



2-сүрөт. Программанын интерфейси

Сөздөрдү жаратуунун ар кандай жолдорун техникалык жактан мүнөздөө өтө оор процесс, анткени ал көптөгөн эрежелерден жана чектөөлөрдөн турат. Бир канча унгудан турган сөздөрдү бир тилден экинчи тилге которуу кийирилген сөзгө бир дагы окшобогон башка морфологиялык сөздүн жаралуусуна алып келет. Мисалы: ВУЗ- ЖОЖ, малоисследованный- аз изилденген ж.б.

Төмөндөгүдөй белгилөөлөрдү кийирип алабыз: w-сөз, p-префикстер, r-сөздүн унгусу, a-аффикстердин көптүгү.

$$\langle w \rangle ::= \langle r \rangle | \langle p \rangle \langle r \rangle | \langle r \rangle \langle a \rangle | \langle p \rangle \langle r \rangle \langle a \rangle$$

Мисалы: игра-оюн, игрушка – оюнчук, играть-ойноо, выиграть – утуу, проиграть- утулуу.

Бул мисалдарда сөздөрдүн бир эле унгуга морфологиялык бирдиктердин уланышынын негизинде маанилик жактан өзгөрүшү көрсөтүлдү.

Бир табигый тилден экинчи тилге машиналык которууда көп маанилүүлүктү чагылдыруу методун пайдалануу которуунун эффективдүүлүгүн жогорулатып, тексттин маанисине жараша көп маанилүүлүктөн бир мааниге өтүүгө жардам берет.

### Корутунду

Негизинен машиналык которуу багытында иштөөдө төмөнкүдөй амалдар аткарылды:

- көп колдонулуучу сөздөрдүн маалыматтар базасы түзүлдү;
- сөздөрдүн морфологиялык өзгөчөлүктөрү базага кийирилди;
- морфологиялык анализатордун программасы иштелип чыкты;
- жөнөкөй тексттерди сүйлөмдөргө ажыратуу жана которуу амалдары аткарылды.

Мындан сырткары көп маанилүүлүктү чагылдыруу методун программада пайдалануу маселеси каралды.

### Колдонулган адабияттар

1. Ж.Ч. Джанузакова «Лингвокультурологические аспекты машинного перевода» [Электрондук булак].
2. К.Х. Ким, А.П. Савинов «Синтаксический анализатор для вопросно-ответной системы»
3. А. Леонтьева, И. Кагиров «Автоматический синтаксический анализ русских текстов». Труды 10-й Всероссийской библиотеки: перспективные методы и технологии, электронные коллекции»-RCDL'2008, Дубна, Россия, 2008.

УДК 004.415.2

## ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ПРОВЕРКИ ОЛИМПИАДНЫХ ЗАДАНИЙ ПО ПРОГРАММИРОВАНИЮ

*Макиева Замира Джумакматовна, доцент КГТУ им. И.Раззакова, Кыргызстан, 720044, г.Бишкек, пр. Ч.Айтматова 66, e-mail: z.makieva@gmail.com*

Целью данной работы является проектирование автоматизированной системы проведения олимпиад по программированию, главной задачей которой является автоматическая проверка программ участников. Автором сделан краткий обзор существующих систем, показана необходимость разработки, созданы функциональная модель, модель потоков данных, диаграмма деятельности и алгоритм проверки решений.

**Ключевые слова:** программирование, автоматическое тестирование, олимпиада, моделирование системы, диаграмма.

## **DEVELOPMENT OF THE AUTOMATED VERIFICATION SYSTEM FOR PROGRAMMING OLYMPIAD TASKS**

*Makieva Zamira Jumakmatovna, the docent of KSTU named after I. Razzakov, Kyrgyzstan, 720044, Bishkek, C.Aitmatov Avenue 66, e-mail: z.makieva@gmail.com*

The purpose of this work is to design of the automated system of holding the Olympiad on programming, the main goal of which is automatic verification of programs of participants. The author has made the short review of the existing systems, need of development was shown, the functional model, model of data flows, activity diagram and program verification algorithm were created.

