция RaiseEvent**

nbResult RaiseEvent(nbHandle this, nbEvent evt, nbHandle sender, const nbData paramA, nbData paramB);

9.5.4.1 Функция вызывается НПБК для сообщения о некотором событии.

9.5.4.2 Параметры

this (входной) — ОПИК Обработчик событий.

evt (входной) — тип события.

paramA (входной/необязательный) — параметр, интерпретация которого зависит от типа события.

paramB (входной/выходной/необязательный) — параметр, интерпретация которого зависит от типа события.

9.5.4.3 Возвращаемое значение зависит от типа события. Типичное возвращаемое значение nbS_OK. Другие значения содержат код ошибки.

## 10 Обработка событий

Во время своей работы НПБК должен отправлять уведомления о событиях биометрическому приложению в случае установки обработчика событий. Обмен событиями должен быть реализован биометрическим приложением как ПИК Обработка событий.

### 10.1 Общие положения

10.1.1 Тип события представляет собой 32-разрядное число по 8.2.7.

10.1.2 Значение события передается в функцию RaiseEvent ПИК Обработчик событий.

10.1.3 Возвращаемое значение функции RaiseEvent ПИК Обработчик событий зависит от типа события.

### 10.2 События во время перечисления объектов

**10.2.1 Событие nbEVENT_ENUMERATE_ITEM**

10.2.1.1 Определяет событие перечисления некоторого множества объектов.

10.2.1.2 Значение

#define nbEVENT_ENUMERATE_ITEM (0x00000001)

10.2.1.3 Параметры функции RaiseEvent ПИК Обработчик событий

paramA (входной) — (должен приводиться к типу int32_t ) порядковый номер объекта или 0.

paramB (входной) — (должен приводиться к указателю на тип запрашиваемого объекта) указатель на перечисляемый объект. Тип объекта должен быть известен на момент установки обработчика сообщений.

10.2.1.4 Для продолжения перечисления функция должна возвращать значение nbS_OK. Значение nbS_CANCEL прерывает перечисление.

### 10.3 События во время обучения нейросетевого преобразователя биометрия — код доступа

#### 10.3.1 Событие nbEVENT_NBCC_PROCESS

10.3.1.1 Определяет динамическое событие, возникающее во время обучения, тестирования, извлечения НПБК.

10.3.1.2 Значение

#define nbEVENT_NBCC_PROCESS (0x00000401)

10.3.1.3 Параметры функции RaiseEvent ПИК Обработчик событий:

paramA (входной) — (должен приводиться к типу nbSlotID) номер текущего слота.

paramB (входной) — (должен приводиться к типу uint32_t) текущее состояние выполнения из диапазона значений [0…$2^{32}$ – 1], причем максимальное значение обозначает конец обучения.

10.3.1.4 Возвращаемое значение nbS_OK продолжает работу функции. Значение nbS_CANCEL прерывает работу функции.

## 11 Обработка ошибок

Все функции ПИМ и ПИК возвращают значение типа nbResult. Обработка ошибок должна проводиться путем анализа этого значения.

### 11.1 Общие положения

11.1.1 Выделение значений ошибок из всего множества допустимых значений nbResult проводят путем анализа старшего бита. После этого выделяют код подсистемы и, в зависимости от его значения, интерпретируют код ошибки.

11.1.2 Для проверки значения результата на наличие значения ошибки используют макросы.

#define nbSUCCEEDED(r) ((nbResult)(r)>=0)

Проверка на получение значения успешного вызова.

#define nbFAILED(r) ((nbResult)(r)<0)

Проверка на получение значения ошибки.

11.1.3 Разработку НПБК выполняют таким образом, чтобы при возникновении ошибки не терялась устойчивость его работы.

### 11.2 Коды подсистем

#define nbF_SYS (0x0000)

Диапазон зарезервирован для подсистем ОС. Коды ошибок соответствуют кодам ошибок операционной системы.

#define nbF_DEF (0x2000)

Подсистема общих ошибок, примененная для текущего компонента. Допустимые коды ошибок определены в 11.4.

#define nbF_SUB (0x4000)

Начальное значение диапазона [0x4000…0x7FFF] подсистемы ошибок для указания источника ошибки внутри компонента. Для ПИК НПБК младшие 14 бит кода подсистемы соответствуют индексу слота схемы преобразования, при работе с которым возникла ошибка. Коды ошибок для подсистемы соответствуют 11.4.

### 11.3 Определяемые реализацией коды ошибок

Разработчики биометрических приложений и НПБК могут определять собственные коды ошибок в диапазоне [0x8000…0xFFFF] для любой подсистемы.

