

УДК 622.862.3: 622.831.312: 004.42

**Шевченко В.Г.**, д-р техн. наук, ст. науч. сотр.,  
**Слащев А.И.**, аспирант  
(ИГТМ НАН Украины)

**РАЗРАБОТКА СХЕМ И АЛГОРИТМА ФУНКЦИОНИРОВАНИЯ  
ИНФОРМАЦИОННОГО КОМПЛЕКСА УПРАВЛЕНИЯ ПЕРСОНАЛОМ  
НА УРАНОВЫХ ШАХТАХ**

**Шевченко В.Г.**, д-р техн. наук, ст. науч. співроб.,  
**Слащов А.І.**, аспірант  
(ІГТМ НАН України)

**РОЗРОБКА СХЕМ І АЛГОРИТМУ ФУНКЦІОНУВАННЯ  
ІНФОРМАЦІЙНОГО КОМПЛЕКСУ КЕРУВАННЯ ПЕРСОНАЛОМ  
НА УРАНОВИХ ШАХТАХ**

**Shevchenko V.G.**, D.Sc. (Tech.), Senior Researcher,  
**Slashchov A.I.**, Doctoral Student  
(IGTM NAS of Ukraine)

**DEVELOPMENT OF SCHEMES AND ALGORITHMS OF FUNCTIONING  
INFORMATION COMPLEX PERSONNEL MANAGEMENT  
FOR URANIUM MINES**

**Аннотация.** Статья посвящена решению актуальной задачи разработки информационного мобильного комплекса управления персоналом на урановых шахтах для повышения безопасности работ, эффективности взаимодействия и уровня контроля руководителями производственных процессов.

Разработаны схемы, алгоритмы и библиотеки программного комплекса управления персоналом, который обеспечивает пересылку письменных сообщений от сервера к пользователю и обратно; передачу фотографий выполненных работ на сервер; передачу с сервера масштабированных или разбитых на части изображений; автоматическую регистрацию и сохранение в базе данных сервера пересылаемой информации; генерацию запросов о получении информации после ее пересылки; уведомление об особо важных сообщениях; автоматическую настройку всей выводимой информации под размер экрана мобильного устройства.

Повышение производительности процесса вычислений системы управления персоналом достигается путем рейтингования ресурсов по частоте их использования с учетом параметра стабильности групп классов проекта. При этом рейтинг малоиспользуемых ресурсов обратно пропорционален рейтингу более стабильных экземпляров класса, что позволяет в процессе выполнения программы выгружать малозначимые ресурсы и снижать ее ресурсоемкость. На основе вычислительных экспериментов предложена формула для определения переменной, характеризующей добавляемый рейтинг ресурса при его однократном использовании.

**Ключевые слова:** мобильная информационная система, схемы и алгоритмы, управление персоналом, урановые шахты, повышение безопасности работ.

Эффективность работы урановых шахт зависит от слаженности функционирования не только технологической и технической подсистем, но и подсистемы

управления персоналом. На горных предприятиях эта подсистема имеет свою специфику, обусловленную наличием значительного количества работников, управлением сложным оборудованием, большой вероятностью аварийных ситуаций, удаленностью руководителей от производственных участков. Эти факторы требуют быстрого и качественного обмена информацией между руководителями и подчиненными.

Передача аудиоинформации, в основном, решена телефонизацией. Вместе с тем на удаленных объектах, где выполняются ответственные работы, стационарная телефонная связь может отсутствовать. Устные недокументируемые распоряжения достаточны при решении задач малой или средней сложности и только в небольших стабильных коллективах. Однако даже в таких коллективах существует возможность неоднозначного толкования смысла устных распоряжений как отдающим, так и получающим распоряжение работником. При этом может возникнуть расхождение между тем, что руководитель хотел сказать, и как сказал на самом деле; как подчиненный понял, и что принял к исполнению.

Изменить такую ситуацию может письменное распоряжение, переданное визуально. Письменное распоряжение предпочтительнее еще и потому, что человек запоминает ограниченный объем информации, а ее визуализация позволяет, при необходимости, неоднократно уточнять поставленную задачу. Кроме того, работнику может срочно понадобиться техническая информация: горно-геологический разрез, карта выхода водоносных горизонтов, принципиальные схемы оборудования, технологические планы разработок и др. Этот комплекс управленческих вопросов может быть решен с помощью информационной системы управления персоналом, обеспечивающей, в том числе, быструю и качественную передачу визуальных данных.