**Keywords:** programming, automatic testing, Olympiad, system modeling, diagram.

По программированию, как и по другим дисциплинам, проводятся соревнования среди школьников и студентов. Но в отличие, например, от соревнований по математике, здесь участники должны не просто решить задачу и представить ответ, а написать компьютерную программу, решающую целый класс однотипных задач с разными исходными данными. Соответственно, для проверки решений задач по программированию недостаточно сравнить ответы, а для каждой задачи необходимо приготовить множество тестов. Проверка вручную связана с определёнными проблемами, в первую очередь с затратами сил и времени и низким качеством ручной проверки в целом. Также приходится придумывать задачи, в которых выходных данных как можно меньше, чтобы упростить проверку. При этом отсеивается часть сложных и интересных задач. При ручной проверке оптимальность решения какой-либо задачи оценивается "на глазок", приблизительно и чаще этот критерий не учитывается за исключением явных "зависаний" проверяемого алгоритма. Наконец, после подведения итогов требуется вручную свести их в турнирную таблицу и необходимым образом её оформить в каком-либо текстовом редакторе.

В мире существует много аналогичных систем, мы рассмотрим две из них. Timus Online Judge [3] – это крупнейший в России архив задач по программированию с автоматической проверяющей системой. Основной источник задач для архива — соревнования Уральского федерального университета, Чемпионаты Урала, Уральские четвертьфиналы ACM ICPC, Петрозаводские сборы по программированию. Timus Online Judge позволяет принять участие в онлайн-версиях большинства соревнований, которые регулярно проходят в Уральском федеральном университете.[3]

Система KRSU Online Judge [2] используется для проведения кыргызстанского четвертьфинала студенческого командного чемпионата мира по программированию по правилам ACM ICPC (Association for Computing Machinery International Collegiate Programming Contest) [3, 4, 5] и для тренировок.

Все эти системы доступны для тренировок, но не для проведения соревнований. Также правила начисления баллов за решение задач в студенческих олимпиадах отличаются от олимпиад для школьников [5]. С 2014 года кафедра Программное обеспечение компьютерных систем КГТУ им. И.Раззакова проводит городскую и республиканскую

олимпиады школьников по информатике, что привело к необходимости разработки собственной системы. В качестве отличий разрабатываемой системы можно отметить то, что система будет проверять решения и по правилам проведения школьных олимпиад (International Olympiad in Informatics) и по правилам проведения студенческих олимпиад (Association for Computing Machinery).

Для решения указанных проблем требуется разработать автоматизированную тестирующую систему, которая способна выполнять проверку быстро и точно, независимо от количества входных и выходных данных.

Был произведен анализ предварительных требований к системе, и в результате была разработана спецификация требований, которая представлена ниже.

### **Функциональные требования к системе**

- 1) Добавление, удаление и редактирование задач и тестов к ним в архив, причем условия задач должны быть на кыргызском и русском языках.
- 2) Поддержка задач с множественными правильными ответами.
- 3) Регистрация участников самостоятельно и/или администратором.
- 4) Организация соревнований: составление списков задач и участников, указание времён начала и окончания.
- 5) Поддержка режимов тестирования по правилам ACM ICPC и IOI.
- 6) Приём исходного кода для проверки как в рамках соревнований, так и вне их.
- 7) Тестирование, включающее компиляцию исходного кода и автоматическую проверку работы получившейся программы на наборе тестов из архива, а также проверку времени работы и потребляемой памяти в процессе работы тестируемого алгоритма.
- 8) Вывод результатов тестирования, включая статус, количество пройденных тестов и сообщения об ошибках, возникших в процессе тестирования.
- 9) Составление таблицы с результатами соревнования для непосредственного просмотра и в формате PDF для скачивания и распечатки.
- 10) Поддержка языков программирования C++, Pascal, QBasic, возможность добавления других языков программирования.
- 11) Функционал резервного копирования системы в облако.

### **Нефункциональные требования**

- 1) Система должна иметь возможность работать как в локальной, так и в глобальной сети (интернет).
- 2) Система должна разграничивать доступ к данным.
- 3) В случае сбоя тестирующей системы, перезапуск должен осуществляться не дольше, чем за 30 секунд.
- 4) Система должна быть достаточно производительной, чтобы участникам соревнований не приходилось подолгу ждать результатов. Время получения результата решения не должно превышать 2 минут в обычном режиме и 10 минут в случае сбоя системы.

