

# Распределенная система группового тестирования и оценки знаний

Н. А. Молева, email: namoleva@gmail.com<sup>1</sup>

А. А. Молев, email: alexei\_molev@mail.ru<sup>2</sup>

<sup>1</sup> Воронежский государственный университет

<sup>2</sup> ВУНЦ ВВС «ВВА» (г. Воронеж)

***Аннотация.** В данной работе рассматривается автоматизированная система для тестирования оценки знаний обучающихся. Описан подход к организации задачи проверки знаний с применением технологии «клиент-сервер», а также приведены примеры работы программы с созданием уникальных тестов для произвольного количества пользователей. Разработанная программа может быть использована для автоматизации работы в больших группах с целью исключения ошибок при вводе и выводе данных в задачах, ускорения процесса обработки результатов тестирования и повышения удобства при анализе преподавателем.*

***Ключевые слова:** дистанционное обучение, технология клиент-сервер, система тестирования, Python, XML-формат, автоматизированные системы обучения.*

## Введение

В настоящее время использование современных технологий в обучении стало повсеместным явлением. Это изменило подход к образованию в целом и обучению в частности. В нашей стране дистанционные информационные образовательные технологии получили интенсивное развитие и стали завоевывать свое место в образовательном процессе наряду с традиционными формами обучения.

Под дистанционными образовательными программами понимаются технологии, реализуемые в основном с применением информационных и телекоммуникационных технологий при опосредованном (на расстоянии) или не полностью опосредованном взаимодействии обучающегося и преподавателя.

В настоящее время в Воронежском государственном университете все электронное образование осуществляется посредством электронной платформы Moodle [1]. Авторы позиционируют ее как среду с широчайшими возможностями для реализации обучающих курсов и

обладающей простотой в использовании.

Однако эта система имеет ряд недостатков, связанных с формированием и загрузкой данных для тестирования. Для каждого задания необходимо заполнять вручную большое количество полей, причем для разных вариантов каждый раз нужно самостоятельно загружать один и тот же тип задач, чтобы исключить возможность списывания. Также каждого задания отдельно загружается ответ с настройками критериев оценивания [2].

Следующим недостатком является недостаточная стабильность платформы, то есть система может аварийно завершить свою работу и не дать возможность закончить тестирование. Кроме того в ряде случаев, правильный ответ засчитывался платформой как ошибочный, несмотря на корректность соблюдения регистра и языка ввода ответа.

Указанные факторы определяют актуальность разработки распределенной системы тестирования знаний обучающихся, с заданиями, формируемыми автоматически в соответствии с предметом и темой обучения, что является целью настоящей работы.

## **1. Общее описание системы тестирования**

Сущность предлагаемого подхода заключается в выполнении обучающимися на своих компьютерах тестовых заданий, автоматической проверки правильности их выполнения, сборе итоговой статистики и отправке ее на сервер для анализа преподавателем.

В обобщенном виде принцип работы системы тестирования приведен на рисунке 1 и заключается в следующем. Первоначально осуществляется идентификация пользователя, необходимая для последующего анализа результатов преподавателем. Это может быть ФИО студента или номер его студенческого билета в зависимости от того, какие данные внесены в систему для связки пользователя с внутренними элементами программы.

Далее выбирается предмет (курс), направление курса (раздел) и тема. Здесь дополнительно может быть указано направление, на котором учится студент для разгрузки задач соответствующего уровня сложности.

После этого обучающийся выполняет задания (решает тесты). Для каждого задания автоматически проверяется результат его выполнения и отображается на экране, а после выполнения всех заданий отображается общая оценка результатов, которая отправляется на сервер. Результаты решения заданий, загруженные на сервер, используются преподавателем для анализа степени изучения дисциплины и достигнутого уровня знаний как отдельными студентами, так и группой.