### 11.4 Коды ошибок

#define nbECODE_FAIL (0x0001)
Общий сбой.
#define nbECODE_POINTER (0x0002)
Неправильный указатель или выход за границы допустимых значений, определенных для указателя.
#define nbECODE_PARAM (0x0003)
Недопустимое значение параметра функции.
#define nbECODE_IDENTIFIER (0x0004)
Неправильный идентификатор.
#define nbECODE_INTEGRITY_FAIL (0x0005)
Нарушена целостность модуля/компонента.
#define nbECODE_CONSISTENCY_FAIL (0x0006)
Нарушена целостность данных или данные имеют другой формат.
#define nbECODE_OUT_OF_MEMORY (0x0007)
Ошибка выделения памяти (при выделении блока данных вызываемой стороной).
#define nbECODE_INSUFFICIENT_BUFFER (0x0008)
Недостаточный размер буфера (при передаче блока данных вызывающей стороной).
#define nbECODE_ACCESS_DENIED (0x0009)
Доступ запрещен.
#define nbECODE_NO_MODULE (0x000A)
Не найден модуль.
#define nbECODE_NO_COMPONENT (0x000B)
Не найден запрашиваемый компонент.
#define nbECODE_NO_INTERFACE (0x000C)
Не найден запрашиваемый ПИК.
#define nbECODE_OBJECT_NOT_FOUND (0x000D)
Объект (файл, ключ и т. д.) не найден.
#define nbECODE_OBJECT_ALREADY_EXISTS (0x000E)
Объект уже существует (файл, компонент и т. д.).
#define nbECODE_TIMEOUT_EXPIRED (0x000F)
Операция не завершена по причине окончания времени.
#define nbECODE_CANCELLED (0x0010)
Операция не завершена, т. к. была отменена извне.
#define nbECODE_INCORRECT_VERSION (0x0011)
Некорректная версия.
#define nbECODE_IO (0x0012)
Ошибка ввода/вывода.
#define nbECODE_UNSUPPORTED_FUNCTION (0x0013)
Неподдерживаемая функция.
#define nbECODE_MATRIX_NOTATION (0x0014)
Неподдерживаемый формат представления матрицы.
#define nbECODE_UNBOUND_INDEX (0x0015)
Индекс выходит за пределы допустимых значений.
#define nbECODE_UNBOUND_DATA (0x0016)
Значения данных не удовлетворяют наложенным ограничениям.
#define nbECODE_META (0x0017)
Недопустимое метаописание.
#define nbECODE_META_PATTERN (0x0018)
Неправильный вектор метаописаний.
#define nbECODE_META_NIL (0x0019)
Пустое метаописание недопустимо (поле count равно 0).
#define nbECODE_NBCC_STATE (0x0020)
Неправильное (неизвестное) состояние НПБК не позволяет выполнить операцию.
#define nbECODE_SLOT_ID (0x0021)
Идентификатор слота неизвестен (слот отсутствует в схеме).
#define nbECODE_UNSUPPORTED_TEST_MODE (0x0022)
Неподдерживаемый режим тестирования.

#define nbECODE_UNSUPPORTED_TRAIN_MODE (0x0023)
Неподдерживаемый режим обучения.
#define nbECODE_UNSUPPORTED_CONSTRAINT (0x0024)
Неподдерживаемое ограничение.
#define nbECODE_UNSUPPORTED_PARAM (0x0025)
Неподдерживаемый параметр.
#define nbECODE_UNSUPPORTED_CSCHEME (0x0026)
Неподдерживаемая или некорректная схема обучения.
#define nbECODE_UNSUPPORTED_BLOCK_TYPE (0x0027)
Неподдерживаемый тип блока НБК.
#define nbECODE_NUMERATION_RULE (0x0028)
Ошибка выбора правила нумерации (слотов).
#define nbECODE_NO_DATA (0x002A)
Не заданы данные параметра или число примеров не соответствует требуемому.
#define nbECODE_NO_DATA_OWN (0x002B)
Не заданы или недостаточно примеров «Свой».
#define nbECODE_NO_DATA_ALL (0x002C)
Не заданы или недостаточно примеров «Чужой».
#define nbECODE_BLOCK_STATE (0x0030)
Неправильное состояние ЭП.
#define nbECODE_BLOCK_FLAG (0x0031)
Неподдерживаемый флаг (или комбинация) ЭП.
#define nbECODE_UNATTAINABLE_BLOCK_STATE (0x0032)
Недостижимое из текущего состояние ЭП.
#define nbECODE_PARAM_IN_NUMBER (0x0033)
Неправильное число входных параметров (слотов).
#define nbECODE_PARAM_SPEC_IN_NUMBER (0x0034)
Неправильное число специальных входных параметров (слотов).
#define nbECODE_PARAM_SPEC_OUT_NUMBER (0x0035)
Неправильное число специальных выходных параметров (слотов).
#define nbECODE_POOR_DIVERGENCE (0x0037)
Плохая сходимость метода для заданных параметров.
#define nbECODE_POOR_QUALITY (0x0038)
Низкое качество образов.
#define nbECODE_POOR_STABILITY (0x0039)
Низкая стабильность образов.
#define nbECODE_POOR_ORIGINALITY (0x003A)
Низкая уникальность образов.
#define nbECODE_TOO_MANY_GROSS_ERRORS (0x003B)
Слишком много сильных отклонений от ожидаемого значения.
#define nbECODE_OBJECT_STATE (0x003D)
Неправильное состояние объекта.
#define nbECODE_NO_DEVICE (0x003E)
Не найдено устройство.

