

Енергетика, контроль та діагностика об'єктів нафтогазового комплексу

УДК 622.24.05-004.4

РОЗРОБКА МЕРЕЖЕВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТЕХНОЛОГІЄЮ ДЖАВА ДЛЯ МОНІТОРИНГУ ТЕХНІЧНОГО СТАНУ МАГІСТРАЛЬНИХ ГАЗОПРОВІДІВ

І. З. Лютак, Р. М. Федорак

*ІФНТУНГ, 76019, м. Івано-Франківськ, вул. Карпатська, 15, м. Івано-Франківськ, 76019
public@nuing.edu.ua*

Для надежной эксплуатации магистральных газопроводов необходимо иметь информацию об их техническом состоянии в реальном времени. Эффективным инструментом при эксплуатации газопроводов является разработка сетевого программного обеспечения с возможностью пересмотра информации о техническом состоянии в конкурентном режиме через общедоступный веб-интерфейс. В статье показан подход к проектированию и реализации такого программного обеспечения, которое построено с учетом существующих современных методов и средств, в частности Универсального языка моделирования и системы тестирования клиент-серверной части программы, и может надежно эксплуатироваться

For reliable exploitation of main gas pipelines it is necessary to have the real-time information about their technical state. During exploitation of gas pipelines development of network software is an effective instrument with possibility of revision the information of technical state in the competition mode through a popular web interface. In the article approach is rotined to planning and realization of such software that can be reliably exploited. Moreover, it is built according to existent modern methods and facilities, in particular by both the Universal Modelling Language and an approach of testing of client - server part of software.

Вступ

Розвиток технічного прогресу, Інтернет технологій, дає змогу використовувати нові програмні засоби для удосконалення моніторингу технічного стану газопроводів, створення мережеских трансакцій, розподілених систем, доставку сервісів тощо. Як результат, на даний час існує значний попит на вказані програмні продукти, зокрема для газотранспортної галузі України. Такі програмні продукти повинні з одного боку бути безпечними, а з іншого – мати значні можливості щодо перегляду звітності та можливості управління через Інтернет. Це значно змінює вимоги щодо підходу до їх планування та реалізації, оскільки використання традиційних технологій програмування не дає можливості забезпечити вимоги щодо якості та відведеного часу для розробки веб-програм [1].

Для реалізації системи моніторингу, що розробляється нами, вибрано як мову програмування платформу Джава, оскільки вона має

низку таких переваг над іншими технологіями мережевого програмування:

- незалежність коду програми від веб-сервера та операційної системи;
- багатий набір бібліотек мережеских сервісів, що є частиною платформи Джава;
- значний набір класів та бібліотек роботи із графічним інтерфейсом;
- безкоштовність платформи Джава та засобів програмування;
- значний набір технологій мережевого програмування корпоративного рівня, що дає змогу значно збільшувати масштабування розроблених програм без затрат на їх перепрограмування;
- широкий вибір веб-серверів.

Особливістю розроблення систем моніторингу технічного стану на базі веб-програм від виключно програмування для Інтернет є необхідність врахування конкурентного доступу до вимірювальних ресурсів, оскільки час запиту

веб-програми до пристроїв і тривалість виконання ними команд значно відрізняються, а тому існує можливість одночасного надходження несумісних команд. При недостатньому проектному вирішенні конкурентних запитів можна в непередбачуваний момент часу отримати невірні результати чи вивести цю систему з ладу, що є неприпустимим з огляду на значення об'єкта моніторингу. Тому однією із супутніх задач розроблення системи моніторингу повинно бути проектне рішення щодо безпечної обробки конкурентних запитів.

Додатковим аспектом розробки веб-програм є важливість системи тестування програмних модулів для забезпечення надійності виконання як мережевих розподілених сервісів, так і забезпечення синхронізації із вимірювальними пристроями. Відсутність тестування веб-програм викликає питання щодо надійності всієї системи моніторингу [2]. На даний час існує низка підходів до тестування веб-програмного забезпечення, яке базується на покритті тестами веб-сторінок та їх взаємозв'язків [3, 4, 5].

