

**РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА УПРАВЛЕНИЯ
ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫМИ
СЕТЯМИ**

И. Ф. Астахова, А. В. Селеменев

Воронежский государственный университет

Поступила в редакцию 26.07.2015 г.

Аннотация. В статье рассматривается решение задачи проектирования, управления и мониторинга локальной вычислительной сети, состоящей из удаленных друг от друга телекоммуникационных узлов связи. Разработан программный комплекс, позволяющий осуществлять планирование IP сети, управление сетевыми устройствами и анализировать состояния оборудования, находящегося на удаленных телекоммуникационных узлах. Результаты данной работы активно применяются в деятельности Воронежского предприятия ОАО «Воронежский опытный завод программной продукции».

Данный комплекс не имеет аналогов, является уникальным и не использует сторонних коммерческих библиотек. Исходный код проекта может быть использован при разработке нового программного обеспечения, так как в нем имеется ряд прикладных библиотек общего назначения.

Ключевые слова: локальная сеть, IP адрес, информационно-телекоммуникационная сеть, проектирование, мониторинг, управление .

Annotation. In article, the solution of a problem of design, management and monitoring of a local area network is considered. This network is consist of remote communication nodes. The program is developed to carry out such function as designing IP network, managing of network devices and analyzing state of facility, which is located on the remote communication nodes.

This software product is unique and uses only open-source libraries. Source code of this program can be used for developing new software.

Keywords: local network, IP address, information and telecommunication network, design, monitoring, management.

1. ВВЕДЕНИЕ

В связи с бурным развитием сетевых устройств появляется необходимость в средствах, которые позволяют спланировать сеть, сетевые диапазоны локальных сетей, предусмотреть сетевое взаимодействие различных устройств и назначаемые устройствам IP адреса. Возникает необходимость в средствах, позволяющих анализировать состояние сетевых устройств и управлять ими через

доступные протоколы, такие как SSH, Telnet, HTTP и другие.

В существующих программах, например, Microsoft Visio, Network Wizard или Network Designer можно спроектировать сеть в WYSIWYG редакторе, где устройства представлены в виде рисунков, из которых непонятно, как соединены между собой физические порты устройств.

Возникает необходимость в программном продукте, с помощью которого можно было бы спланировать информационно-те-

лекоммуникационную сеть; создать шаблон телекоммуникационного узла с возможностью ввода IP адресов для них в момент его добавления в общую схему. Этот программный комплекс позволил бы анализировать состояние физических портов оборудования и его виртуальных интерфейсов; конфигурирование устройств через протокол, доступный для типа конфигурируемого оборудования, например, через протоколы HTTP, SSH или Telnet клиенты. Программный комплекс должен иметь понятный пользователю интерфейс отображения устройств, их сетевые адреса, состояние оборудования, а также связи между физическими их портами. Должны быть доступны возможности сохранения графических схем в различных форматах, таких как pdf, Microsoft Visio для дальнейшего вывода на печать и распространения между пользователями.

Задачами исследования являются:

- проектирование сети с возможностью создания шаблонов для часто повторяющихся наборов сетевого оборудования;
- управление сетевыми устройствами по протоколам HTTP, Telnet и SSH;
- мониторинг сетевых устройств, проверка состояний устройств через протокол ICMP и отображение изменения состояния оборудования в реальном времени.

2. РЕАЛИЗАЦИЯ ПРОГРАММНОГО КОМПЛЕКСА

Для реализации программного комплекса используется высокоуровневый язык программирования общего назначения Python [1, 2], который ориентирован на повышенную производительность разработчика и читаемость кода. Для разработки графического интерфейса пользователя используется фреймворк Qt, позволяющий запускать программное обеспечение в большинстве современных операционных систем путем простой компиляции программы для каждой ОС без изменения исходного кода. Программный комплекс работает в ОС семейства Windows и ОС, базирующихся на ядре Linux. Для удобства работы с объектами используется объ-

ектно-реляционное отображение (ORM), для этого предлагается свободная реализация этой технологии – SQLAlchemy программная библиотека на языке Python для работы с реляционными СУБД. В данной работе для хранения информации программный комплекс опирается на реляционную СУБД SQLite. Такой выбор СУБД позволяет с легкостью обмениваться файлом базы данных различным пользователям, так как СУБД SQLite является компактной встраиваемой реляционной базой данных и работает без сервера. SQLite хранит всю базу данных в единственном стандартном файле на том компьютере, на котором исполняется программный код.

