

УДК 004.78

А.Л. Єрохін<sup>1</sup>, Є.А. Пашкевич<sup>2</sup><sup>1</sup> Харківський національний університет внутрішніх справ, Харків<sup>2</sup> Харківський національний університет радіоелектроніки, Харків

## ДОСЛІДЖЕННЯ МЕТОДІВ ТА ЗАСОБІВ МОНІТОРИНГУ СЕРВЕРІВ

У статті проведено аналіз методів і засобів моніторингу серверів та сервісів сучасних інформаційних мереж. Запропоновано оригінальний підхід до реалізації програмного сервісу моніторингу серверів комп'ютерної мережі на основі аналізу можливої програмної автоматизації.

**Ключові слова:** сервіси інформаційних мереж, моніторинг серверів мережі, протокол HTTP.

### Вступ та постановка задачі

Сучасні мережеві бази даних усе більш орієнтовані не тільки на спрощення процесів зберігання на пошуку інформації, але й на реалізацію одного з важливих принципів засвоєння знань - осмислення й мотивації. Починаючи роботу з більш простими програмами і завданнями (занесення даних, реалізація переходу одного типу даних в інший і т.д.), непрофесіоналу-програмісту буде легше надалі поєднувати вже отриманий досвід і готові частини в одне ціле - програму, що виконує більш складну функцію або ж просто більше функцій. Саме в таких випадках проявляються всі переваги використання систем об'єктів.

Метою даного дослідження є аналіз методів моніторингу серверів і сервісів та розробка сервісу моніторингу серверів. Для досягнення цієї мети досліджені основні теоретичні аспекти мови програмування Java та СКБД MySQL. Сервіс призначений для моніторингу різних сервісів (наприклад, працездатність сайту - HTTP запит) або ж цілком серверів, (наприклад, ECHO запит - ping). У випадку, якщо якийсь сервіс не відповідає, або не задовольняє параметрам завдання - клієнтові відправляється повідомлення на телефон, у вигляді смс, на номер ICQ, чи на e-mail.

Сервіс моніторингу серверів і служб у мережі Інтернет надає власникам серверів статистику, коли й з якою тривалістю їх сервери або служби були доступні користувачам (online) або ж вони були недоступні (offline) по яких або причинах (наприклад, відключення від мережі Інтернет або відключення електроживлення). Власники серверів також можуть одержувати повідомлення по електронній пошті про зміну стану їх служб. Також сучасні сервіси можуть надати звіти про завантаження центрального процесора, пам'яті, мережних інтерфейсів, або протестувати веб-додаток користувача на критичні вразливості.

Таким чином, на основі аналізу існуючих проєктів моніторингу серверів необхідно розробити підхід для реалізації сервісу та засобів моніторингу серверів.

### 1. Аналіз існуючих систем

Сучасні сервіси моніторингу повинні забезпечувати наступні функції:

ping-запит - звичайний запит до сервера, із записом часу відповіді;

head-http-запит - перевірка працездатності сайту;

get-http-запит - перевірка працездатності сайту, але більш розширений, із завантаженням сторінки, і обліком часу, за який сторінка завантажилася;

head-https-запит - перевірка працездатності сайту, з передачею даних по захищеному протоколу;

get-https-запит - перевірка працездатності сайту, з передачею даних по захищеному протоколу, із завантаженням сторінки, і відліком часу - за скільки сторінка завантажилася;

ftp-запит - перевірка працездатності ftp-сервера, (анонімне з'єднання, або з авторизацією); dns-запит - перевірка доступу до сайту або сервера по dns.

Окрім цього, вони повинні забезпечувати оцінку завантаження процесора, використання пам'яті, роботи жорстких дисків системи, завантаження мережних інтерфейсів, тестування критичних місць.

Однією з першочергових завдань будь-якого центру керування мережею (ЦКМ) повинне бути зниження часу простою. Це містить у собі такі очевидні дії, як негайна реакція на відмову на ІТ-підприємстві. Також важливим є відстеження ознак проблем й складання звітів чи коректування до того, як система вийде з ладу.

