Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет телекоммуникаций

Кафедра сетей и устройств телекоммуникаций

|  |
| --- |
| *К защите допустить*: |
| Заведующий кафедрой СУ |
| \_\_\_\_\_\_\_\_\_\_\_\_В.К. Конопелько |

Пояснительная записка

к курсовой работе

на тему

**лазерная система автоматического  
сопровождения по направлению**

БГУИР КР 1-98 01 02 003 ПЗ

|  |  |  |
| --- | --- | --- |
| Студент |  | С.Р. Логинов |
| Руководитель |  | Е.Г. Макейчик |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Минск 2013

**СОДЕРЖАНИЕ**

**ВВЕДЕНИЕ**

Темой курсовой работы было выбрано написание приложения, объединяющего в себе функционал сетевых утилит *ping* и *tracert(traceroute).*

*Ping –* утилита для проверки соединений в сетях на основе TCP/IP. Утилита отправляет запросы (ICMP Echo-Request) протокола ICMP указанному узлу сети и фиксирует поступающие ответы (ICMP Echo-Reply). Время между отправкой запроса и получением ответа (RTT, от англ. *Round Trip Time*) позволяет определять двусторонние задержки (RTT) по маршруту и частоту потери пакетов, то есть косвенно определять загруженность на каналах передачи данных и промежуточных устройствах. На основе этих сведений можно сделать вывод о доступности удаленного узла и текущей нагрузке на используемые для соединения с ним каналы связи. Потеря ста процентов отправленных удаленному узлу пакетов может свидетельствовать о том, что данный узел недоступен, либо о выходе из строя промежуточного сетевого оборудования. Это может также означать, что какое-либо из осуществляющих маршрутизацию промежуточных устройств блокирует обработку запросов ICMP Echo.

*Traceroute* – это утилита, предназначенная для определения маршрутов следования данных в сетях TCP/IP. Traceroute может использовать разные протоколы передачи данных в зависимости от операционной системы устройства. Такими протоколами могут быть UDP, TCP, ICMP или GRE. Компьютеры с установленной операционной системой Windows используют ICMP-протокол, при этом операционные системы Linux и маршрутизаторы Cisco – протокол UDP. Программа *tracert* выполняет отправку данных указанному узлу сети, при этом отображая сведения о всех промежуточных маршрутизаторах, через которые прошли данные на пути к целевому узлу. В случае проблем при доставке данных до какого-либо узла программа позволяет определить, на каком именно участке сети возникли неполадки.

Эти программы являются одними из самых используемых в повседневной работе системного администратора, поэтому мне показалось хорошей идеей объединить их в общей графической оболочке. Это упростит их использование и сэкономит время системного администратора, которое можно будет потратить на более важные задачи.

1. **ОПИСАНИЕ СРЕДЫ РАЗРАБОТКИ**
   1. **Язык программирования Java**

Java – [объектно-ориентированный язык программирования](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), разработанный компанией [Sun Microsystems](http://ru.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems) (в последующем приобретённой компанией [Oracle](http://ru.wikipedia.org/wiki/Oracle" \o "Oracle)). Приложения Java обычно [компилируются](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%82%D0%BE%D1%80) в специальный [байт-код](http://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D1%82-%D0%BA%D0%BE%D0%B4), поэтому они могут работать на любой [виртуальной Java-машине](http://ru.wikipedia.org/wiki/Java_Virtual_Machine) (JVM) вне зависимости от [компьютерной архитектуры](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0).

Как и любой язык программирования, язык Java имеет свою структуру, правила синтаксиса и парадигму программирования. Парадигма программирования языка Java основана на концепции объектно-ориентированного программирования (ООП), которую поддерживают элементы языка.

Язык Java является производным языка Си, так что правила его синтаксиса выглядят во многом так же, как у Си. Например, блоки кода оформлены как модульные методы и разделены фигурными скобками ({ и }), а переменные объявляются до их использования.