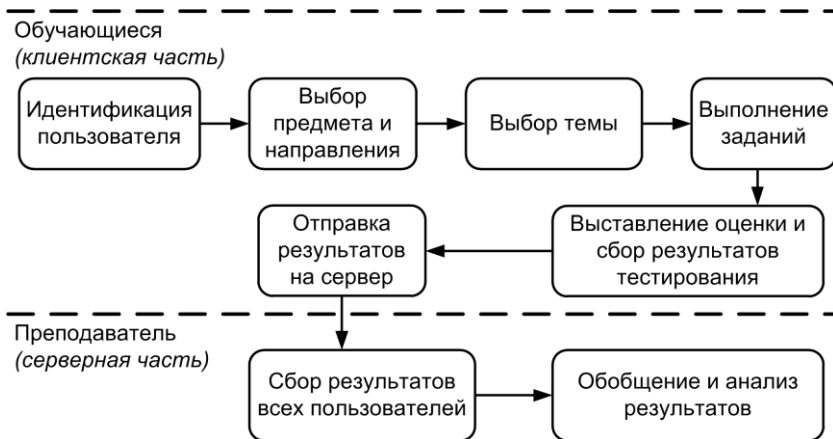
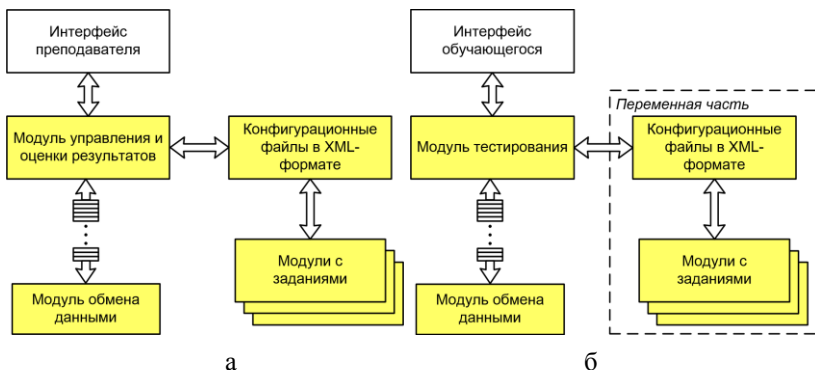


Рис. 1. Обобщенная схема функционирования

Система тестирования включает в себя две основных части – серверную и клиентские (рисунок 2).

Серверная часть содержит модуль управления и оценки результатов, модуль обмена данными и модули с заданиями.

Клиентские части включают в себя модуль тестирования, модуль обмена данными и, загружаемые в зависимости от предмета, конфигурационные файлы и модули с заданиями, загружаемые в соответствии с выбранным предметом, направлением и темой.



а – серверная часть, б – клиентская часть

Рис. 2. Структура программы

## 2. Реализация программы

Система тестирования реализована на языке Python, с применением которого возможно гибко модифицировать элементы программы без перекомпиляции ее в полном объеме, а также обладает кроссплатформенностью, что позволяет не ограничиваться рамками операционной системы Windows, и иметь возможность запуска разработанной программы в любой современной операционной системе, поддерживающей интерпретатор языка [3].

Основой для формирования модулей тестирования является использование унифицированного описания в формате XML (eXtensible Markup Language). С помощью XML-описания автоматически формируется интерфейс ввода/отображения принимаемой и передаваемой информации, а также запуск модулей тестирования. Использование XML-формата обуславливается его удобством для описания информации, имеющей заранее определенную структуру [4].

Для задания структуры заданий, отображаемых модулями тестирования, используется XML-описание, структура которого в виде диаграммы XSD приведена на рисунке 3. На верхнем уровне располагается элемент TRAINER, включающий в себя совокупность элементов SUBJECT, каждый из которых соответствует отдельному предмету обучения. Для каждого предмета задается атрибут NAME – наименование предмета. Предметы включают перечень тем TOPIC, для каждой из которых задаются атрибуты двух видов – name и module, соответствующие наименованию и файлу .ru с заданиями для этой темы. Внутри каждой темы содержатся задания TASK, каждое из которых характеризуется атрибутами name и function – наименование и функция в файле с заданиями, который указан в теме.

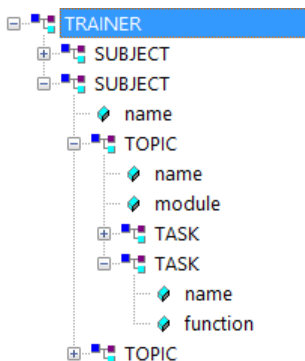


Рис. 3. XSD - диаграмма описания конфигурационных файлов

В качестве примера приведен фрагмент XML-описания файла конфигурации для предметов «Математика» и «Информатика» и имеющего вид:

```
<?xml version="1.0" encoding="utf-16"?><!DOCTYPE TRAINER []>
<TRAINER>
  <SUBJECT name="Математика">
  </SUBJECT>
  <SUBJECT name="Информатика">
    <TOPIC name="Алфавитный подход " module="informatics">
      <TASK name="Автомобильные номера"
function="autonomera"/>
      <TASK name="Номера спортсменов"
function="nomerasport"/>
    </TOPIC>
    <TOPIC name="Системы счисления" module="system_calc">
      <TASK name="Перевод из одной системы счисления"
function="trans"/>
      <TASK name="Определение основания системы
счисления" function="def_base"/>
    </TOPIC>
  </SUBJECT>
</TRAINER>
```