При наявності розширеного набору засобів моніторингу серверів у професійно організованому ЦКМ час розв'язку проблеми значно скорочується. У деяких випадках ЦКМ може навіть відновити сервіс без залучення системного адміністратора, якщо структурування взаємин ЦКМ - ІТ це передбачає.

Моніторинг допоможе розширити знання про загальні проблеми, які можуть бути використані для зміцнення ІТ середовища в цілому.

Перевагами використання сервісів моніторингу серверів є наступні. Це зниження виробничих ви-

трат у сфері ІТ - вартість усунення навіть незначних неполадок обчислювальних сервісів може бути значною. У такому випадку, якщо ЦКМ запобігає хоча б один перебіг у роботі в рік, то витрати на його послуги окупаються наперед. Підвищення продуктивності роботи - ЦКМ повинен опікуватися про повсякденні проблеми та питаннях, які виникають із обладнанням на ІТ підприємстві, що дозволить ІТ адміністраторам займатися фактичною технічною роботою.

Це може включати такі проекти, як впровадження передового досвіду, удосконалення й планування потужностей і багато інших продуктивних заходів замість вирішення проблем, пов'язаних з усуненням проблем. Захист прибутку - моніторинг серверів підприємства дозволяє вчасно помітити критичну ситуацію й уникнути збою роботи, що напряму пов'язане з одержанням прибутку.

Однак при поломці сервісу втрата прибутку – це не єдина втрата. Клієнти перейдуть до інших поставальників послуг, якщо сервіс буде повільним або ненадійним. Важливим є збереження клієнтської бази, оскільки клієнти можуть відмовитись від послуг через ненадійність або негідну продуктивність у цілому. Наприклад, якийсь незначний 10-хвилинний простій DNS може призвести до втрати клієнта. Тому важливо в мінімальний термін виправляти не тільки великі поломки. Планування пропускну здатності мережі за наявності професійних засобів моніторингу, ЦКМ може надати величезну допомогу при здійсненні планування пропускну здатності.

У ЦКМ є в наявності спеціальне обладнання і якщо споживання ресурсів збільшується, то вони можуть із легкістю розробити оперативний план по впровадженню збільшених потужностей. Відповідність заявленому рівню обслуговування - моніторинг дозволяє ІТ організаціям перевіряти доступність сервісу в режимі реального часу, перевіряти дані по минулій доступності сервісу й використовувати цю інформацію для відповідності рівня обслуговування, гарантованому користувачам даного програмного забезпечення й клієнтові в цілому.

Для розробки ефективного підходу до моніторингу серверів досліджувались поширені існуючі сервіси.

Так, сервіс Webpinger знаходиться за адресою: <http://www.webpinger.ru/> (рис. 1).

Сервіс Webpinger знаходиться за адресою: <http://www.ping-admin.ru/> (рис. 2).

Цей сервіс має більше можливостей з перевірок протоколів та систем керування базами даних, але є небезкоштовним. Його докладні характеристики, переваги і недоліки досліджені на рис.2.

Наступний сервіс - Spae, знаходиться за адресою: <http://shalb.com/> (рис. 3).