Одним із напрямків розвитку тестування веб-програм є розроблення та удосконалення бібліотек для тестування, що адаптує логіку покриття тестами традиційних програм. До такого підходу включається розроблення тестових наборів підпрограм, виконання цього набору та моніторинг результатів із послідовним усуненням похибок. Такий підхід є надзвичайно трудомістким та містить велику ймовірність пропуску критичних частин логіки веб-програми. Іншим підходом є абстракція неструктурованого коду веб-програми за допомогою об'єктів

та їх взаємозв'язків. Такий напрямок є найбільш доцільним, оскільки може покривати всю складність логіки передачі управління веб-програми використовуючи інтуїтивно зрозумілі концепції. Недоліком такого підходу є те, що неможливо розробити єдиний шаблон тестування для всіх типів веб-програм. На даний час широкого розповсюдження набули такі програми тестування сторінок серверної частини Джава (ССЧДж) як Http Unit та Sactus, проте їх використання лише автоматизує роботу з написання тестів, але не гарантує покриття ними всього коду веб-програми.

Враховуючи вище сказане актуальним є розроблення системи моніторингу технічного стану магістральних газопроводів, яка володітиме значною стабільністю та визначеністю поведінки внутрішніх алгоритмів, що має бути забезпечено надійною системою тестування основної частини веб-програми.

Вимірювальні пристрої системи моніторингу

Система моніторингу технічного стану газопроводів складається із ряду датчиків для контролю таких параметрів як товщина стінок та величина напружень (рис. 1), оскільки ці величини характеризують експлуатаційні параметри.

Як датчі товщини ми використали ультразвукові датчі, які генерують пластинчасті хвилі. Перевага такого підходу полягає в тому, що можна контролювати стаціонарними датчами певну ділянку газопроводу, довжина якої може складати декілька десятків метрів. Такі датчі