Проект представляет собой приложение, написанное на языке Python со следующей структурой:

- каталог bin хранит скомпилированные исполняемые файлы программы, а также библиотеки, конфигурационный файл и файлы баз данных;
 - каталог src хранит исходные тексты программы;
 - подкаталог src/pics хранит файлы ресурсов приложения, сюда входят разнообразные изображения, используемые в программе, css и javascript файлы;
 - подкаталог src/networkdesigner хранит основные исходные файлы приложения и подкаталоги с файлами, которые в терминах языка Python называются пакетами. Пакеты должны содержать внутри каталога файл `__init__.py` для того чтобы их можно было импортировать в других модулях;
 - подкаталог src/networkdesigner/db/data.py хранит метаданные для создания таблиц в базе данных через объектно-реляционное отображение SQLAlchemy.
- Запуск проекта из исходных кодов осуществляется через файл `src/networkdesigner/main.py`.
- Запуск проекта из исполняемого файла осуществляется через файл `bin/NetworkDesigner.exe`.
- Запуск проекта из исполняемого файла с выводом отладочной информации осуществляется через файл `bin/NetworkDesignerDebug.exe`.

Если ранее этого приложения не было создано баз данных, то при первом запуске приложения в каталоге bin будет создана база данных SQLite с именем networkdesigner.ndb.

Каталог install хранит скрипты для сборки дистрибутива программы.

На рис. 1 представлена структура разработанного приложения. Пользователь с помощью персонального компьютера через интерфейс этой программы осуществляет выполнение различных задач, поставленных перед ним, например, планирование сети и назначаемых IP адресов, компоновка устройств в логически связанные блоки, управление сетевыми устройствами и др. Модули программного комплекса анализируют действия пользователя и осуществляют обработку полученных данных через интерфейс доступа к данным (API). В данном проекте это ORM система SQLAlchemy. ORM система, в свою очередь, осуществляет процесс взаимодействия с СУБД. Логические модули приложения проверяют корректность заполнения форм редактирования, уникальность IP адресов, обработку шаблонов IP адресов с переменными. Интерфейс пользователя отвечает за отображаемые пользователю данные, которые выводятся на экран на основе сведений, хранящихся в базе данных. Модель данных описывает структуру базы данных, логические связи между таблицами, ограничения целостности и индексы, использующиеся для быстрого поиска информации.

В связи с тем, что комплексы сетевого оборудования часто повторяются на различных коммуникационных узлах, возникает необходимость в автоматизации назначения IP адресов таким образом, чтобы каждый был уникален в своей сети, а для меняющихся адресных частей должны быть введены переменные, значения которых можно изменять во время добавления шаблона комплекса оборудования в коммуникационный узел.

На рис. 2 представлена форма ввода шаблона IP адреса.

На рис. 3 приведен скриншот графической схемы во время режима редактирования шаблона комплекса оборудования.