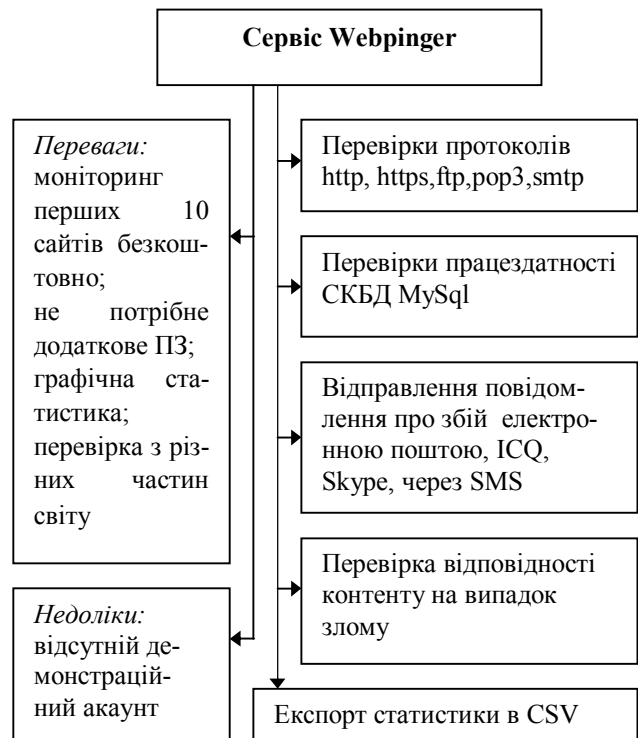


Рис. 1. Характеристики сервісу Webpinger

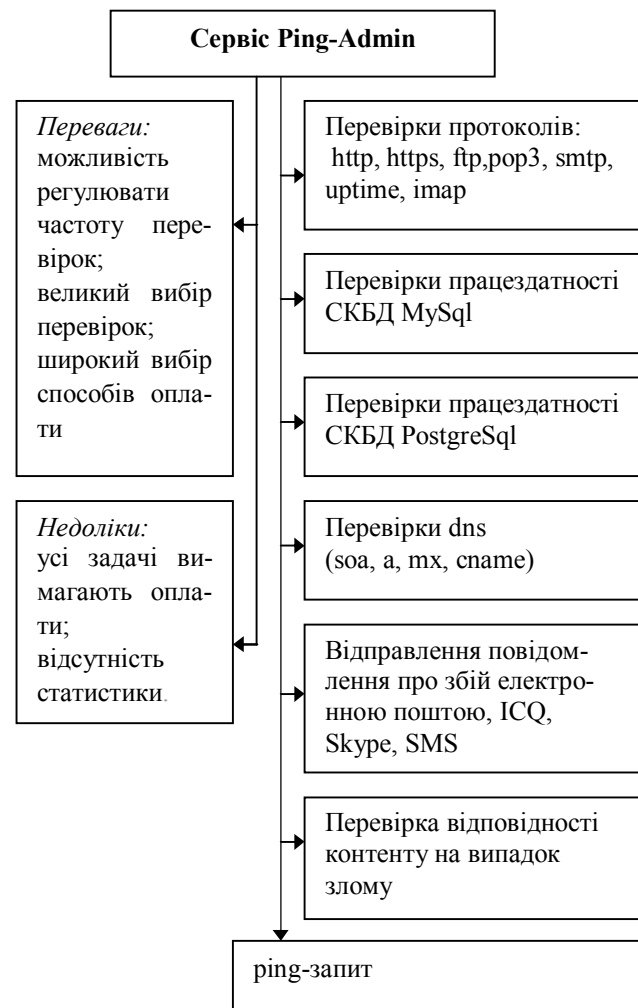


Рис. 2. Характеристики сервісу Ping-Admin

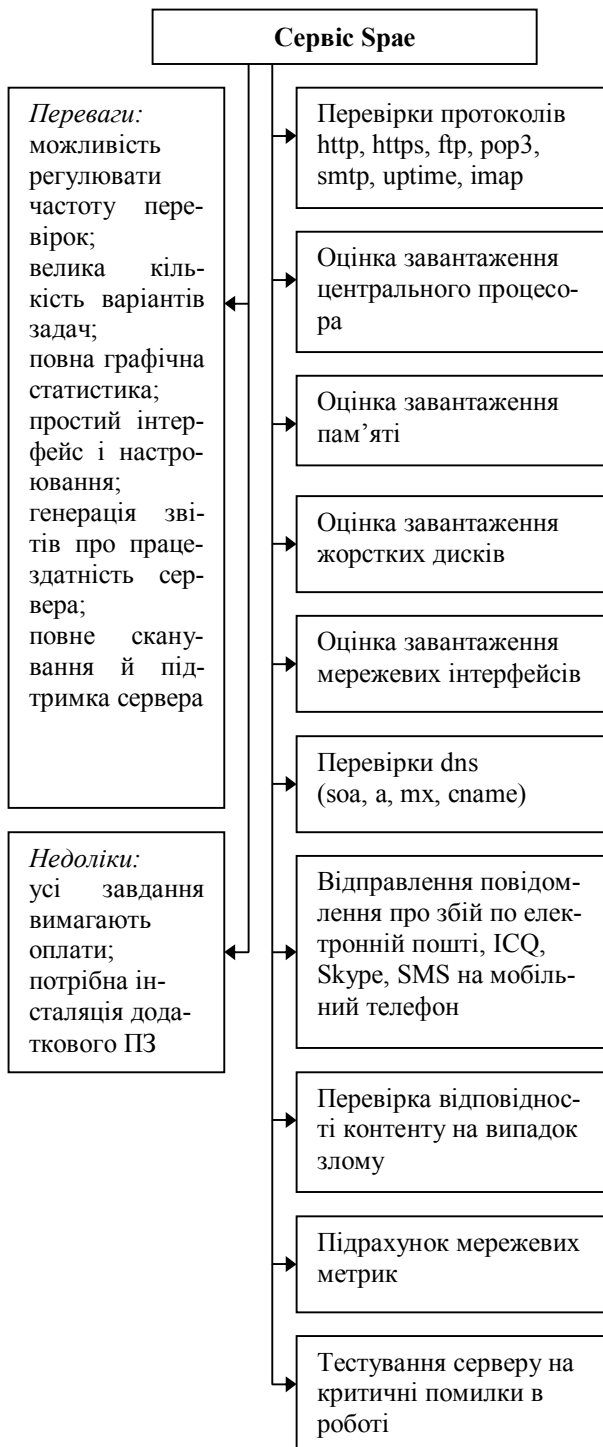


Рис. 3. Характеристики сервісу Spae

## 2. Розробка архітектури системи моніторингу

Архітектура розробленого програмного сервісу моніторингу серверів організована за допомогою пакетів (рис.4). Кожен пакет має власний набір імен для типів, який допомагає запобігати конфліктам між іменами. Структура найменування для пакетів є ієрархічною. Елементи