**Приложение А**
**(обязательное)**

## Формат сетевого представления данных

### А.1 Правила отображения данных с помощью АСН.1

А.1.1 Все описываемые ниже типы предназначены для описания данных, расположенных в непрерывном блоке памяти, причем:

а) не должны использоваться разрывы и заполнители, а также вспомогательные элементы представления (тэги, идентификаторы классов, типов и т. д.);

б) внутри блока данных должно использоваться представление полей типов по значению;

в) выравнивание простых и составных типов должно проводиться по границам байтов, при этом дополняемые биты должны устанавливаться в «0»;

г) должен использоваться прямой порядок следования байт (little-endian);

д) размер типа должен определяться по внешнему полю (условию) или по комбинации начальных полей данных типа.

А.1.2 Для представления данных в соответствии с ГОСТ Р ИСО/МЭК 8824-1 применяется модифицированная АСН.1 в части:

- дополнительные ограничения типов и условные зависимости между ними записываются внутри ограничений в виде строк с выражениями на языке Си;
- используется параметризация типов по ГОСТ Р ИСО/МЭК 8824-4;
- синтаксис оператора CHOICE расширен вариантом с применением условия выбора значения CHOICE(x).

А.1.3 Типы, объявленные в стандарте языка Си и в настоящем стандарте, вводятся в спецификацию АСН.1 с помощью ключевого слова EXTERN с указанием исходного типа и комментария, определяющего источник экспорта.

### А.2 Внешние типы данных

```
uint8_t          ::= EXTERN «uint8_t»
int8_t           ::= EXTERN «int8_t»
uint16_t         ::= EXTERN «uint16_t»
uint32_t         ::= EXTERN «uint32_t»
nbSlotID         ::= EXTERN «nbSlotID»
nbMeta           ::= EXTERN «nbMeta»
nbBlockFlags     ::= EXTERN «nbBlockFlags»
nbBlockState     ::= EXTERN «nbBlockState»
nbTestMode       ::= EXTERN «nbTestMode»
nbTime           ::= EXTERN «nbTime»
nbTimeType       ::= EXTERN «nbTimeType»
nbBlockHeader    ::= EXTERN «nbBlockHeader»
nbConstraint     ::= EXTERN «nbConstraint»
nbPurpose        ::= EXTERN «nbPurpose»
```

### А.3 Схема преобразования

А.3.1 Представление схемы преобразования.

```
- - схема преобразования
C-Scheme ::= SEQUENCE {
  slot-out-descr   SlotPrm{i-out},
  slots-in-descr   SET OF SlotPrm{i-in} OPTIONAL,
  transf-descr     SET OF CHOICE {
                       SlotPrm{i-spec-in | i-spec-out},
                       SlotConv
                   } OPTIONAL
}
- - типы идентификатора слота
SlotID ::= CHOICE {
  i-out          nbSlotID{«nbSLOT_OUT»},
  i-in           nbSlotID{«nbSLOT_IN(1)»..«nbSLOT_IN(16383)»},
  i-spec-in      nbSlotID{«nbSLOT_SPEC_IN(1)»..«nbSLOT_SPEC_IN(16383)»},
  i-spec-out     nbSlotID{«nbSLOT_SPEC_OUT(1)»..«nbSLOT_SPEC_OUT(16383)»},
  i-conv         nbSlotID{«nbSLOT_CONV(1,1)»..«nbSLOT_CONV(127,127)»}
}
```

```
- - слот параметра
SlotPrm{SlotID:type} ::= SEQUENCE {
 id            SlotID{type},                  - - идентификатор слота
 meta          nbMeta                         - - метаописание параметра
}

- - слот ЭП
SlotConv ::=SEQUENCE {
 id            SlotID{i-conv},          - - идентификатор слота
 meta          nbMeta,                  - - метаописание выходного параметра ЭП
 block-flags   nbBlockFlags,            - - флаги ЭП
 block-state   nbBlockState,            - - состояние ЭП
 ids-count     uint8_t,                 - - число связанных слотов
 ids           SET SIZE(ids-count) OF SlotID{i-in | i-spec-in | i-spec-out} OPTIONAL
                                        - - связанные слоты
}
```