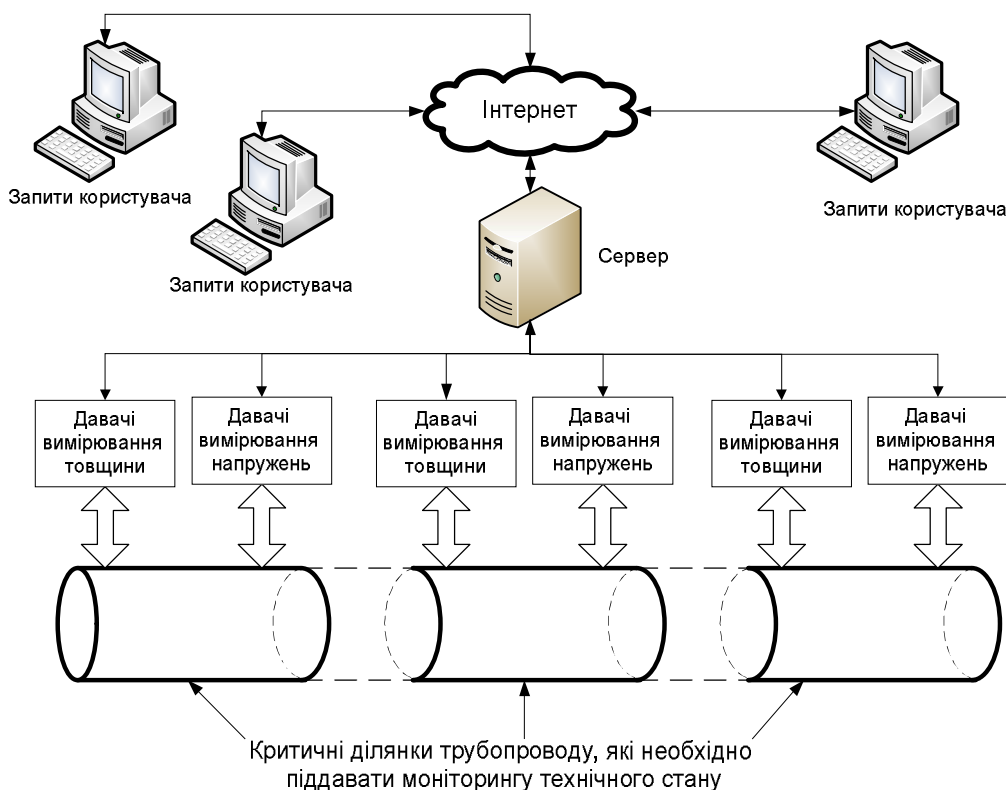


Рисунок 1 – Схема системи моніторингу технічного стану газопроводів

потрібно монтувати в місцях з підвищеною кількістю деструктивних чинників впливу на газопровід, наприклад, технологічні коліна, місця кріплення тощо.

Як давачі визначення напружень нами використано як ультразвукові, так і тензометричні, оскільки це значно підвищує достовірність вимірюваних результатів.

Вимірювана інформація передається до сервера, який здійснює кінцеву обробку результатів та надає оброблену інформацію на веб-сервер для її перегляду через Інтернет. Користувач Інтернет, який заходить авторизовано на веб-сервер для перегляду інформації про технічний стан газопроводу, ініціалізує нову серію вимірювань для вибраної ним ділянки трубопроводу. В цьому випадку веб-сервер подає сигнал вимірювальним пристроям для генерації сигналів та початку вимірювань. Через визначений період часу вимірювана інформація надходить до користувача веб-сервера в обробленому вигляді.

Зважаючи на те, що мережеве програмне забезпечення є активним, тобто ініціалізує вимірювання і надає повну інформацію про технічний стан газопроводу, важливим є забезпечення надійної системи авторизації користувача та обробки конкурентних запитів в реальному часі.

Моделювання напрямку виконання

Забезпечення надійності роботи системи мережевого програмного забезпечення можна досягти шляхом правильного підходу до проектування всієї програми та покриття її тестами. Для проектування програмного забезпечення нами використано уніфіковану мову програмування (УМП), яка є стандартним інструментом

для створення графічного представлення взаємозв'язку різноманітних компонентів та логіки виконання алгоритмів програми. Перевагами використання УМП є значна кількість напрацьованих та системних рішень на її основі, а також те, що вона здатна моделювати будь-які системи - від інформаційних систем масштабу підприємства до розподілених веб-програм та систем реального часу.

Для наглядності процесу розроблення мережевого програмного забезпечення представимо спрощену систему авторизації користувача на веб-сервер (рис. 2). Така схема дає змогу концептуально визначити складність частини коду та його об'єм, що, в свою чергу, дає змогу визначити приблизні терміни та кількість ресурсів для її реалізації. Ще однією перевагою представлення частини коду програми у вигляді УМП є визначеність її структури на рівні сутностей, класів та об'єктів, які їх реалізують, що дає змогу встановити ключові сигнатури для методів створених класів та об'єктів. Це, в свою чергу, дає змогу використовувати для реалізації коду програми декілька фахівців одночасно і тим самим значно зменшити загальний час розроблення мережевого програмного забезпечення.

Проте, зображати відповідальні частини коду тільки за допомогою статичних подань є недостатнім, оскільки, як можна побачити з рис. 2, така схема є складною, що не дає змоги із впевненістю відстежувати всі можливі взаємозв'язки при конкурентному використанні при її експлуатації. Тому важливим є представити частину коду програми її динамічним аспектом. Для цього, як правило, використовується діаграма взаємодії УМП. За допомогою діаграми взаємодії можна представити моделі