* + 1. **Компилятор Java**

При программировании на платформе Java программист пишет исходный код в файлах .java, а затем компилирует его. Компилятор проверяет код на соблюдение правил синтаксиса языка, а затем записывает байт-коды в файлы .class. Байт-коды – это стандартные инструкции, предназначенные для работы на виртуальной машине Java (Java Virtual Machine – JVM). С добавлением этого уровня абстракции компилятор Java отличается от компиляторов других языков, которые создают инструкции для процессора, на котором впоследствии будет работать программа.

* + 1. **Виртуальная машина Java (JVM)**

Во время выполнения кода JVM читает и интерпретирует файлы .class и выполняет команды программы на той аппаратной платформе, для которой написана JVM. JVM интерпретирует байт-коды так же, как процессор – инструкции на языке ассемблера. Разница в том, что JVM – это программа, написанная для конкретной платформы. JVM составляет основу принципа языка Java "написано однажды – работает везде". Ваш код будет работать на любом процессоре, для которого есть реализация JVM. Реализация JVM существует для всех основных платформ, таких как Linux и Windows, а подмножества языка Java реализованы в виртуальных машинах для мобильных телефонов и встраиваемых устройств.

* + 1. **Сборщик мусора**

Вместо того чтобы вынуждать программиста следить за выделением памяти (или использовать для этого специальную библиотеку), платформа Java сама управляет памятью. Когда во время выполнения Java-приложения создается экземпляр объекта, JVM автоматически выделяет память для этого объекта из *хипа* – отведенного программе пула адресов памяти. *Сборщик мусора* Java работает в фоновом режиме, отслеживая объекты, в которых приложение больше не нуждается, и освобождая от них память. Такой подход к управлению памятью называется *неявным управлением памятью*, поскольку не требует написания какого бы то ни было кода для управления памятью. Сборщик мусора – одна из важнейших особенностей платформы Java.

* + 1. **Среда исполнения Java и набор разработчика Java**

Среда исполнения Java (Java Runtime Environment – JRE) включает в себя JVM, библиотеки кода и компоненты, необходимые для исполнения программ на языке Java. Имеются версии для многих платформ. JRE можно свободно распространять с собственными приложениями в соответствии с условиями лицензии JRE, предоставляя пользователям платформу для работы с вашим ПО. JRE включена в JDK.

Загрузив Java Development Kit (JDK), появляется доступ – в дополнение к компилятору и другим инструментам – к полной библиотеке классов готовых утилит, которые помогут решить практически любую общую задачу разработки приложений. Лучший способ получить представление о пакетах и библиотеках JDK – изучить документацию API JDK.

JDK включает в себя набор инструментов командной строки для компиляции и запуска Java-кода, включая полную копию JRE. Хотя эти инструменты, конечно, можно использовать для разработки приложений, большинство программистов ценят дополнительные функции, средства управления задачами и визуальный интерфейс IDE.

Ниже в виде таблицы перечислены возможности и преимущества языка Java:

|  |  |
| --- | --- |
| **Характеристика** | **Преимущество** |
| Независимость платформы | Java работает на большинстве аппаратных платформ и операционных систем крупных производителей либо с программным обеспечением JVM корпорации Oracle, либо через одного из многочисленных партнеров в экосистеме Java, либо как часть сообщества OpenJDK. |
| Высокая производительность | HotSpot и JRockit – это примеры проверенных технологий точной работы виртуальных машин, которые делают Java одной из самый быстрых сред программирования.  Встроенная оптимизация для многопотоковых сред делает ее еще более быстрой. |
| Легкая для освоения | Java – это язык программирования, который выбирают университеты и образовательные учреждения по всему миру.  Модель Java для управления памятью, работы в многопотоковых средах и обработки исключительных ситуаций делает его мощным языком программирования как для новичков, так и для квалифицированных разработчиков. |
| Стандартизирован | Язык Java и связанные технологии развиваются через Java Community Process – механизм, предназначенный для разработки технических спецификаций для технологии Java. |
| Господство на мировом уровне | Java – это самая популярная на планете платформа для разработки приложений. Она представляет собой динамичную экосистему для разработчиков, поддерживаемую мощными инструментами, библиотеками, примерами кода и многим другим. |
| Унифицированная среда выполнения программ | Вы можете с уверенностью развертывать приложения в таких средах выполнения программ как Java SE (на настольном компьютере), Java SE for Embedded Devices и Oracle Java Micro Edition Embedded Client. |
| Проверенная модель обеспечения безопасности | Java представляет собой современную очень надежную среду, которая идеально подходит для сетевых приложений. |