### A.4 Матрица ВБП (nbMatrix)

A.4.1 Представление типов матрицы ВБП с помощью модифицированной АСН.1

```
MatrixType ::= CHOICE {
 - - заполненная матрица
 m-full        Matrix{nrows{«0x00000001»..«0x0000FFFF»},
                      ncols(ALL),
                      pmeta(0), pdata(0),
                      meta(ALL),data(ALL)},
 - - матрица, содержащая шаблон данных
 m-template    Matrix{nrows(«0x00000001»..«0x0000FFFF»),
                      ncols(0),
                      pmeta(0), pdata(0),
                      meta(ALL), NULL},
 - - пустая матрица
 m-empty       Matrix{nrows(«0x00000000»),
                      ncols(«0x00000000»),
                      pmeta(0), pdata(0),
                      NULL, NULL},
 - - матрица, содержащая УУИД
m-guid         Matrix{nrows(«0x00010000»..«0xFFFFFFFF»),
                      ncols(ANY),
                      pmeta(ALL EXCEPT 0), pdata(ALL),
                      NULL, NULL},
 - - матрица, содержащая ОПИК для доступа к базе данных матрицы
 m-pointer     Matrix{nrows(«0x00000000»),
                      ncols(«0x00000000»),
                      pmeta(ALL EXCEPT 0), pdata(ALL),
                      NULL, NULL}
}
```

A.4.2 Представление матрицы ВБП с помощью модифицированной АСН.1

```
Matrix ::= SEQUENCE {
 nrows         uint32_t,                - - число частей матрицы
 ncols         uint32_t,                - - число примеров матрицы
 pmeta         uint32_t,                - - дополнительное значение
 pdata         uint32_t,                - - дополнительное значение
                                        - - блок метаописаний
 meta          SET OF nbMeta SIZE(nrows) OPTIONAL,
                                        - - набор данных для всех частей и всех примеров матрицы
 data          SET SIZE(nrows) OF SET SIZE(ncols) OF Vbp{meta[irow]} OPTIONAL
                                        - - где irow — номер текущего измерения data (0…nrows – 1)
}

Vbp{nbMeta:meta} ::= SET SIZE(«meta.size()») OF uint8_t
        - - для определения размера используется функция meta.size(),
        - - вычисляющая размер вектора ВБП на основе формата
        - - элемента meta.format и числа элементов в векторе meta.count.
```