Рис. 1. Структура приложения планирования сети

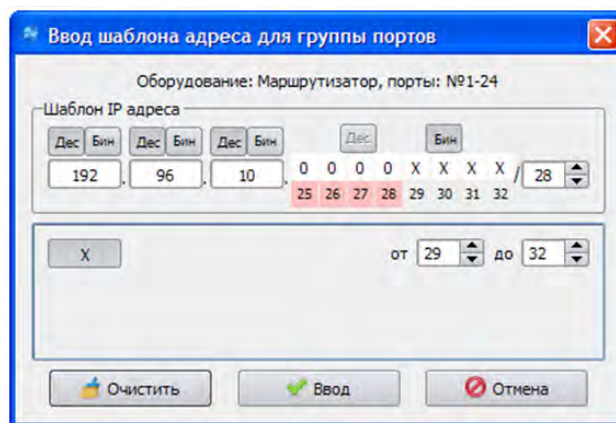


Рис. 2. Форма ввода шаблона IP адреса

После создания необходимых шаблонов комплексов оборудования они добавляются на схему телекоммуникационных узлов. Между различными телекоммуникационными узлами можно также прокладывать кабельные соединения [3, 4].

Для решения задачи мониторинга оборудования был разработан модуль, который опрашивает порты оборудования с заданными адресами через протокол ICMP, с заданной периодичностью посылая запросы состояния. Для того чтобы мониторинг осуществлялся эффективно, процесс, который опрашивает оборудование, был распараллелен при помощи модуля threading входящего в стандартную библиотеку языка Python, благодаря этому удалось добиться значительного улучшения в скорости работы модуля