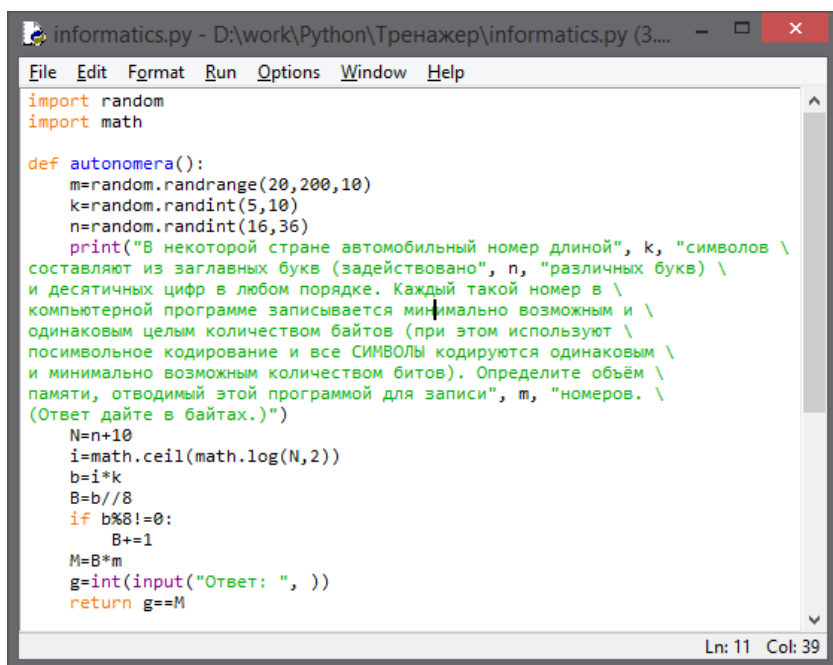
На приведенном примере видно, что здесь пользователь может выбрать предмет, по которому он собирается проходить тестирование, а дальше на примере дисциплины «Информатика» переходит к выбору основных тем, а уже внутри них находятся соответствующие тесты.

Особенностью реализации программы является формирование заданий со случайными параметрами и автоматическая проверка правильности их выполнения в соответствии с программным кодом, что позволяет для каждого обучающегося создавать неповторяющиеся варианты заданий для любого количества студентов в группе, исключив ручные операции ввода набора заданий, ответов на них и их проверки.

В этой части как раз и решаются основные проблемы, существующие в платформе Moodle [2]. Для исключения списывания студентами необходимо заполнить форму такое количество раз, чтобы число вопросов было для каждого индивидуальным. На заполнение одного вопроса требуется примерно порядка 10 минут, при условии, что все задачи решены, и все ответы уже есть в готовом виде.

Преимуществом распределенной системы является однократное написание кода для задачи, где числа в номера берутся случайным образом с помощью функции `gandint` [3]. Далее система сама решает и сверяет данные внесенные пользователем.

Пример задания «Автомобильные номера» [5] для темы «Алфавитный подход» приведен на рисунке 4, а результат выполнения программы – на рисунке 5.