### А.5 Нейросетевой биометрический контейнер

А.5.1 Представление НБК и блоков НБК с помощью модифицированной АСН.1

```
- - нейросетевой биометрический контейнер
Nbc ::= SEQUENCE {
  nbc           Block{«nbBT_NBC»},
  blocks        SET OF ExtentiableBlock
}
- - расширяемый с помощью вспомогательных блоков блок НБК
ExtentiableBlock{nbBlockType:type} ::= SEQUENCE {
  block         Block{type EXPECT «nbBT_EXTENTION»},
  ext           SET OF Block{«nbBT_EXTENTION»}
}
- - блок НБК
Block{nbBlockType:type} ::= SEQUENCE {
  header        nbBlockHeader,              - - заголовок блока НБК
  body          Body{type}                  - - блок НБК
}
- - варианты тела блока НБК в зависимости от типа блока
Body{nbBlockType:type} ::= CHOICE(type) {
  nbcc-def              [1..63]                                 BodyNbccDef,
  nbc                   [«nbBT_NBC»]                            BodyNbc,
  nn-converter          [«nbBT_NEURAL_NET_CONVERTER»]           BodyNbccDef,
  fuzzy-converter       [«nbBT_FUZZY_CONVERTER»]                BodyNbccDef,
  crypto-converter      [«nbBT_CRYPTO_CONVERTER»]               BodyNbccDef,
  hard-indicator        [«nbBT_HARD_INDICATOR»]                 BodyNbccDef,
  soft-indicator        [«nbBT_SOFT_INDICATOR»]                 BodyNbccDef,
  error-spectator       [«nbBT_ERROR_SPECTATOR»]                BodyNbccDef,
  connector             [«nbBT_CONNECTOR»]                      BodyNbccDef,
  bio-auditor           [«nbBT_BIO_AUDITOR»]                    BodyNbccDef,
  fuzzy-addresser       [«nbBT_FUZZY_ADDRESSER»]                BodyNbccDef,
  code-transformer      [«nbBT_CODE_TRANSFORMER»]               BodyNbccDef,
  tester                [«nbBT_TEST_RESPONSE»]                  BodyNbccDef,
  security              [«nbBT_SECURITY»]                       BodySecurity,

  app-def               [128..191]                              BodyAppDef,
  nbc-id                [«nbBT_NBC_ID»]                         BodyNbcId,
  user-id               [«nbBT_USER_ID»]                        BodyUserId,
  dates                 [«nbBT_DATE»]                           BodyDates,
  bio-tech              [«nbBT_BIO_TECH»]                       BodyBioTech,
  cscheme               [«nbBT_CSCHEME»]                        BodyCScheme,
  test-response         [«nbBT_TEST_RESPONSE»]                  BodyTestResponse,
  ext                   [«nbBT_EXTENSION»]                      BodyExtension,
…
}
- - тело заголовка НБК
BodyNbc ::= SEQUENCE {
  size          uint32_t,           - - полный размер НБК без блока заголовка НБК
  nbc-type      nbUuid              - - УУИД типа НБК
}
- - тело блока ЭП НБК
BodyNbccDef ::= SET OF uint8_t OPTIONAL
- - тело информационного блока НБК
BodyAppDef ::= SET OF uint8_t OPTIONAL
- - тело блока безопасности
BodySecurity ::= SEQUENCE {
  zone              int16_t,          - - число контролируемых блоков
          - - положительное значение равно числу блоков после блока безопасности
          - - отрицательное значение равно числу блоков перед блоком безопасности
          - - значение «0x0000» определяет особый порядок подсчета (не блочный);
          - - заголовок НБК считается отдельным блоком
```

```
  owner          uint16_t,                          - - автор формата (0 — разработчик НПБК)
  format         uint16_t,                          - - формат данных блока
  data           SET OF uint8_t                     - - контекстно-зависимые данные
                                                    - - (хэш, контрольная сумма, ЭЦП)
}
- - тело идентификатора контейнера
BodyNbcId ::= SET OF uint8_t OPTIONAL
- - тело идентификатора пользователя
BodyUserId ::= SET OF uint8_t OPTIONAL
- - тело блока дат
BodyDates ::= SET OF SEQUENCE {
  time-type    nbTimeType,
  time         nbTime
}
- - тело описания биометрической технологии, используемой для выполнения
- - преобразования биометрия-код
BodyBioTech ::= SET OF SEQUENCE {
            - - тип биометрического образа
  bim-type          nbUuid                        DEFAULT {«nbUUID_NIL»},
            - - идентификатор провайдера, обеспечившего ввод биометрических образов
  provider-id       nbUuid                        DEFAULT {«nbUUID_NIL»},
            - - идентификатор процессора, выполнившего перевод биометрических
            - - образов в матрицу биометрических параметров
  processor-id      nbUuid                        DEFAULT {«nbUUID_NIL»},
            - - число параметров на выходе устройства обработки биометрических образов,
            - - для которых производится связывание
  n                 uint16_t                      DEFAULT {0},
            - - соответствующие частям идентификаторы входных параметров НПБК
            - - (для пропускаемой части значение устанавливается в SlotID:i-out)
  slot-ids    SET SIZE(n) OF SlotID OPTIONAL
}
- - тело схемы преобразования
BodyCScheme ::= C-Scheme
- - тело блока откликов ИНС по результатам тестирования
BodyTestResponse :: =  SEQUENCE {
  test-mode         nbTestMode,                   - - режим тестирования
  time              nbTime,                       - - дата и время тестирования
  response-count    uint16_t,                     - - число записей откликов
  response-base     SET SIZE(response-count) OF TestRec,        - - отклики
  input-count       uint16_t DEFAULT{0},          - - число входных записей
  input-base        SET SIZE(input-count) TestRec,        - - входные данные
  cert              nbData OPTIONAL               - - данные тестирующего
}
TestRec ::= SEQUENCE {
  con          nbConstraint,                 - - тип ограничения (входных/выходных параметров)
  ids-count    uint16_t,
  ids          SET SIZE(ids-count) OF SlotID,  - - номера слотов, для которых
                                             - - устанавливаются или снимаются ограничения
  data         Matrix                        - - данные ограничения
}
```

### A.6 Блок данных (nbData)

A.6.1 Представление блока данных или строки в кодировке UTF-8 с помощью модифицированной ACH.1

```
Data :: = SEQUENCE {
  size         uint32_t,                              - - длина блока данных в байтах
  data         SET SIZE(size) OF uint8_t OPTIONAL      - - блок данных или строка
}
```

**Приложение Б**
**(обязательное)**

## Типовые схемы организации параметров элементарных преобразователей

Типовые схемы организации и интерпретации параметров ЭП приведены в таблице Б.1.