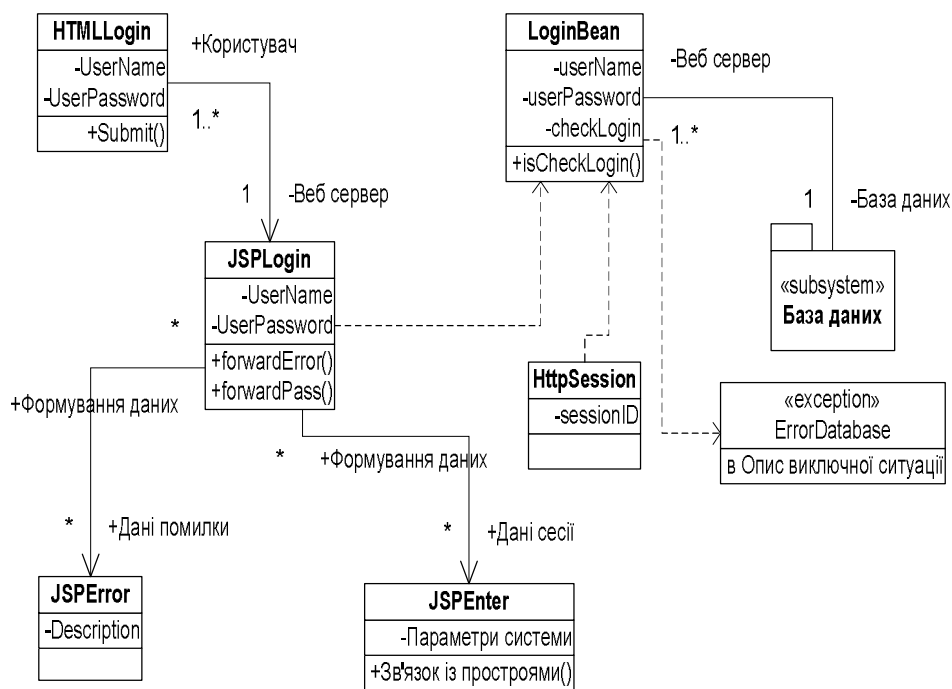


Рисунок 2 – Структурна схема УМП блока авторизації користувача веб-сервером

конкретних і прототипних екземплярів класів, компонентів і вузлів, а також повідомлення, якими вони обмінюються, для забезпечення виконання конкретного заданого сценарію роботи програми.

Структуризована діаграма-послідовність має такі властивості:

- акцентує увагу тільки на одному аспектові динаміки системи;
- має необхідний і достатній набір компонентів, важливих для розуміння одного аспекту динаміки системи;
- має необхідний і достатній набір компонентів, які відповідають одному рівневі абстракції і необхідні для розуміння системи;
- має достатню кількість компонентів, щоб однозначно виразити семантику частини коду програми.

На рис. 3 зображено динамічний аспект авторизації користувача. Як можна побачити з рис. 3, спершу активізується гіпертекстова сторінка форми для внесення інформації про логін і пароль користувача розроблюваної системи. Така інформація не була очевидною, якщо аналізувати рис. 2. Додатковою інформацією є час життя об'єктів та їх активація. Так, можна побачити, що два об'єкти - HttpSession та ErrorDatabase динамічно створюються і не є постійними протягом всієї активації коду програми, більш того, об'єкт ErrorDatabase створюється тільки при виконанні умови, а інший об'єкт створюється безумовно. Така інформація є важливою при аналізі стабільності та надійності виконання частин коду програмного забезпечення.

Для завершення точного аналізу стабільності та надійності коду програми необхідно розробити систему тестів.

Розроблення системи тестування

Розроблення системи тестування необхідно проводити з врахуванням як структури коду програми, так і динаміки її виконання. Така інформація дає змогу зрозуміти логіку роботи програми і змодельовати потік даних, необхідний для написання тестів.

Тестування потоку даних в основному фокусується на визначенні змінних та їх потенційному використанні або для аналізу та пошуку похибок, або ходу виконання, або результатів роботи коду програми. В загальному розрізняють два види використання змінних: а) для обчислення, б) предикатне [3]. Змінні типу а) використовуються для виразів, що обчислюють дані, або у виразах, що виводять ці дані назовні розглядуваного програмного блоку, який розглядається.

Змінні типу б використовуються у предикатних виразах.