пакета - це класи та інтерфейси, які оголошені в модулях компіляції пакета. Набір пакетів, доступних програмі на Java, визначений базовою системою, завжди повинен включати пакет java.lang [1].

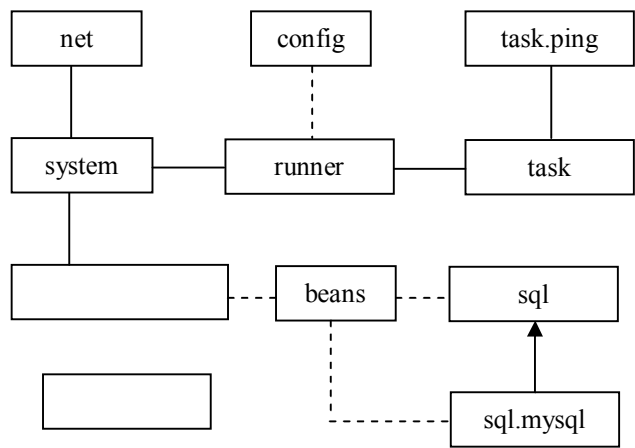


Рис. 4. Архітектурна схема системи

Класи пакету beans, включають «прості» класи-структури, для зберігання даних. Організовані ці класи по моделі POJO, тобто простий java-об'єкт, не успадкований від якогось специфічного об'єкту і не реалізуючий жодних службових інтерфейсів понад тих, які потрібні для бізнес-моделі.