Т а б л и ц а Б.1 — Типовые схемы организации и интерпретации параметров элементарного преобразователя

| Тип ЭП | Тип параметра(ов) | Число параметров | Формат параметра(ов) | Интерпретация параметра(ов) |
|---|---|---|---|---|
| nbBT_NEURAL_NET _CONVERTER | Входной | Любое | По таблице В.1 приложения В | Непрерывные или дискретные биометрические параметры |
| | Специальный входной | 0 | — | — |
| | Специальный выходной | 0 | — | — |
| | Выходной | 1 | Не определен* | Выходной код |
| nbBT_FUZZY_CONVERTER | Входной | Любое | По таблице В.1 приложения В | Непрерывные или дискретные биометрические параметры |
| | Специальный входной | 0 | — | — |
| | Специальный выходной | 0 | — | — |
| | Выходной | 1 | Не определен* | Выходной код |
| nbBT_CRYPTO_CONVERTER | Входной | Любое | По таблице В.1 приложения В | Непрерывные или дискретные биометрические параметры |
| | Специальный входной | 0 или 1 | По таблице В.5 приложения В | Код |
| | Специальный выходной | 0 | — | — |
| | Выходной | 1 | Не определен* | Выходной код |
| nbBT_HARD_INDICATOR | Входной | 1 | По таблице В.5 приложения В | Код |
| | Специальный входной | 0 | — | — |
| | Специальный выходной | 0 | — | — |
| | Выходной | 1 | По таблице В.5 приложения В | Значение жесткого индикатора |
| nbBT_SOFT_INDICATOR | Входной | Любое | Не определен* | — |
| | Специальный входной | Любое | Не определен* | — |
| | Специальный выходной | Любое | Не определен* | — |
| | Выходной | 1 | По таблице В.5 приложения В | Значение мягкого индикатора |

*Окончание таблицы Б.1*

| Тип ЭП | Тип параметра(ов) | Число параметров | Формат параметра(ов) | Интерпретация параметра(ов) |
|---|---|---|---|---|
| nbBT_ERROR_DETECTOR | Входной | Любое | Не определен* | — |
| | Специальный входной | 0 | — | — |
| | Специальный выходной | 2 | По таблице В.5 приложения В | Число обнаруженных ошибок, число исправленных ошибок |
| | Выходной | 1 | По таблице В.1 приложения В | Выходной код |
| nbBT_CONNECTOR | Входной | Любое | Не определен* | Код |
| | Специальный входной | Любое | По таблице В.5 приложения В | Внешний код связывания |
| | Специальный выходной | 0 или 1 | По таблице В.5 приложения В | Собственный код связывания |
| | Выходной | 1 | По таблице В.1 приложения В | Выходной код |
| nbBT_BIO_AUDITOR | Входной | Любое | Не определен* | — |
| | Специальный входной | Любое | Не определен* | — |
| | Специальный выходной | 0 | — | — |
| | Выходной | 1 | Не определен* | Значение индикатора атаки |
| nbBT_FUZZY_ADDRESSER | Входной | Любое | Не определен* | — |
| | Специальный входной | Любое | Не определен* | — |
| | Специальный выходной | 0 | — | — |
| | Выходной | 1 | Не определен* | Адрес в пространстве идентификаторов пользователей |
| nbBT_CODE_TRANSFORMER | Входной | 1 | По таблице В.5 приложения В | Код |
| | Специальный входной | 0 | — | — |
| | Специальный выходной | 0 | — | — |
| | Выходной | 1 | По таблице В.1 приложения В | Выходной код |
| nbBT_SECURITY | Входной | 0 | — | — |
| | Специальный входной | 0 | — | — |
| | Специальный выходной | 0 | — | — |
| | Выходной | 1 | Не определен* | — |

* Формат параметра должен определяться разработчиком НПБК. Разработчик биометрического приложения должен анализировать поле «Метаописание» соответствующего слота.

**Приложение В**
**(обязательное)**

## Формат представления параметров нейросетевого преобразователя в виде матриц векторов биометрических параметров

Формат представления параметров нейросетевого преобразования биометрия — код доступа и элементарных преобразований приведен в таблице В.1.