Важливим аспектом для розробки тестів програмного забезпечення є визначення шляху тестування програми. Він не у всіх випадках буде прямувати за напрямками, взятими із діаграм-послідовностей, оскільки діаграми розробляються для визначення логіки взаємодії об'єктів на досягнення проектованого результату, а шляхи тестування визначаються на основі реального коду реалізації програми. Для оптимального визначення шляху тестування програми необхідно використовувати так зва-

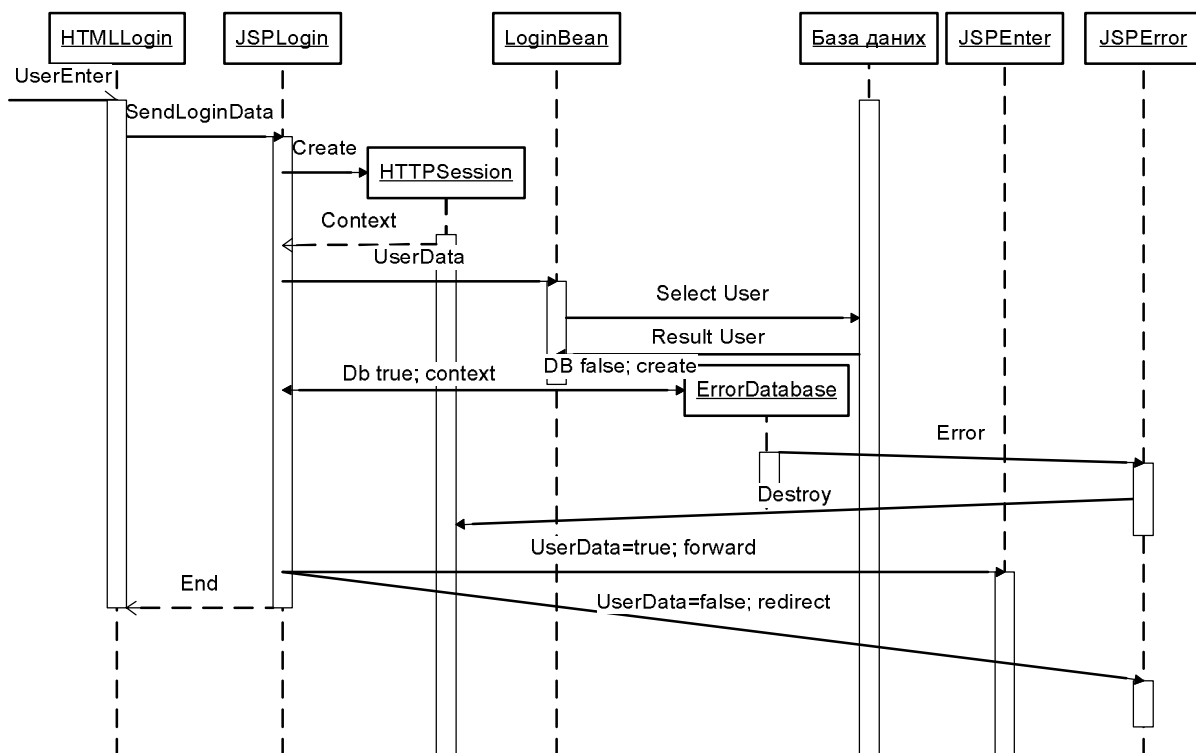


Рисунок 3 - Діаграма-послідовність УМП блока авторизації користувача веб-сервером

ний підхід визначення-використання змінної, що полягає у побудові шляху від декларування змінної до її використання без врахування будь-яких проміжних етапів передекларування [6]. Для визначення правильного шляху тестування необхідно врахувати і приховані змінні технології JSP. Такі приховані змінні як `request` та `response` відкривають доступ до ресурсів мережових пакетів, реалізованих у платформі Джава, без наявного їх декларування. Інші змінні дають доступ до потоків вводу/виводу інформації та впливають на напрям виконання сторінок JSP. За допомогою прихованих змінних можна обробляти всі запити гіпертекстових сторінок однієї сесії користувача, що дає змогу передавати дані між окремими HTML сторінками та модулями, між JSP сторінками, що набагато ускладнює розуміння можливих шляхів тестування. Одним із найбільш вживаних прихованих об'єктів є `session`, який працює як контейнер для зберігання даних представлених параметрами "ім'я – значення", що використовується багатьма JSP сторінками і може бути модифікованим будь-якою із них.