Класи пакету config містять в собі класи для роботи з параметрами системи. Для завантаження параметрів, розбору параметрів, а також створення спеціальних об'єктів з пакету beans. Для реалізації даного функціонала використані класи System (для здобуття системного об'єкту), Properties (клас, що інкапсулює дані з параметрами), InputFileReader (клас, що містить функціонал для завантаження даних з файлу), з пакетів java.lang і java.util.

Класи пакету exception – класи-виключення. Використовуються для виявлення і обробки помилок, виняткових ситуацій, що виникають при роботі сервісу. Всі пакети даного класу успадковані від стандартного класу-виключення Exception з пакету java.lang. Також кожен клас з даного пакету має конструктор з параметром-рядком, для виведення додаткового повідомлення про помилку. Кожне виключення може бути «обгорнуте» в інше, таким чином, ми отримуємо повний стек виниклої помилки.

Класи з пакету net служать для зв'язку головного і дочірніх серверів сервісу. Робота даних класів заснована на сокетах. Сокети – назва програмного інтерфейсу для забезпечення обміну даними між процесами. Процеси при такому обміні можуть виконуватися як на одній ЕОМ, так і на різних ЕОМ, зв'язаних між собою мережею. Сокет – абстрактний об'єкт, що представляє кінцеву точку з'єднання. Слід розрізняти клієнтські і серверні сокети. Клієнтські сокети грубо можна порівняти з крайовими апаратами телефонної мережі, а серверні – з комутаторами. Клієнтське застосування (наприклад, браузер) використовує лише клієнтські сокети, а серверне (наприклад, веб-сервер-сервер, якому браузер посилає запити) — як клієнтські, так і серверні сокети.

Інтерфейс сокетів вперше з'явився в BSD Unix. Програмний інтерфейс сокетів описаний в стандарті Posix.1 і в тій чи іншій мірі підтримується всіма сучасними операційними системами.

Класи з пакету `runner` містять функціонал для початку роботи серверів: головних або другорядних. У методи класу входять визначення параметрів, завантаження даних, старт сервісів. Кожен клас успадкований від стандартного класу `Thread` з пакету `java.lang`, тому виконується в окремому потоці. Багатопоточність – властивість платформи (наприклад, операційної системи, JVM) або додатка, що полягає в тому, що процес, породжений в операційній системі, може складатися з декількох потоків, що виконуються «паралельно», тобто без наказаного порядку в часі. При виконанні деяких завдань таке розділення може досягти більш ефективного використання ресурсів обчислювальної машини.

Слід відмітити, що суттю багатопоточності є багатозадачність на рівні одного виконуваного процесу, тобто всі потоки виконуються в адресному просторі процесу. Окрім цього, всі потоки процесу мають не лише загальний адресний простір, але і загальні дескриптори файлів. Процес, що виконується, має як мінімум один (головний) потік. Багатопоточність (як доктрину програмування) не слід плутати ні з багатозадачністю, ні з багатопроцесорністю, не дивлячись на те, що операційні системи, що реалізують багатозадачність, як правило, реалізують і багатопоточність.

Пакет `sql` містить в собі в основному інтерфейси для роботи з об'єктами системи: користувач, сервер, кредити, зони, тарифи і так далі. Дані інтерфейси входять під доктрину шаблону DAO. Окрім інтерфейсів в пакеті знаходиться клас `DaoFactory`, який знаходить реалізацію для кожного класу, ініціалізує і повертає її. Для кожного класу можна написати безліч реалізацій, під різні СКБД, зареєструвати їх, і програма працюватиме вже з іншою базою даних, без перекомпіляції.

Пакет `sql.mysql` містить в собі реалізацію інтерфейсів з пакету `sql`, для СКБД `MySQL`. Кожному об'єкту відповідає клас, в інтерфейсі визначається необхідний функціонал, який реалізується в кожному з класів даного пакету.