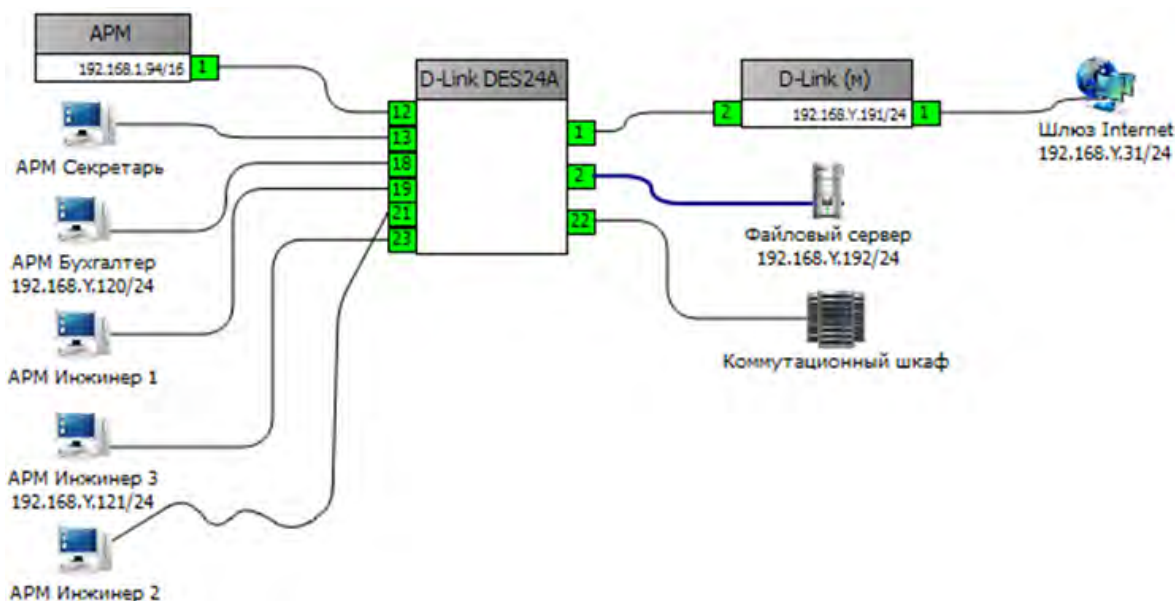


Рис. 3. Режим редактирования шаблона комплекса оборудования

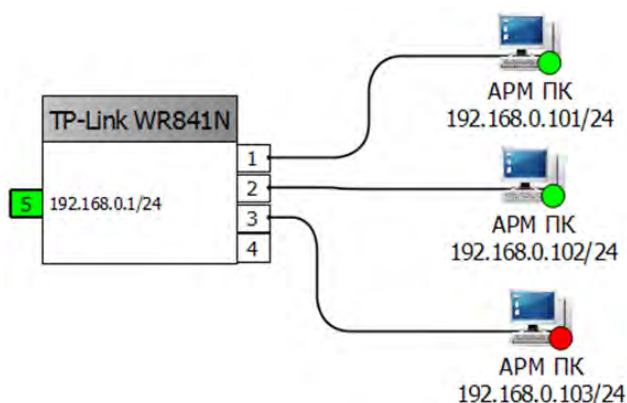


Рис. 4. Мониторинг сетевого оборудования TP-Link WR841N

мониторинга. Если до оптимизации последовательный опрос 250 устройств занимал 28 с, то теперь он занимает 3 с. Большое количество времени расходуется на опрос устройства, которое недоступно, так как необходимо ожидать максимальное время задержки (обычно 2000 мс), чтобы сделать вывод, что оборудование не отвечает на запросы состояния.

Задача управления сетевым оборудованием реализована с помощью библиотек с открытым исходным кодом для управления по наиболее распространенным протоколам. На данный момент программа поддерживает конфигурирование оборудования через веб-браузер по протоколам HTTP и HTTPS, через сессии SSH и Telnet, а также с помощью внешних программ, для которых предусмо-

трена возможность задания опций командной строки. Тип протокола, по которому доступно управление оборудованием, выбирается при создании/изменении типа оборудования.

На рис. 5 представлена модель базы данных приложения [5, 6].

Таким образом упрощается настройка оборудования за счет того, что пользователь не тратит времени на установку, запуск и настройку стороннего ПО, а работает непосредственно в одном программном комплексе. Для реализации встроенного в программу веб-браузера используется библиотека WebKit, предлагаемая такими популярными браузерами как Google Chrome, Apple Safari и другими. Для управления сетевыми устройствами по протоколам SSH и Telnet используются библиотеки, написанные на языке Python, для Telnet – встроенный в Python одноименный модуль, а для SSH – модуль Paramiko SSH, который можно найти в репозитории библиотек для языка Python <http://pypi.python.org/>.

В данной работе разработан программный комплекс, выполняющий поставленные задачи, а именно планирование, мониторинг и управление информационно-телекоммуникационными сетями. В ходе разработки были использованы библиотеки с открытым исходным кодом, что позволяет при необходи-