# **Среда разработки** **Eclipse**

* + 1. **Описание IDE Eclipse**

Eclipse - это расширяемая среда разработки (Integrated Development Environment, далее IDE) с открытым исходным кодом (далее OpenSource). Проект был запущен в ноябре 2001 года, когда IBM передала исходный код своего Websphere Studio Workbench стоимостью 40 миллионов долларов на OpenSource и сформировала консорциум Eclipse для управления продолжающейся разработкой. Цель создания Eclipse была сформулирована следующим образом: "разработать богатую, полнофункциональную индустриальную платформу коммерческого качества для разработки сильно-интегрированных инструментов".

Платформа Eclipse в сочетании с JDT включает многие из возможностей, которые включаются в коммерческие IDE: редактор с подсветкой синтаксиса, инкрементальная компиляция кода, потокобезопасный отладчик, навигатор по классам, менеджеры файлов и проектов, а также интерфейсы к стандартным системам контроля исходных текстов, таким как CVS и ClearCase.

**1.2.2 Поддержка сообществом**

Проект Eclipse представляет собой первую столь мощно поддержанную мировым IT-сообществом попытку создания единой открытой интегрированной платформы разработки приложений, обладающей надежностью, функциональностью и уровнем качества коммерческого продукта. Фактически эта платформа предназначена для всего и ни для чего конкретно: она представляет собой основу, имеющую блочную структуру и интегрирующую инструменты разработки ПО различных производителей для создания приложений на любом языке, с использованием любых технологий и для любой программной платформы. Вокруг проекта Eclipse в настоящее время сформировано сообщество крупнейших IT-компаний, среди которых Borland, IBM, SAP AG, RedHat и другие.

Также Eclipse предлагает множество уникальных возможностей, например рефракторинг кода, автоматическое обновление/установка кода (с помощью Менеджера Обновлений), список текущих задач, отладку модулей с помощью JUnit и интеграцию с инструментом компоновки Jakarta Ant.

Несмотря на большое число стандартных возможностей, Eclipse отличается от традиционных IDE по ряду особенностей. Наверное самое интересное в Eclipse то, что она *полностью* независима от платформы и языка. Помимо языков, поддерживаемых консорциумом в настоящий момент (Java, Cobol, C/C++), ведутся разработки по добавлению в Eclipse поддержки таких языков, как Python, Eiffel, PHP, Ruby, и C#.

Консорциум предоставляет готовые исполняемые файлы для Windows, Linux, Solaris, HP-UX, AIX, QNX и Mac OS X. Большой интерес в Eclipse представляет plug-in архитектура, а также богатый API, предоставляемый PDE, позволяющий расширять Eclipse. Добавление поддержки для нового редактора, представления или языка программирования является достаточно простым, благодаря грамотно разработанным API и большим строительным блокам, предоставляемым Eclipse.

Учитывая сотни проектов по разработке plug-in, ведущихся в настоящее время, таких индустриальных гигантов как IBM, HP и Rational, предоставляющих ресурсы, а также проектных тяжеловесов вроде Erich Gamma,

Классы, которые будут использоваться для разработки:

<http://docs.oracle.com/javase/1.4.2/docs/api/java/io/BufferedReader.html>  
<http://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Runtime.html>  
<http://docs.oracle.com/javase/1.4.2/docs/api/java/lang/Process.html>  
<http://docs.oracle.com/javase/1.4.2/docs/api/java/io/InputStreamReader.html>  
<http://docs.oracle.com/javase/1.4.2/docs/api/java/lang/System.html>