

маркерів в позиціях і забезпечує управління ходом виконання програми від інструментальної машини). Цей модуль є ПЗ клієнтської машини та має назву NetManager.exe.

- Отримання байт-коду мови Java, згенерованого ПЗ інструментальної машини, і запуск інтерпретатора Java + <задача два>. Цей модуль є ПЗ клієнтської машини та має назву Programm.java.

В якості мови програмування конструктора обрано об'єктно-орієнтовану мову C ++, а в якості середовища програмування Borland C ++ Builder.

Опишемо структуру реалізованих програм. Програма PetryNet.exe складається з наступних модулів:

- головний модуль;
- модуль редагування атрибутів вершини;
- модуль опису змінних програми;
- модуль компіляції;
- модуль мережевої взаємодії

Програма NetManager.exe встановлюється на клієнтській машині та складається з наступних модулів:

- модуль мережевої взаємодії;
- модуль запуску програми.

Програма Programm.java написана мовою програмування Java та є оболонкою для програми управління об'єктом.

Висновок

У роботі розроблено та реалізовано візуальний конструктор для програмованих мікроконтролерів PC-архітектури з використанням апарату мереж Петрі, який забезпечує можливість генерування програми з введеної мережі і можливість динамічної імітації її функціонування.

В якості мови програмування конструктора обрано об'єктно-орієнтовану мову C ++, а в якості середовища програмування Borland C ++ Builder.

Список використаних джерел

- IEC 1131-3 (Язика програмування ПЛК) в вопросах и ответах, 2007. Режим доступа: [ftp://ftp.cle.ab.com/stds/iec/sc65bwg7tf3/html/faq.htm]
- ULTRALOGIC – система подготовки программ для промышленных контроллеров. – Современные технологии автоматизации., 1997. – №3.
- Коровкин Б.Г. Системы программного управления промышленными устройствами и робототехническими комплексами. Учебное пособие для вузов. Л.: Энергоатомиздат. 1990 – 352с.: ил.
- Нортон П., Станек У. Руководство Питера Нортон. Программирование на Java. В 2-х книгах. Книга 1. М.: «СК Пресс», 1998.-552с.: ил.

УДК 004.4'242

СИСТЕМА ТРАНСФОРМАЦІЇ ТЕКСТУ ПРОГРАМ НА ПРОТОТИП-ОРІЄНТОВАНИХ МОВАХ ЗАСОБАМИ UML

Шпінталь М.Я.¹⁾, Кучварський А.І.²⁾

Тернопільський національний економічний університет

¹⁾ к.т.н, доцент; ²⁾ магістрант

I. Постановка проблеми

Питання доцільності та актуальності візуалізації програмного коду, написаного на прототип-орієнтованих мовах є актуальним. Якщо врахувати, що прототипні мови затребувані, розвиваються і мають майбутнє, то можна сказати, що програмного коду написаного на цих мовах є досить багато і його будуть писати і далі. І тут вже можна сказати, що питання доцільності візуалізації програмного коду у вигляді діаграм або схем, відноситься не тільки до прототип-орієнтованим системам, а до будь яких мов,

які використовують будь-яку парадигму є актуальним. Можу сказати, що розбиратися в програмному коді, що б зрозуміти загальну структуру програми, без візуального у вигляді схем представлення дуже складно, особливо якщо мова йде про прототипні мови, у яких немає класів.

Підвищення складності програм на скриптових мовах і підвищення вимог до їх якості обумовлює актуалізація розробки інструментальних засобів, для підвищення рівня технологічності розробки програмного забезпечення на прототипними мовами. Одним з таких засобів є візуалізація програмного забезпечення засобами UML.

II. Мета роботи

Метою роботи є досягнення таких цілей:

1. Дослідити особливості прототип-орієнтованих мов і способи їх візуалізації.
2. Розробка інструменту для візуалізації програм, написаних на мові JavaScript у вигляді UML.

III. Модель системи трансформації тексту в нотацію UML

Прототипним програмуванням називається стиль ООП, в якому відсутнє поняття класу, а спадкування здійснюється за допомогою клонування екземпляра об'єкта - прототипу.

В клас-орієнтованих мовах всі об'єкти розділені на два основних типи - Класи і екземпляри. Клас визначає структуру і функціональність (Поведінка), однакову для всіх екземплярів даного класу. Об'єкт – це носій даних, що має задане класом стан, який змінюється по принципом, заданому класом. Якщо подивитися на визначення класу для мови C ++ в книзі Бен-Арі «Мови програмування». То клас це – одиниця інкапсуляції і абстракції, який містить оголошення підпрограм (Підпрограма тут метод або функція) і типів даних. З класу створюються фактичні об'єкти, звані екземплярами.

Основна діаграма, яка може допомогти при візуалізації програмного коду - це Діаграма класів. Але в прототипних мовах класів немає. Це ключова відмінність прототип-орієнтованої мови від клас-орієнтованого. Тому для фахівця зручно представити проект, у звичній для нього парадигмі класів. Тим більше що визначення класів дане в мові UML дозволяє це зробити. В прототипних мовах роль класу, може виконувати або об'єкт, або функція породжує об'єкти, або яка-небудь інша конструкція. Далі можна сказати, що, так як ключовим у прототипній мові є об'єкт, а в свою чергу клас може бути присутнім на діаграмі об'єктів. І якщо подивитися на програму, написану прототипною мовою з боку взаємодії об'єктів, то, ключовий для прототипної мови може бути діаграма об'єктів, а не Діаграма класів, і при цьому деякі мовні конструкції для зручності сприйняття розробників, можна піднести як клас. Тому клас можна показувати на діаграмі об'єктів для зручності сприйняття даної реалізації.

Висновок

Досліджено особливості прототип-орієнтованих мов, показана актуальність розробки для них засобів візуалізації з метою підвищення технологічності програмування. Досліджено можливість відображення прототип-орієнтованих мов в діаграми UML на прикладі мови JavaScript. Розроблено процесор розбору конструкцій мови JavaScript засобами генератора компіляторів Coco / R в проміжне представлення у вигляді дерева функцій і змінних.

Список використаних джерел

1. Іан Грэхем «Объектно-ориентированные методы. Принципы и практика. 3-е издание». Москва 2005.
2. Classifying Prototype-based Programming Languages. C. Dony, J. Malenfant, D. Bardon. Universite de Montpellier 1999.