|  |
| --- |
| **Система распределённого оптического распознавания символов Nebulous** |
| Спецификация |
|  |
| Техническое и функциональное описание системы. |
|  |
| **Игорь Калашников, Павел Пересторонин** |
| **20.10.2011** |
|  |

[1 Введение 3](#_Toc307357075)

[2 Обзор архитектуры 3](#_Toc307357076)

[2.1 Master 3](#_Toc307357077)

[2.2 Slave 4](#_Toc307357078)

[2.3 Gatherer 4](#_Toc307357079)

[2.4 Storage 4](#_Toc307357080)

[2.5 CLI (Command-Line Interface) 4](#_Toc307357081)

[3 Соглашения по написанию кода 4](#_Toc307357082)

[3.1 Создание нового проекта в Visual Studio 4](#_Toc307357083)

# Введение

Целью данного проекта является проектирование и реализации оптического распознавания символов на большом потоке растровых изображений. Построенная система должна принимать на вход несколько потоков растровых изображений, выделять и распознавать символы на них и сохранять результаты распознавания вместе с остальными метаданными в централизованном хранилище.

# Обзор архитектуры

Компоненты системы взаимодействуют между собой следующим образом:

Master

Slave

Storage

CLI

Slave

Slave

Gatherer

Slave

Gatherer

Gatherer

Управляющее TCP-соединение

UDP-датаграммы

Все данные между всеми узлами передаются в формате [Protocol Buffers](http://code.google.com/p/protobuf/) с помощью собственной библиотеки [Cloudy Message Passing Library](https://github.com/eigenein/cloudy), которая, кроме того, отвечает за поддерживание работы распределённой системы в целом, абстрагируясь от бизнес-логики конкретного приложения.

## Определения:

* *Доверенным* назовем узел вычислительной сети, который может с точки зрения безопасности получить результаты работы всей сети, а также получить информацию о любом другом узле сети. Все остальные узлы назовем *недоверенными*.

## Компоненты системы:

### Доверенные узлы

#### Master

Задачи Master'а:

* Отслеживать появление и исчезновение узлов в сети. Вести списки узлов и групп узлов.
* Поддерживать топологию сети.
* Выполнять аутентификацию узлов.
* Обеспечивать возможность peer-to-peer соединений между узлами, находящимися за NAT и Firewall'ами.

#### Storage

Распределенное хранилище данных о результатах распознавания.

### Недоверенные узлы

#### Slave

Вычислительный узел с поддержкой хранения состояния.

#### Gatherer

Задача Gatherer'а – передача изображений из внешней среды в систему.

### CLI (Command-Line Interface)

Консоль для управления настройками системы и контроля над её работой.

# Соглашения по написанию кода

## Создание нового проекта в Visual Studio

После создания нового проекта через мастер:

1. Изменить Default Namespace на: Nebulous.<название проекта>
2. Для всех конфигураций поставить Treat warnings as errors: All
3. Подписать сборку ключом Nebulous.snk
4. Для всех конфигураций включить Code Analysis on Build
5. Изменить AssemblyInfo.cs по образу и подобию существующих проектов

## Соглашения по именованию

Стандартные для C#.

## Скобки

1. Открывающаяся фигурная скобка ставится на отдельной строке, за исключением случая объявления атрибутов:

public Example { get; set; }

1. После закрывающейся фигурной скобки ставится пустая строка.
2. После if ставится пробел.
3. Перед и после скобок в вызове и объявлении функции не ставятся.

## Пространства имен

1. Пространство имен верхнего уровня – название проекта.
2. Не более одного в одном файле.
3. Именование соответствует каталогам проекта. Вложенность обозначается точкой.
4. Множественное число следует использовать в случае, если пространство имен объединяет некоторое количество разных, но семантически похожих сущностей. И наоборот, когда пространство имен содержит некую подсистему, стоит использовать единственное число.
5. Не используйте одно и то же имя для класса и пространства имен.

## Оформление

1. Используйте **табуляцию**, а не пробелы для отступов. Размер табуляции равен четырем пробелам.
2. При форматировании текста (кроме отступа в начале строки) используйте пробелы.
3. Избегайте строк длиннее 100 символов, переносите инструкцию на другую строку при необходимости.
4. При переносе части кода инструкций и описаний на другую строку вторая и последующая строки должны быть отбиты вправо на один отступ (табуляцию).
5. Оставляйте запятую на предыдущей строке так же, как вы это делаете в обычных языках (русском, например).
6. Избегайте лишних скобок, обрамляющих выражения целиком. Лишние скобки усложняют восприятие кода и увеличивают возможность ошибки. Если вы не уверены в приоритете операторов, лучше загляните в соответствующий раздел документации.
7. Не размещайте несколько инструкций на одной строке. Каждая инструкция должна начинаться с новой строки.

## Другое

1. Директивы using помещаются вне пространства имен.
2. Неиспользуемые using-и удаляются.