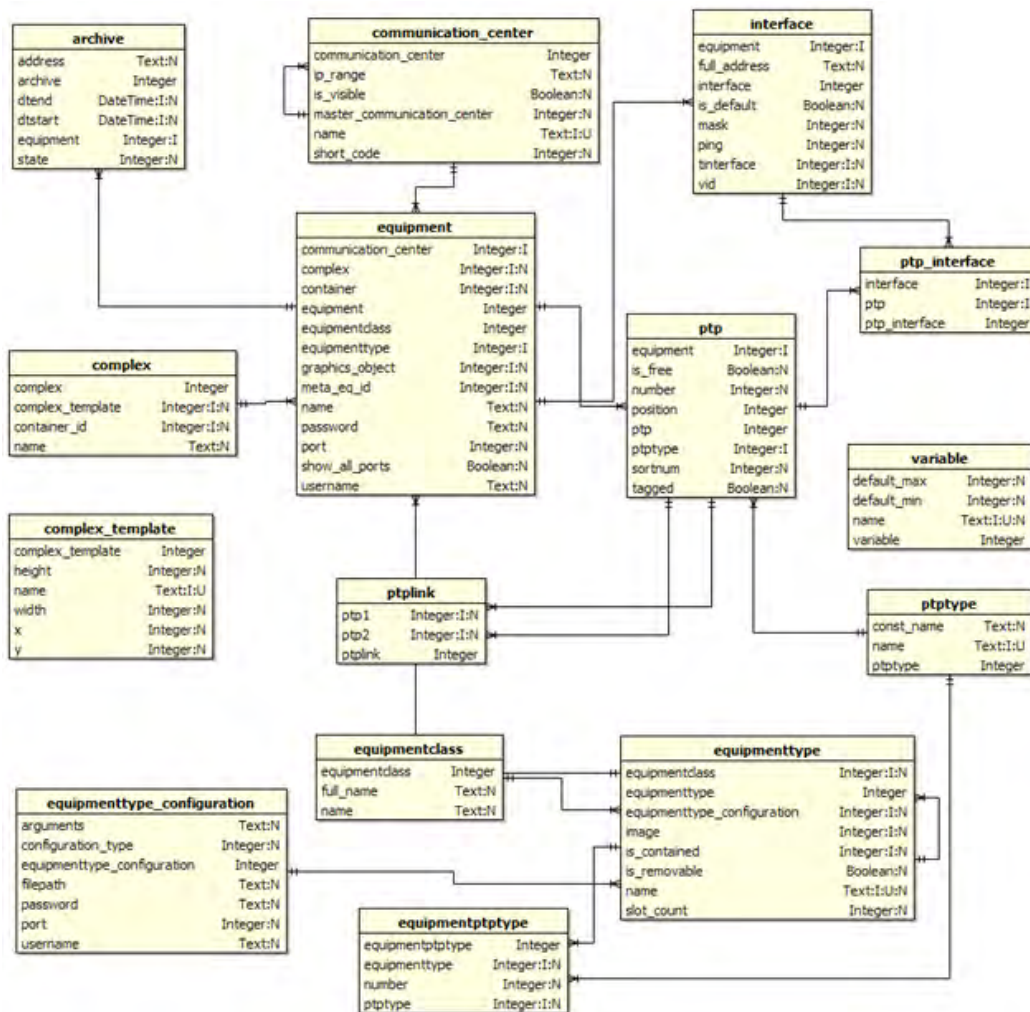


Рис. 5. Модель данных приложения в виде ER диаграммы

The screenshot shows a window titled "Управление типом оборудования" (Equipment Type Management). It contains several sections for configuring a new equipment type:

- Класс типа оборудования** (Equipment Type Class): Radio buttons for "Маршрутизатор" (selected), "Абонент", "Коммутатор", and "Сетевой мост".
- Имя типа оборудования** (Equipment Type Name): Text field containing "TP-Link WR841N".
- Количество портов типа LAN** (Number of LAN ports): Spin box.
- Количество портов типа WAN** (Number of WAN ports): Spin box.
- Конфигурирование оборудования** (Equipment Configuration): Radio buttons for "Нет", "Веб-браузер" (selected), "SSH", "Telnet", and "Внешняя программа".
- Имя пользователя** (Username): Text field containing "admin".
- Пароль** (Password): Text field containing "admin".
- Порт** (Port): Check box "Включен" (checked) and spin box "80".
- Buttons for "OK" and "Отмена" (Cancel).

Рис. 6. Форма создания нового типа оборудования

мости произвести процедуру сертификации программного кода, и доработать необходимые программные модули. Все модули были протестированы и имеют сопровождающие их автоматические тесты (модульные тесты).

3. СТРУКТУРА ПРОГРАММНЫХ МОДУЛЕЙ

Приложения, созданные при помощи фреймворка Qt, начинают свою работу с инициализации класса QApplication. Экземпляр этого класса (всегда существует только один экземпляр приложения в соответствии с паттерном проектирования Singleton) запускает основной поток приложения.