Класи з пакету `system`, виконують внутрішні системні функції: призначення типів акаунтов, призначення записів про контакти, завантаження даних, призначення вибраних тарифів, а також інші. Слід зазначити клас `VenatorGarbageCollector`, який виконує роль контролера за звільненням пам'яті. Даний клас через певний проміжок часу, який вказаний в параметрах сервісу, намагається викликати складальник сміття. Коли ж насправді запуститься складальник сміття, сказати неможливо, оскільки це завдання виконуваного середовища.

Оскільки сервіс умовно роздільний на дві частини (ядро і інтерфейс) безпосередньо не зв'язаних між собою, виникла проблема для підтримки актуальності даних в ядрі системи. Тому був створений функціонал, який знаходиться в пакеті `system.updater`. Даний функціонал повинен забезпечувати перевірку даних в базі даних з актуальними завантаженими в ядрі, і при необхідності змінити їх.

Класи з пакету `task`, містять функціонал, який безпосередньо виконує завдання системи. Перевірку працездатності різних сервісів. Пакет містить інтерфейси для реалізації завдань, «фабрику» виконавців завдань, а також інші класи. В разі помилки при виконанні завдання, вона реєструється і користувачеві вирушає повідомлення на вибраний контакт.

Класи з пакету `task.ping`, містять допоміжні функції для завдання "ping". Оскільки `java` – кросплатформенна мова, даний сервер може бути виконаний на різних операційних системах. Але утиліта `ping` кожної ОС може відрізнитися, тому в даному пакеті міститься функціонал, який забезпечує визначення ОС, а також розбір результату від виконання утиліти.

Для розробки програмної системи були використані шаблони [2 – 6].

Dao-шаблон шаблон використовується для організації роботи системи і СКБД. DAO (data access object) – це об'єкт, який надає абстрактний інтерфейс до якого-небудь типу бази даних або механізму зберігання. Певні можливості надаються незалежно від того, який механізм зберігання використовується і без необхідності спеціальним чином відповідати цьому механізму зберігання. Цей шаблон проектування застосовний до безлічі мов програмування, використовується у великій кількості програмного забезпечення, що потребує зберігання інформації і до більшої частини баз даних, але традиційно цей шаблон пов'язують з додатками на платформі `Java Enterprise Edition`, що взаємодіють з реляційними базами даних через інтерфейс `JDBC`, тому що він з'явився в рекомендаціях від фірми `Sun Microsystems`.

Архітектурно DAO реалізується також за допомогою шаблону «фабрика». Шаблон «фабрика» – служить для створення об'єктів, у випадку якщо заздалегідь не відомо який об'єкт може знадобитися в даний момент.

У системі використовується в трьох випадках `TaskFactory` – «фабрика» виконавців завдань, `DaoFactory` – «фабрика» класів реалізацій, для роботи з СКБД, `PingParserFactory` – «фабрика» об'єктів для визначення ОС, а також розбору даних відповіді від утиліти `ping`.

Таким чином `*factory` – шаблон проектування, що породжує, надає підкласам інтерфейс для створення екземплярів деякого класу. У момент створення спадкоємці можуть визначити, який клас інстанціювати. Іншими словами, фабрика делегує

створення об'єктів спадкоємцям батьківського класу. Це дозволяє використовувати в коді програми не специфічні класи, а маніпулювати абстрактними об'єктами на більш високому рівні. Також відомий під назвою віртуальний конструктор.

Шаблон Singleton призначений для створення гарантій того, що у класа є лише один екземпляр, і надає до нього глобальну точку доступу. Важливо те, що можна користуватися саме екземпляром класу, оскільки при цьому у багатьох випадках стає доступною ширша функціональність. До описаних компонент класу можна звертатися через інтерфейс, якщо така можливість підтримується мовою програмування.