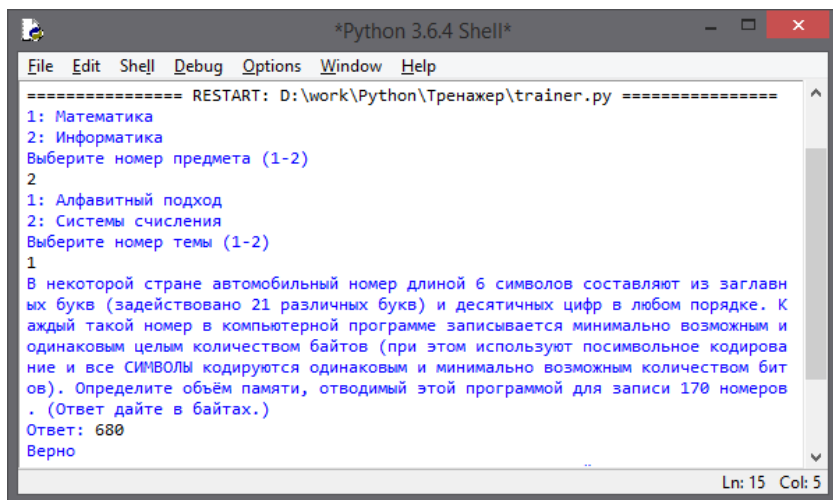


```
informatics.py - D:\work\Python\Тренажер\informatics.py (3...
File Edit Format Run Options Window Help
import random
import math

def autonomera():
    m=random.randrange(20,200,10)
    k=random.randint(5,10)
    n=random.randint(16,36)
    print("В некоторой стране автомобильный номер длиной", k, "символов \
составляют из заглавных букв (задействовано", n, "различных букв) \
и десятичных цифр в любом порядке. Каждый такой номер в \
компьютерной программе записывается минимально возможным и \
одинаковым целым количеством байтов (при этом используют \
посимвольное кодирование и все СИМВОЛЫ кодируются одинаковым \
и минимально возможным количеством битов). Определите объём \
памяти, отводимый этой программой для записи", m, "номеров. \
(Ответ дайте в байтах.)")
    N=n+10
    i=math.ceil(math.log(N,2))
    b=i*k
    B=b//8
    if b%8!=0:
        B+=1
    M=B*m
    g=int(input("Ответ: ", ))
    return g==M

Ln: 11 Col: 39
```

Рис. 4. Пример задания «Автомобильные номера»



```
*Python 3.6.4 Shell*
File Edit Shell Debug Options Window Help
===== RESTART: D:\work\Python\Тренажер\trainer.py =====
1: Математика
2: Информатика
Выберите номер предмета (1-2)
2
1: Алфавитный подход
2: Системы счисления
Выберите номер темы (1-2)
1
В некоторой стране автомобильный номер длиной 6 символов составляют из заглавн
ых букв (задействовано 21 различных букв) и десятичных цифр в любом порядке. К
аждый такой номер в компьютерной программе записывается минимально возможным и
одинаковым целым количеством байтов (при этом используют посимвольное кодирова
ние и все СИМВОЛЫ кодируются одинаковым и минимально возможным количеством бит
ов). Определите объём памяти, отводимый этой программой для записи 170 номеров
. (Ответ дайте в байтах.)
Ответ: 680
Верно

Ln: 15 Col: 5
```

Рис. 5. Пример работы программы в ходе тестирования

## **Заключение**

В статье рассматриваются вопросы разработки распределенной системы тестирования знаний студентов, гибко конфигурируемой в зависимости от предмета и раздела учебной программы. Показано, что для обеспечения объективной оценки знаний обучающихся и исключения ручных операций формирования неповторяющихся заданий необходима разработка модулей тестирования, с заданиями, формируемыми автоматически в зависимости от предмета и темы. Приводятся структура программы, требования к формату унифицированного описания конфигурационных файлов. Унификация реализуется путем использования XML-описания, имеющего единый формат для любого структурного элемента независимо от алгоритма его функционирования. Программная реализация предлагаемой программы выполнена на языке Python.

Разработанная программа может быть использована для контроля уровня знаний студентов в ходе проведения занятий по дисциплинам «Математика» и «Информатика», а также может быть дополнена другими предметами, например «Физика» или «Химия», в которых знания могут быть проверены с помощью решения задач разного типа.

## **Список литературы**

1. Moodle [Электронный ресурс]. – Режим доступа: <https://moodle.org>
2. Как добавить и настроить элемент «Тест» [Электронный ресурс]. – Режим доступа: [https://moodle.tsu.ru/pages/pdf/el\\_test.pdf](https://moodle.tsu.ru/pages/pdf/el_test.pdf)
3. Прохоренок, Н. А. Python 3. Самое необходимое / Н. А. Прохоренок, В. А. Дронов. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2020. – 608 с.
4. XML. Базовый курс / Хантер Дэвид [и др.] ; перевод Красиков И. В. – 4-е издание. – Киев : Вильямс, 2018. – 1344 с.
5. СДАМ ГИА: Решу ЕГЭ. Информатика [Электронный ресурс]. – Режим доступа: <https://inf-ege.sdangia.ru>