При построении **концептуальной модели** можно выделить три основных типа пользователей данной системы:

- 1) Пользователь – зарегистрированный пользователь, который не принимает участия в соревнованиях и решает задачи из архива.
- 2) Участник соревнований – пользователь, который участвует в соревновании и имеет доступ только к задачам этого соревнования.
- 3) Администратор – пользователь, который может организовывать соревнования, добавлять в базу данных задачи, тесты и других пользователей. Администратор имеет полный доступ ко всей системе.

На рис. 1 приведена Use-case диаграмма, соответствующая этой модели.

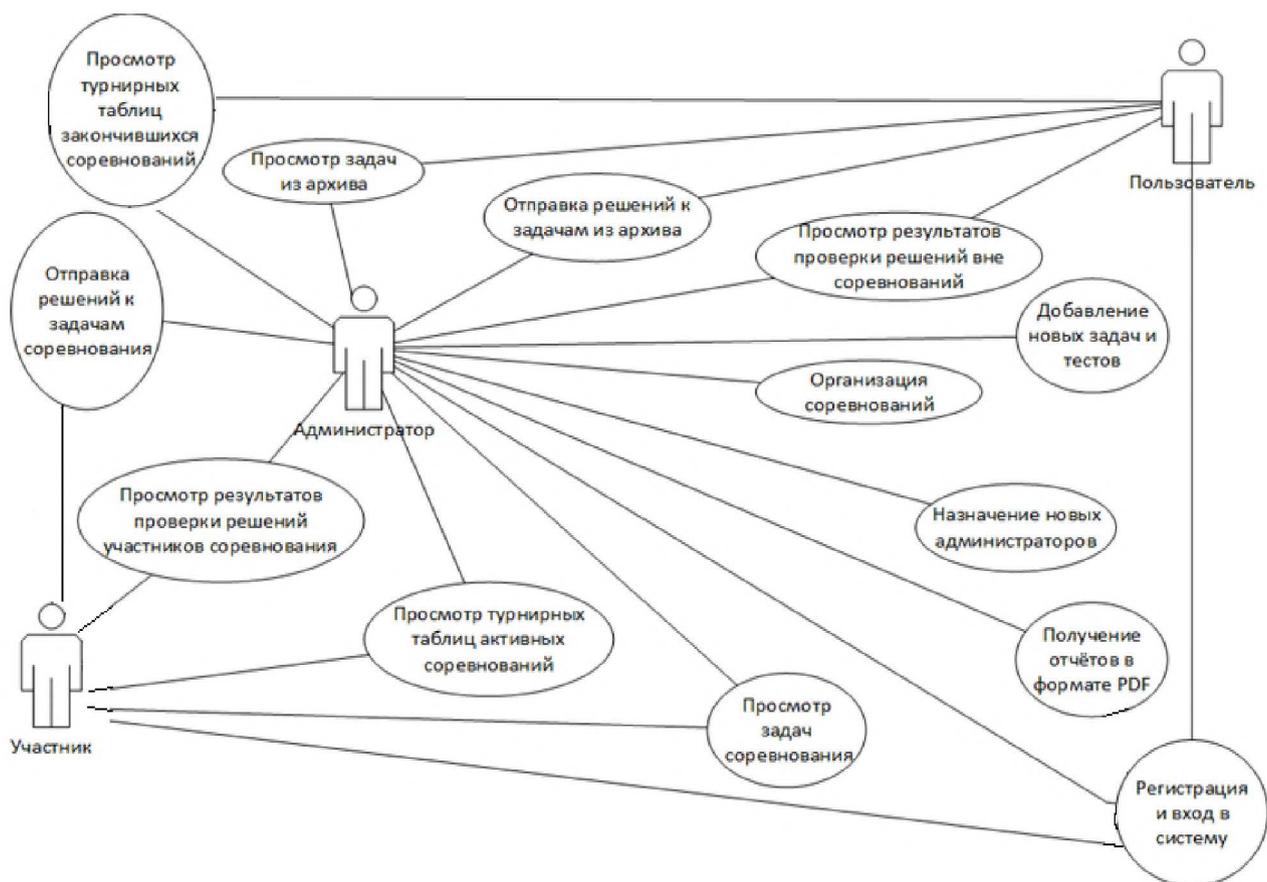


Рис. 1 Диаграмма вариантов использования

На диаграмме представлены следующие варианты использования:

- 1) Регистрация и вход в систему – регистрация доступна всем, вход в систему с помощью логина и пароля – зарегистрированным пользователям.
- 2) Просмотр задач из архива – доступен зарегистрированному пользователю, не участвующему в соревнованиях либо администратору. Задачи из соревнований в архиве не видны.
- 3) Отправка решений к задачам из архива – доступна простому пользователю и администратору. После отправки решение проверяется системой, и система выводит результат.
- 4) Просмотр результатов проверки вне соревнований – пользователи, тренирующиеся на задачах из архива, могут просматривать статусы своих и чужих решений, но не имеют доступа к решениям в рамках соревнований.
- 5) Просмотр турнирных таблиц закончившихся соревнований – доступен всем, кроме тех, кто участвует в активном соревновании.
- 6) Просмотр задач соревнования – доступен участникам этого соревнования и администратору. После окончания соревнования задачи становятся доступны в архиве.
- 7) Отправка решений к задачам соревнования – доступна участникам соревнований и администратору.
- 8) Просмотр турнирных таблиц активных соревнований – участники соревнований могут просматривать только турнирную таблицу своего соревнования. Администратор может просматривать любые турнирные таблицы.
- 9) Просмотр результатов проверки решений в рамках соревнований – участники соревнований могут просматривать только статусы друг друга.
- 10) Добавление задач и тестов – эта функция доступна только администратору.
- 11) Назначение новых администраторов – администратор может назначить себе помощников из числа пользователей.

12) Организация соревнований – включает в себя составление списков задач и участников и назначение времени начала и окончания. Доступ только для администратора.

13) Получение отчётов в PDF – отчёты для распечатки и вывешивания, доступно администратору.

### Модель потоков данных

Подробная модель представлена на рис. 2 в виде диаграммы потоков данных (Data Flow Diagram).



Рис. 2 Диаграмма потоков данных (DFD)

Внешними сущностями являются участники – пользователи системы и администрация – авторы задач, организаторы соревнований.

Для получения доступа к системе, участник должен зарегистрироваться и авторизоваться. Авторизованные участники могут читать условия задач и отправлять решения на проверку, указывая при этом язык программирования и задачу, которую они решали. По итогам проверки система выводит результаты проверки отдельных решений и общие итоги соревнований.

Администрация может добавлять новые тесты и задачи, организовывать соревнования и получать их итоги в формате PDF.

На рис. 3 представлена декомпозиция DFD-диаграммы тестирующей системы для демонстрации потоков данных внутри неё.