Ще одним аспектом використання прихованих змінних є методи передавання управління до інших JSP сторінок, наприклад, `sendRedirect()` об'єкту `response`. Передавання управління між сторінками JSP не тільки змінює хід виконання коду програми, але може бути використаним при передаванні даних та взаємодії JSP сторінок.

Додатково до прихованих об'єктів в JSP технології використовуються стандартні теги виконання операцій, таких як `<jsp:useBean>` та `<jsp:forward>`. Ці теги асоційовані із обробниками коду, які дають змогу змінювати дані та передавати управління ходом виконання коду програми іншим JSP сторінкам. Як результат, при написанні тестів до мережового програмного забезпечення, написаного з використанням технології Джава, необхідно враховувати всі приховані об'єкти та теги незалежно від їх використання чи невикористання в конкретній JSP сторінці.

Приховані об'єкти технології JSP та теги, які необхідно враховувати при складанні тестування програмного забезпечення є такими:

- `request` - може використовуватись при передаванні інформації між JSP сторінками та при обробці даних користувача, введених у клієнтському запиті гіпертекстової сторінки;

- `response.sendRedirect()` - може змінювати напрям виконання операторів JSP сторінок, сервлетів, а також передавати інформації між серверними компонентами;

- `jsp:useBean` - може передавати інформацію між JSP сторінками та іншими модулями мережової програми, зокрема базою даних та портами вводу/виводу персонального комп'ютера до якого під'єднані вимірювальні чи передавальні пристрої;

- `jsp:forward` - може змінювати напрям виконання операторів та передавати інформацію між JSP сторінками та сервлетами;

- `jsp:include` - може використовуватись при інтерактивній взаємодії JSP сторінок за наявності додаткових параметрів у тега;

- `jsp:setProperty` - встановлює параметри об'єктів JSP сторінок;

- `jsp:useProperty` - зчитує інформації із об'єктів в JSP сторінках;

- `session` - контейнер, який використовується для передавання інформації між JSP сторінками та сервлетами на рівні сесії;

- `application` - отримує доступ до контекста конфігурації сервлетів та серверної частини мережового програмного забезпечення.

З врахуванням особливостей прихованих об'єктів та тегів JSP сторінку можна розглядати, з точки зору складання тестів, як набір процедур, оскільки важливими спільними рисами їх є єдина точка входу та доступ до декларованих процедур тільки всередині сторінки. Відмінністю JSP сторінок від традиційних програм є відсутність визначеної ієрархії викликів, оскільки послідовність виклику JSP сторінок визначається користувачем під час користування мережевою програмою.

Для врахування особливостей тестування JSP сторінок використаємо підхід побудови графа контролю ходу виконання операторів, анованих `def/use` мітками [4]. Вузли такого анованого графу представляють команди JSP сторінки, а ребра - напрям потоку виконання команд. За допомогою такого графа можна прослідкувати всі зміни, які трапляються із змінними та прихованими об'єктами. Блок `def` визначає декларацію змінних, а блок `use` визначає використання задекларованих змінних у процедурах.

Висновки

У цій роботі представлено комплексний підхід до розробки мережового програмного забезпечення моніторингу технічного стану газопроводів. В запропонованому підході використано технологію Джава та JSP. При проектуванні програмного забезпечення нами розглянуто комплексний підхід як до використання сучасних методів при побудові модулів серверної частини з використанням УМП, так і до тестування із врахуванням прихованих об'єктів. Такий підхід, що враховує конкурентний доступ до ресурсів серверної частини на етапі проектування та інші особливості функціонування системи моніторингу, значно зменшує час розроблення та підвищує надійність системи загалом.

Література

1 Pressman R.S. Can internet-based applications be engineered? IEEE Software (1998) pp. 104–110.

2 Ginige A., Murugesan S. Web engineering: an introduction, IEEE Multimedia 8 (1) (2001) pp. 14–18.

3 Benedikt M., Freire J., Godefroid P. VeriWeb: Automatically Testing Dynamic Web Sites, in: Proceedings of the 11th International World Wide Web Conference, May 2002, Hawaii, USA.