Таблица В.1 — Формат представления параметров нейросетевого преобразования биометрия — код доступа

| Интерпретация параметра | Значение полей матрицы ВБП (nbMatrix) | | Значение полей метаописания матрицы ВБП (nbMeta) | | |
|---|---|---|---|---|---|
| | nrows | ncols | count | format | type |
| Выходной код (во время обучения) | 1 | 1 | Слот* | nbMF_I1 | nbMT_DISCRETE_OWN |
| Выходной код (во время эмуляции) | 1 | Любое** | Слот* | nbMF_I1 | nbMT_DISCRETE_OWN |
| Непрерывные биометрические параметры «Свой» (во время обучения) | 1 | Не менее 8 | Слот* | nbMF_R32 | nbMT_CONTINUOUS, nbMT_CONTINUOUS_OWN |
| Непрерывные входные биометрические параметры «Все чужие» (во время обучения) | 1 | Не менее 100 | Слот* | nbMF_R32 | nbMT_CONTINUOUS, nbMT_CONTINUOUS_OWN |
| Непрерывные биометрические параметры (во время эмуляции) | 1 | Любое** | Слот* | nbMF_R32 | nbMT_CONTINUOUS, nbMT_CONTINUOUS_OWN |
| Дискретные биометрические параметры «Свой» (во время обучения) | 1 | Не менее 8 | Слот* | nbMF_I1 | nbMT_DISCRETE, nbMT_DISCRETE_OWN |
| Дискретные биометрические параметры «Все чужие» (во время обучения) | | Не менее 100 | Слот* | nbMF_I1 | nbMT_DISCRETE, nbMT_DISCRETE_OWN |
| Дискретные биометрические параметры (во время эмуляции) | 1 | Любое** | Слот* | nbMF_I1 | nbMT_DISCRETE, nbMT_DISCRETE_OWN |
| Выходной параметр | 1 | Любое** | Слот* | Слот* | Слот* |
| Специальные входные параметры | 1 | 1 | Слот* | Определяется типом ЭП | Слот* |
| Специальные выходные параметры | 1 | Любое** | Слот* | Определяется типом ЭП | Слот* |

* Значение, равное значению поля «Метаописание» соответствующего слота.
** Значение, равное числу примеров входных биометрических параметров.

Формат представления ограничений приведен в таблице В.2.

Таблица В.2 — Формат представления ограничений

| Значение типа ограничения (nbConstraintType) | Значение полей матрицы ВБП (nbMatrix) | | Значение полей метаописания матрицы ВБП (nbMeta) | | |
|---|---|---|---|---|---|
| | nrows | ncols | count | format | type |
| nbCT_DATA | 1 | Любое** | Слот* | Слот* | Слот* |
| nbCT_DATA_OWN | 1 | Любое*** | Слот* | Слот* | Слот* |
| nbCT_DATA_ALL | 1 | Любое*** | Слот* | Слот* | Слот* |

*Окончание таблицы В.2*

| Значение типа ограничения (nbConstraintType) | Значение полей матрицы ВБП (nbMatrix) | | Значение полей метаописания матрицы ВБП (nbMeta) | | |
|---|---|---|---|---|---|
| | nrows | ncols | count | format | type |
| nbCT_SIGMA | 1 | 1 | Слот* | Слот* | Слот* |
| nbCT_INTERVAL | 1 | 1 | Слот* | Слот* | Слот* |
| nbCT_DISPLACEMENT | 1 | 1 | Слот* | Слот* | Слот* |
| nbCT_DISCRETE_STEP | 1 | 1 | Слот* | Слот* | nbMT_DISCRETE_OWN |
| nbCT_RANGE_LIMIT | 1 | 1 | Слот* | Слот* | Слот* |
| nbCT_OWN_LAW | 1 | 1 | Слот* | nbMF_I32 | nbMT_DISCRETE_OWN |
| nbCT_ALL_LAW | 1 | 1 | Слот* | nbMF_I32 | nbMT_DISCRETE_OWN |
| nbCT_ALGORITHM | 1 | 1 | 1 | nbMF_I32 | nbMT_DISCRETE_OWN |
| nbCT_SALT | 1 | 1 | Любое | nbMF_I1 | nbMT_DISCRETE_OWN |
| nbCT_ERROR_DETECTION_RATE | 1 | 1 | 1 | nbMF_I32 | nbMT_DISCRETE_OWN |
| nbCT_ERROR_CORRECTION_RATE | 1 | 1 | 1 | nbMF_I32 | nbMT_DISCRETE_OWN |
| nbCT_CRYPTO_STRENGTH_REDUCTION | 1 | 1 | 1 | nbMF_I32 | nbMT_DISCRETE_OWN |
| nbCT_CONN_ID | 1 | Любое | Любое | nbMF_I1 | nbMT_DISCRETE_OWN |

* Значение, равное значению поля «Метаописание» соответствующего слота.
** Значение, равное числу примеров входных биометрических параметров.
*** Значение, достаточное для выполнения обучения НПБК или дообучения ЭП.

Формат представления результатов тестирования приведен в таблице В.3.