### 3. Розробка структури бази даних

Гнучкість СКБД MySQL забезпечується підтримкою великої кількості типів таблиць: можна обирати як таблиці типу MYISAM, що підтримують повнотекстовий пошук, так і таблиці INNODB, що підтримують транзакції на рівні окремих записів. В СКБД MySQL є спеціальний тип таблиць EXAMPLE, що демонструє принципи створення нових типів таблиць. Розглянемо короткий опис таблиць:

- account\_type – таблиця містить в собі дані про типів акаунтах користувачів;
- contact – таблиця містить записи про контакти користувачів;
- contact\_type – таблиця містить записи про типи контактів для користувачів;
- rates – таблиця містить інформацію про вибрані користувачами тарифи;
- rates\_types – таблиця містить інформацію про типи тарифів;
- servers – таблиця містить в собі інформацію про сервери користувачів;
- system\_server – таблиця містить інформацію про системні сервери сервісу;
- system\_zone – таблиця містить інформацію про системні зони;
- tasks – таблиця містить в собі інформацію про вибрані завдання для кожного конкретного серверу;

- tasks\_done – таблиця містить інформацію про виконані завдання, а також результат виконання;
- tasks\_types – таблиця містить інформацію про типів завдань;
- users – таблиця містить в собі інформацію про користувачів системи.

### Висновки

В результаті проведених досліджень були розглянуті та проаналізовані різні методи моніторингу серверів та сервісів. Також були проаналізовані існуючі проекти, їх достоїнства та недоліки, для того, щоби мати представлення про архітектуру таких систем.

Такі системи призначені для моніторингу серверів та сервісів в мережі Інтернет. Такий сервіс надає власникам серверів статистику, коли і з якою тривалістю їх сервери або служби були доступні користувачам (online) або ж вони були недоступні (offline) по яким причинам (наприклад, відключення від мережі Інтернет або відключення електроживлення). Власники серверів також можуть отримувати повідомлення по електронній пошті про зміну стану їх служб.

### Список літератури

1. Культин Н.Б. *Java в задачах и примерах* / Н.Б. Культин. – СПб.: БХВ, 2007. – 239 с.
2. Лабор В.В. *Создание приложений под Windows с помощью Java* / В.В. Лабор. – М.: Харвест, 2003. – 385 с.
3. Азуров П. *Разработка приложений, Java 1.5* / П. Азуров. – СПб.: БХВ, 2008. – 480 с.
4. Фролов А.В. *Язык Java. Самоучитель* / А.В. Фролов. – М.: Диалог-Мифи, 2003. – 560 с.
5. Карли Ватсон. *Java / Ватсон Карли*. – М.: Лори, 2005. – 862 с.
6. Трозсен Э. *Java 6, четвертое издание* / Э. Трозсен. – М.: Аpress, 2007. – 1370 с.

Надійшла до редколегії 23.02.2011

**Рецензент:** д-р техн. наук, проф. І.П. Захаров, Харківський національний університет радіоелектроніки, Хаарків..

### ИССЛЕДОВАНИЕ МЕТОДОВ И СРЕДСТВ МОНИТОРИНГА СЕРВЕРОВ

А.Л. Ерохин, Е.А. Пашкевич

*В статье представлен анализ и сравнение методов и средств мониторинга серверов и сервисов. Также разработана программная служба мониторинга серверов компьютерной сети, на языке Java. Программный продукт разработан на основе анализа возможной программной автоматизации. Программная система сохраняет информацию о тарифах, потенциальных задачах, а также тех, что уже были выполнены.*

**Ключевые слова:** сервисы информационных сетей, мониторинг серверов сети, протокол HTTP.

### RESEARCH INTO METHODS AND TOOLS FOR MONITORING SERVERS

A.L. Yerokhin, E.A. Pashkevich

*The article deals an analysis and comparison of methods and tools for monitoring servers and services. Additionally, the development of software monitoring service server computer network, in Java. The software product is based on analysis of the possible software automation. This product will store information on pricing, potential tasks, as well as those that have already been implemented.*

**Keywords:** services of informative networks, monitoring of servers of network, protocol of HTTP.