4 Chien-Hung Liu. Data flow analysis and testing of JSP-based Web applications, Information and Software Technology 48 (2006) pp. 1137–1147.

5 Kallepalli C., Tian J. Measuring and modeling usage and reliability for statistical web testing, IEEE Trans. Software Eng. 27 (11) (2001) pp. 1023–1036.

6 Rapps S., Weyuker E.J. Selecting software test data using data flow information, IEEE Trans. Software Eng. SE-20 (No. 4) (1995) 385–403.

УДК 550.34

ИСПОЛЬЗОВАНИЕ ХРОМАТОГРАФИЧЕСКОГО МЕТОДА ДЛЯ ИДЕНТИФИКАЦИИ НЕФТИ

В.И.Фаловский, Р.А.Минахметов

*ГИПРОВОСТОКНЕФТЬ, Россия, 443041, г. Самара, ул. Красноармейская, 93
тел. (846) 3334696, факс (846) 2792058, e-mail gipvn@gipvn.ru*

Предложен метод, позволяющий выявить специфические особенности состава нефти. При этом использованы различия в распределении углеводородов нормального и изомерного строения. Подобранные критерии для количественного определения содержания отдельно взятой нефти в смеси.

Method allowing to identify specific features of oil composition has been proposed. In this regard differences in distribution of normal and isometric hydrocarbons have been used. Criteria for quantitative determination of individual oil content in mixture have been selected.

Проблема идентификации нефти возникает в различных случаях, связанных с ее добычей и транспортом.

В некоторых случаях эксплуатация залежей оказывается более эффективной при совместном дренировании продуктивных горизонтов. При этом разделение продукции по пластам становится задачей, требующей постоянного решения.

В ходе разработки старых месторождений залежь может переходить на режим истощения. В коллекторе образуется газонасыщенная область. В условиях резкого изменения термобарических условий в воронке депрессии возникает совместная фильтрация газовой и нефтяной фаз и массообмен между ними. Показателем такого хода событий является обогащение нефти легкими фракциями (газовым конденсатом). Эти же явления возможны при прорыве газа, если первоначально нефтяная залежь имела газовую шапку.

Скважина может быть прострелена на один объект разработки, но вследствие негерметичности эксплуатационной колонны возможны перетоки из других пластов.

Во всех перечисленных случаях необходимо своевременное получение информации об изменении состава продукции скважин, выяснение его причин и принятие соответствующих решений с целью оптимизации процесса разработки месторождения.

Рациональная организация работы систем внутри- и межпромыслового транспорта нефти также требует решения вопросов, подобных описанным выше.

Поэтому актуальным является выявление взаимосвязи между характеристиками состава для достоверной идентификации нефти.

В разное время в качестве специфических признаков нефти рассматривались такие свойства, как коэффициент светопоглощения [1], равнообъемные «анилиновые» точки [2]. Последние отражают определенное соотношение ароматических, нафтеновых и парафиновых углеводородов [2,3]. Вследствие многостадийности пробподготовки, влияния различных факторов, эти методы, несмотря на длительную историю своего развития, находят в настоящее время ограниченное применение.

Данные по микроэлементному составу нефти также используются для идентификации нефти и выработки рекомендаций по эксплуатации совместных скважин [4, 5].

В случае светлых нефтей с незначительным содержанием асфальтенов, смол и микроэлементов, а также близким структурно-групповым составом вышеприведенные методы практически не применимы.

В настоящей работе для выявления специфических признаков нефти использована капиллярная газовая хроматография.

Газохроматографический метод на капиллярных колонках позволяет выявлять особенности состава нефти, характерные только для данного нефтяного пласта или стабильного потока. В связи с этим стал употребляться криминалистический термин «отпечаток пальца» применительно к хроматограмме. Для получения надежных результатов, не искаженных различными вспомогательными операциями (фракционирование, депарафинизация, деасфальтизация и др.), обычно применяемыми при изучении состава нефтей, наилучшим вариантом является введение в хроматограф «сырой» нефти.