Взаимодействие интерпретатора языка Python с библиотеками фреймворка Qt осуществляется через модули привязки. На момент написания данной работы существуют

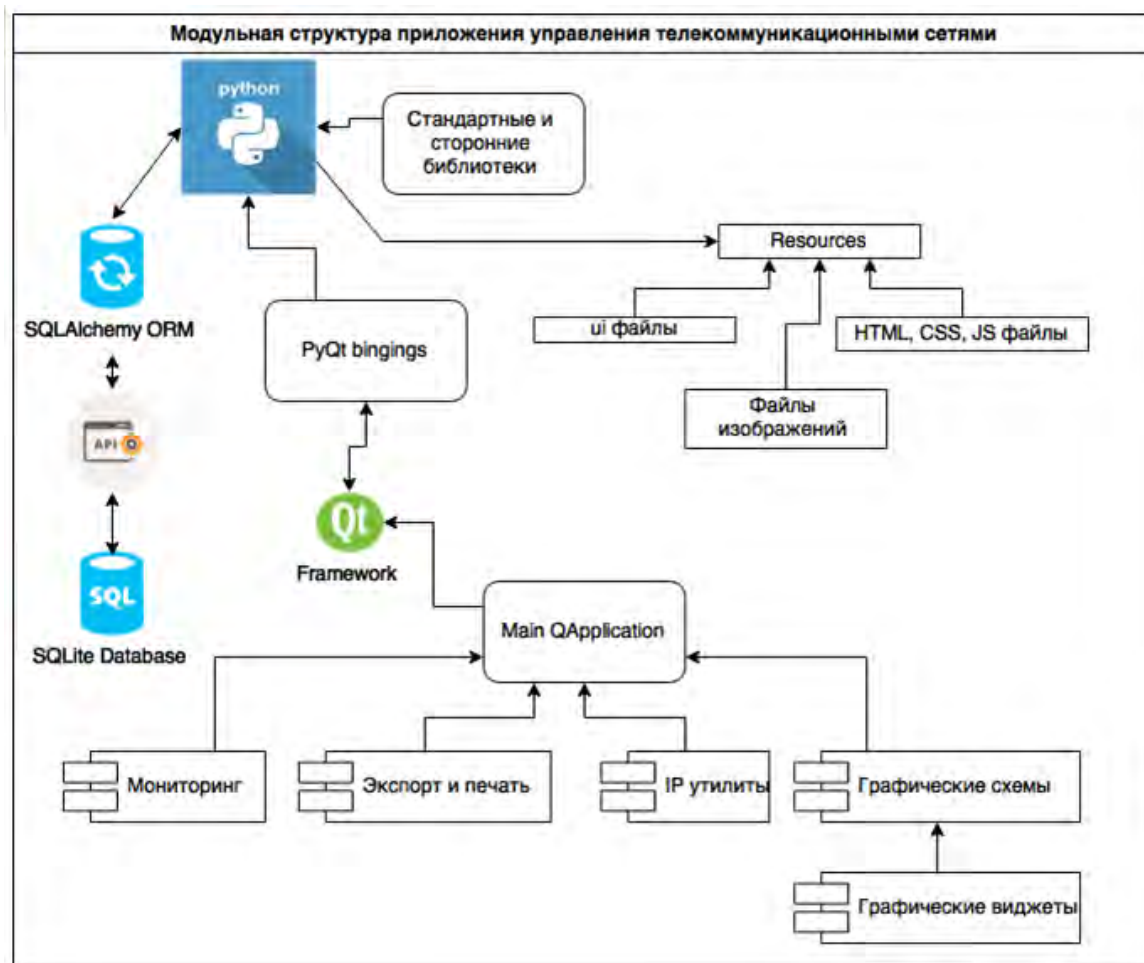


Рис. 7. Модульная структура приложения планирования сети

два наиболее популярных варианта такой привязки. Это PySide и PyQt реализации. В данной работе была выбрана реализация PyQt в связи с большей популярностью этого программного продукта и более активным сообществом разработчиков, регулярно выпускающих обновления его.

Взаимодействие приложения с базой данных осуществляется средствами фреймворка SQLAlchemy. Фреймворк осуществляет свою работу через API интерфейс библиотеки SQLite. При инициализации соединения он создает привязку объектов языка Python к записям таблиц базы данных. Управление транзакциями осуществляется программистом в коде обращения к объектам БД.

Приложение включает в себя несколько больших подсистем.

Подсистема мониторинга – это пакеты и модули, обеспечивающие выполнение задачи анализа состояния сетевого оборудования.