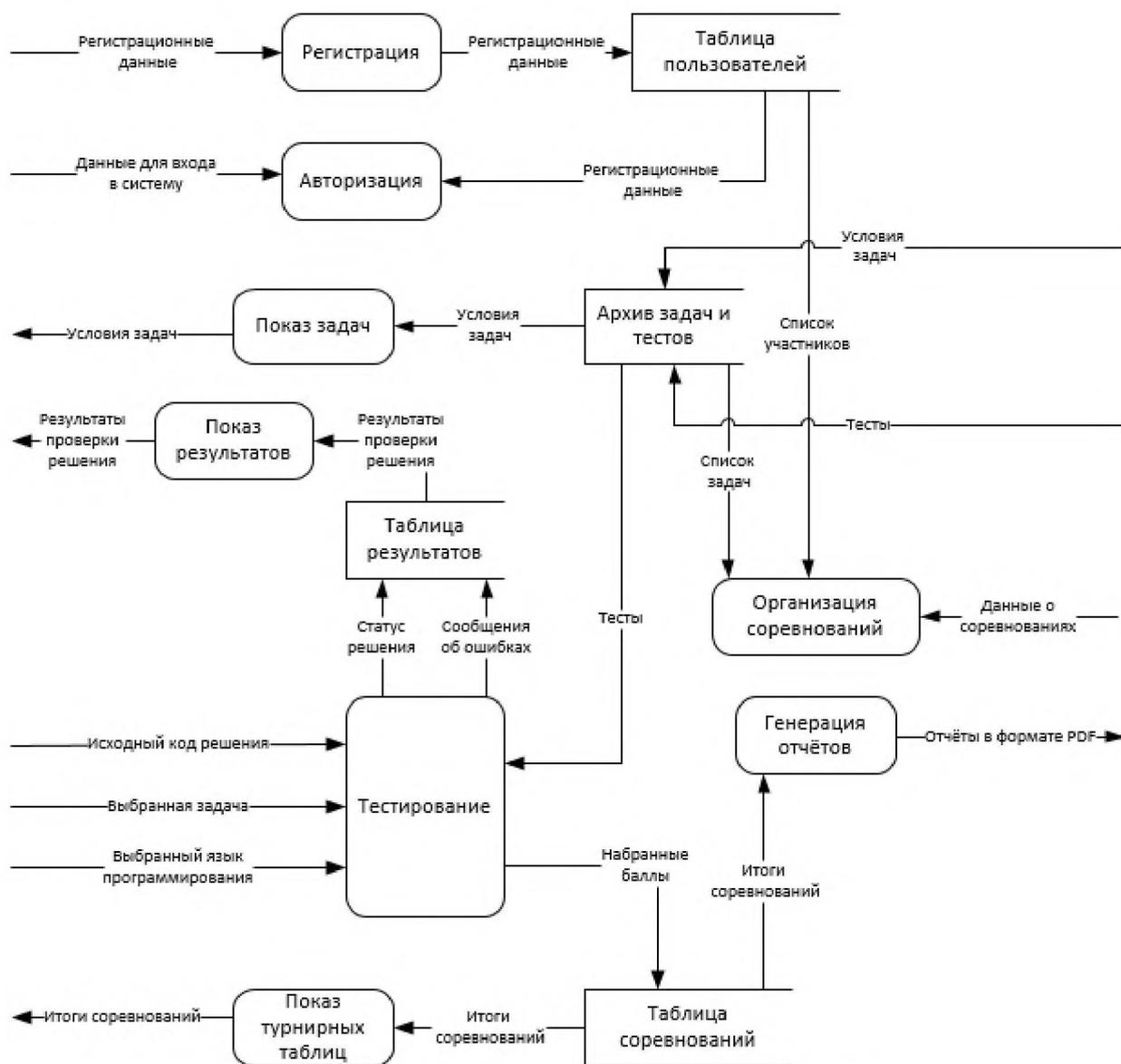


Рис. 3 Декомпозиция диаграммы потоков данных (DFD)

Здесь видно, что результаты проверки решений представляют собой статусы и сообщения об ошибках, возникших в процессе тестирования. Помимо выставления статусов и выдачи сообщений при тестировании ведётся подсчёт баллов, набранных за решение задачи, которые в дальнейшем отображаются в итогах соревнований.

#### Алгоритм тестирования решений к задачам

При проверке решений используется следующий алгоритм:

- 1) Скомпилировать исходный код. Если компиляция не удалась, вывести сообщение об ошибке компиляции и прервать тестирование.
- 2) Подготовить входные данные теста и запустить полученную программу.
- 3) Если программа работает слишком долго, прервать тестирование и выполнение программы.
- 4) Если программа потребляет слишком много памяти, прервать тестирование и выполнение программы.
- 5) Если программа завершила работу с ошибкой, прервать тестирование.
- 6) Если программа завершила работу успешно, то сравнить результаты работы программы и выходные данные теста. Если результат неправильный, прервать тестирование.
- 7) Продолжать тестирование с пункта 2, пока не кончатся тесты.

Алгоритм представлен в виде диаграммы деятельности на рис. 4.