Т а б л и ц а В.3 — Формат представления результатов тестирования

| Значение режима тестирования (nbTestMode) | Значение полей матрицы ВБП (nbMatrix) | | Значение полей метаописания матрицы ВБП (nbMeta) | | |
|---|---|---|---|---|---|
| | nrows | ncols | count | format | type |
| nbTEST_MODE_E1_BIO | 1 | 1 | 1 | nbMF_I32, nbMF_R32* | nbMT_DISCRETE_OWN |
| nbTEST_MODE_E2_BIO | 1 | 1 | 1 | nbMF_I32, nbMF_R32* | nbMT_DISCRETE_OWN |
| nbTEST_MODE_E2_BIO _HUMAN_COMPROMISED | 1 | 1 | 1 | nbMF_I32, nbMF_R32* | nbMT_DISCRETE_OWN |
| nbTEST_MODE_E2_BIO _COMPROMISED | 1 | 1 | 1 | nbMF_I32, nbMF_R32* | nbMT_DISCRETE_OWN |
| nbTEST_MODE_E2_CODE | 1 | 1 | 1 | nbMF_I32, nbMF_R32* | nbMT_DISCRETE_OWN |
| nbTEST_MODE_E2_BIO_WHITE_NOISE | 1 | 1 | 1 | nbMF_I32, nbMF_R32* | nbMT_DISCRETE_OWN |
| nbTEST_MODE_E2_BIO_CORR_NOISE | 1 | 1 | 1 | nbMF_I32, nbMF_R32* | nbMT_DISCRETE_OWN |
| nbTEST_MODE_E2_CONV | 1 | 1 | 1 | nbMF_I32, nbMF_R32* | nbMT_DISCRETE_OWN |

* Допускается поддержка нескольких вариантов.

Формат представления отдельных параметров функций приведен в таблице В.4.

Т а б л и ц а В.4 — Формат представления отдельных параметров функций

| Имя параметра | Значение полей матрицы ВБП (nbMatrix) | | Значение полей метаописания матрицы ВБП (nbMeta) | | |
|---|---|---|---|---|---|
| | nrows | ncols | count | format | type |
| Индикатор отклонений входных биометрических параметров от среднего значения | 1 | 1 | число примеров | nbMF_I32, nbMF_R32* | nbMT_DISCRETE_OWN |
| * Допускается поддержка нескольких вариантов. | | | | | |

Формат представления отдельных параметров ЭП приведен в таблице В.5.

Т а б л и ц а В.5 — Формат представления отдельных параметров ЭП

| Интерпретация параметра(ов) | Значение полей матрицы ВБП (nbMatrix) | | Значение полей метаописания матрицы ВБП (nbMeta) | | |
|---|---|---|---|---|---|
| | nrows | ncols | count | format | type |
| Код | 1 | Любое* | Любое | nbMF_I1 | nbMT_DISCRETE_OWN |
| Значение жесткого индикатора | 1 | Любое* | 1 | nbMF_I32 | nbMT_DISCRETE_OWN |
| Значение мягкого индикатора | 1 | Любое* | 1 | nbMF_I32 | nbMT_DISCRETE_OWN |
| Число обнаруженных ошибок | 1 | Любое* | Любое | nbMF_I32 | nbMT_DISCRETE_OWN |
| Число исправленных ошибок | 1 | Любое* | Любое | nbMF_I32 | nbMT_DISCRETE_OWN |
| Число ошибок после исправления | 1 | Любое* | Любое | nbMF_I32 | nbMT_DISCRETE_OWN |
| Внешний код связывания | 1 | Любое* | Любое | nbMF_I1 | nbMT_DISCRETE_OWN |
| Собственный код связывания | 1 | Любое* | Любое | nbMF_I1 | nbMT_DISCRETE_OWN |
| * Во время обучения принимается за 1, во время эмуляции равно числу примеров входных биометрических параметров. | | | | | |

## Библиография

[1] ГОСТ Р ИСО/МЭК 19784-1—2007 Автоматическая идентификация. Биометрическая идентификация. Биометрический программный интерфейс. Спецификация биометрического программного интерфейса

УДК 681.18:006.354 ОКС 01.040.01

Ключевые слова: техническая защита информации, биометрия, нейросетевой преобразователь биометрия — код доступа, программный интерфейс

Редактор *В.Н. Копысов*
Технический редактор *В.Н. Прусакова*
Корректор *Л.Я. Митрофанова*
Компьютерная верстка *Л.А. Круговой*

Сдано в набор 26.06.2012. Подписано в печать 09.08.2012. Формат 60 × 84 1/8. Гарнитура Ариал.
Усл. печ. л. 5,12. Уч.-изд. л. 4,50. Тираж 86 экз. Зак. 676.

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
www.gostinfo.ru info@gostinfo.ru
Набрано во ФГУП «СТАНДАРТИНФОРМ» на ПЭВМ.
Отпечатано в филиале ФГУП «СТАНДАРТИНФОРМ» — тип. «Московский печатник», 105062 Москва, Лялин пер., 6.