Подсистема экспорта и печати включает в себя виджеты предпросмотра печати, модули экспорта схем в форматы Microsoft Visio, SVG, PNG.

Подсистема утилит IP состоит из модулей, обеспечивающих различные преобразования над IP адресами, включая работу с переменными в адресах, конвертацию адресов, проверку вхождения в подсети, виджеты ввода и редактирования шаблонов IP адресов и обычных IP адресов.

Подсистема графических схем включает в себя виджеты графических сцен и графические виджеты, которые размещаются на сценах. Это наиболее объемная подсистема, которая является важной частью программного продукта.

Ресурсы приложения, а именно, файлы стилей, файлы графических форм хранятся в скомпилированном файле ресурсов resources.py и соответствующем ему qrc-файле, хра-

нящем пути ресурсов. Это обеспечивает быстрый доступ к ресурсам приложения. Ресурсы компилируются при помощи утилиты, входящей в дистрибутив фреймворка PyQt – rугсс и руиис.

На рис. 7 приведена модульная структура приложения.

4. ЗАКЛЮЧЕНИЕ

Целью работы было проектирование и разработка программного продукта, выполняющего задачи планирования, мониторинга и управления информационно-телекоммуникационными сетями. При проектировании использовались современные средства для создания UML диаграмм и проектирования структуры базы данных (ER диаграммы). Для программной реализации были использованы современные средства (активно развивающийся язык Python и графический фреймворк Qt) и методики разработки программного обеспечения (шаблоны проектирования, нормализация базы данных, модульное тестирование). Использованы методы оптимизации скорости выполнения программы, в частности, применение методик параллельного программирования и методы оптимизации SQL запросов. В результате был разработан программный комплекс, в полном объеме выполняющий поставленные задачи, а именно: планирование сетевых IP адресов с возможностью ввода переменных,

управление сетевыми устройствами по наиболее распространенным протоколам и анализ состояния сетевых устройств.

В ходе разработки были использованы библиотеки с открытым исходным кодом, что позволяет при необходимости произвести процедуру сертификации программного кода и дополнить необходимые программные модули.

СПИСОК ЛИТЕРАТУРЫ

1. Dive into Python / Mark Pilgrim.; United States, New York : Apress, 2010. – 495 с.
2. Лутс М. Изучаем Python. 4-е издание / М. Лутс.; пер. с англ. – СПб. : Символ-Плюс, 2011. – 1280 с.
3. Фаулер. М. Архитектура корпоративных программных приложений / М. Фаулер.; пер. с англ. – М. : Издательский дом Вильямс, 2006. – 544 с.
4. Таненбаум. Э. Компьютерные сети / Э. Таненбаум.; пер. с англ. – СПб. : Издательство Питер, 2007. – 992 с.
5. Астахова И. Ф. SQL в примерах и задачах / И. Ф. Астахова, А. П. Толстобров, В. М. Мельников. – М: ФИЗМАТЛИТ, 2007. – 174 с.
6. Астахова И. Ф. Модель для обучающей контролирующей системы / И. Ф. Астахова, И. В. Сухотерина // Вестник Воронежского ун-та. Серия: Системный анализ и информационные технологии. – 2014. – №1. – С. 95–101.

Астахова И. Ф. – д. т. н., профессор, кафедра математического и прикладного анализа, факультет прикладной математики и механики, Воронежский государственный университет. E-mail: astahova@list.ru

Селемнев А. В. – аспирант, факультет прикладной математики и механики, Воронежский государственный университет. E-mail: andreyjkee@gmail.com

Astahova I. F. – Doctor of Engineering Sciences, Professor, Applied Mathematics and Mechanics Faculty, Department of Mathematical and Applied Analyse, Voronezh State University. E-mail: astahova@list.ru

Selemenev A. V. – graduate student of AMM faculty of Voronezh State University. E-mail: andreyjkee@gmail.com