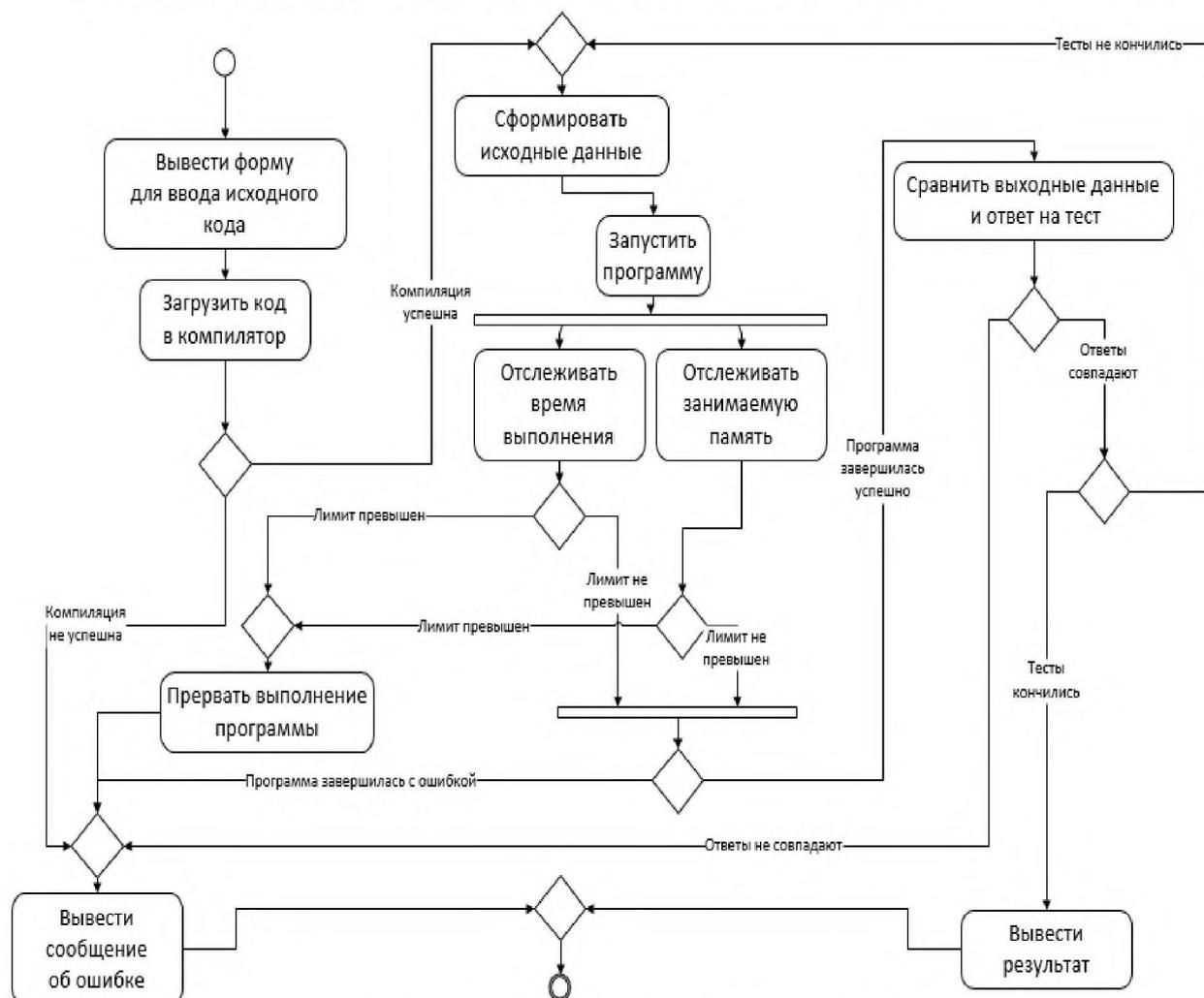


Рис. 4 Диаграмма деятельности, ядро системы

### Интерфейс

Учитывая назначение системы, интерфейс должен быть простым и лаконичным. Таким образом, основной интерфейс пользователя состоит из 6 страниц:

- 1) Список задач, в котором представлены номера и названия задач.
- 2) Страница задачи, на которой представлено условие задачи и примеры тестов.
- 3) Форма отправки решения с тремя полями: номер задачи, выбор языка программирования из списка, текстовое поле для исходного кода.
- 4) Таблица результатов проверки, в которой указаны название задачи, имя пользователя, время отправки, время работы, статус и номер теста, на котором был неверный ответ, если не все тесты пройдены, сообщение об ошибке (открывается на новой странице), если есть ошибка.
- 5) Страница со списком прошедших соревнований.
- 6) Итоги соревнования в виде таблицы, в которой показано, сколько тестов прошёл каждый участник соревнования в каждой задаче.

### Выводы

Проектируемая система разрабатывается на кафедре ПОКС КГТУ им.И.Раззакова, это даёт КГТУ возможность организовывать свои соревнования, в частности республиканские олимпиады по информатике для школьников и олимпиады по программированию среди студентов нашего университета, что и является основной причиной для разработки. Текущая

разработка предлагает те же основные функции, что и системы-аналоги, но отличается тем, что подсчет баллов будет вестись и по правилам ACM ICPC для студенческих олимпиад по программированию, и по правилам IOI для школьных олимпиад по информатике. Разработка этой системы выводит проведение Республиканской олимпиады школьников по информатике в Кыргызстане на уровень международных олимпиад.

### Список литературы

1. Сайт проведения международных студенческих командных соревнований: <https://icpc.baylor.edu>
2. Сайт проверочной системы КPCY <http://olymp.krsu.edu.kg/>
3. Сайт проверочной системы ТИМУС <http://acm.timus.ru/>
4. Корнеев Г.А., Станкевич А.С. Методы тестирования решений задач на соревнованиях по программированию. //Труды II межвузовской конференции молодых ученых. – СПб.: СПбГУ ИТМО, 2004. С.36-40.
5. Станкевич А.С. Общий подход к подведению итогов соревнований по программированию при использовании различных систем оценки. //Компьютерные инструменты в образовании. №2, 2011. С 27-38.
6. Макиева З.Дж., Каримова Г.Т., Макаева А.Д. Требования к веб-ориентированной информационной системе по секторальной квалификационной рамке Кыргызской Республики для ИКТ направлений. //Теоретический и научно-методический журнал «Вестник университета (ГУУ)» ФГБОУ ВПО «Государственный университет управления», Москва, № 19/2014. С.167-172.

UDC 025.4.03

## APPLICATION OF MORPHOLOGICAL MARKUP OF KAZAKH LANGUAGE TO AUTOMATED FILLING OF THE ONTOLOGY OF FACTOGRAPHIC RETRIEVAL SYSTEM

*Madina Mansurova, al-Farabi Kazakh National University, Almaty, Kazakhstan,*

*Kairat Koibagarov; Institute of Information and Computational Technologies, Almaty, Kazakhstan.*

*Vladimir Barakhnin; Institute of Computational Technologies SB RAS, Novosibirsk, Russia, Novosibirsk State University, Novosibirsk, Russia.*

*Madina Soltangeldinova, al-Farabi Kazakh National University, Almaty, Kazakhstan.*

*Serzhan Berdibekov, al-Farabi Kazakh National University, Almaty, Kazakhstan.*

**Abstract.** This work is concerned with the development of the parser for automation of morphological markup of the texts of the Kazakh National corpus. The parser includes the lexical and morphological analyzers to perform the morphological markup of the texts. The task of a lexical analyzer is to determine the boundaries of sentences, to display words, identifiers and punctuation marks. The morphological analyzer performs the search for words in the dictionary (which is a separate database) and determines their morphological parameters. At the output of the morphological analyzer, we have a list of lemmas (a normal form of the word), affixes and morphological characteristics of the word. Morphological markup of the texts is a stage of automatic text processing, which allows to use the marked texts to solve the different problems of Natural Language processing. This paper describes the application of morphological parser of the Kazakh language to automated filling of the ontology of factographic retrieval system.

**Keywords:** Morphological Parser, Morphological Markup, Factographic Retrieval, Facts Extraction, Automated Ontology Filling.