Сегодня на производстве нет достаточно удобной технологии передачи визуальной информации, хотя такая задача является актуальной и востребованной, а ее решение существенно упростит организационные вопросы и процесс управления персоналом. Практически каждый работник горного предприятия имеет мобильное устройство (телефон, планшет), и на каждом горном предприятии есть интернет и серверы. То есть, на предприятиях уже есть часть технической базы для развертывания комплекса по управлению персоналом, а его внедрение не потребует значительных затрат. Разработка мобильной технологии управления персоналом для урановых шахт позволит повысить безопасность и эффективность работ путем внедрения современных технических и программных средств.

На сегодняшний день технические системы в горном производстве контролируют аэрогазовый режим шахт, технологические параметры и напряженное состояние породного массива. В частности, телекоммуникационная система диспетчерского и автоматизированного управления горными машинами и комплексами «УТАС» обеспечивает непрерывные измерения параметров состояния промышленных и горно-технологических объектов, система АПСС-1 и ЗУА-98 выполняет прогноз выбороопасности пластов на базе акустических методов, система «АЛЕРТ» выполняет классификацию текущего состояния

безопасности шахты, автоматическое извещение о предаварийном и аварийном состояниях и другие системы [1]. Вместе с тем отмечается, что тяжелые аварии на шахтах, связанные с взрывами газа и пыли происходят, в основном, по организационным причинам [2]. Из анализа обстоятельств и причин крупных аварий в горном производстве, произошедших за последние годы, установлено, что все реже их причиной является отказ современного оборудования, а на первое место выходит человеческий фактор [3]. Поэтому задача повышения дисциплины и исполнительности сотрудников на производстве является приоритетной. Ее решение выполнимо при совершенствовании системы контроля над исполнением выданных инженерно-техническими работниками приказов и распоряжений, а также повышении дисциплинарной ответственности за соблюдением правил безопасности [4].

Предлагаемая система управления персоналом представляет собой программный комплекс, который адаптирован под нужды урановых шахт. Она предназначена для быстрой передачи визуальной информации и выполняет следующие функции: пересылки письменных сообщений от сервера к пользователю и наоборот; передачи фотографий выполненных работ на сервер (для контроля руководителями, отвечающими за производство работ); передачи с сервера потребителю масштабированных или разбитых на части рисунков (геологических разрезов, блок-схем и др. графической информации); автоматической регистрации пересылаемой информации и сохранения копий в базе данных сервера (для возможности последующего контроля); генерации запроса о получении информации после ее пересылки с просьбой об обратном уведомлении и регистрацией на сервере факта получения сообщения; обязательного уведомления особо важных сообщений (сигнал работает до тех пор, пока пользователь не отреагирует на экстренное сообщение); автоматической настройки всей выводимой информации под размер экрана мобильного устройства и поворота изображения при повороте экрана.

Программный комплекс позволит визуально передавать указания руководителя и, при необходимости, получать визуальный отчет об их выполнении. Автоматическое сохранение всех переданных через систему приказов и фотографий результатов работ повысит уровень контроля над производственными процессами. При необходимости (в случае аварии, квалификационной оценки персонала, периодического контроля для повышения дисциплины и др.), проведя анализ задокументированных на сервере приказов и фотоснимков выполненных работ, можно определить компетентность и качество работы сотрудников. Эффективность взаимодействия между иерархичными структурами персонала обеспечивается скоростью и автоматизацией работы системы. Например, директор предприятия, продиктовав приказ, за секунды разошлет его любой по количеству группе сотрудников, а через несколько минут он будет знать, кто конкретно из сотрудников его распоряжение не получил.

Выбор языка программирования Java [5] был обусловлен следующими обстоятельствами: возможностью запуска программы в любой операционной системе и на любом устройстве, которое поддерживает виртуальную машину Java;

надежностью программного кода и простоту отладки (Java проверяет выходы за пределы массивов и еще множество исключительных ситуаций); возможностью доступа к объектам в Java путем использования ссылочных типов данных, что позволяет эффективно обрабатывать данные и передавать их в функции; универсальностью обработки любых программных классов.

Браузер мобильного устройства (программа для взаимодействия с сервером и выполнения ряда специальных функций) спроектирован так, чтобы избегать обработки излишней информации. Например, если есть возможность быстро вычислить ширину и высоту графического элемента, то ранее вычисленные габариты не хранятся, так как количество графических элементов может измеряться тысячами, а на каждый элемент нужно дополнительно 8 байт памяти. Минимизация числа обращений к хранимым данным на сервере или на винчестере достигается тем, что программа избегает частых вызовов длительно выполняющихся функций.

В браузере реализован менеджер управления ресурсами [6], который обеспечивает хеширование часто используемых данных (преобразование входного массива данных в короткое число фиксированной длины), благодаря чему уменьшено обращение к серверу и постоянной памяти мобильного устройства. Например, приведем алгоритм загрузки изображений (рис. 1). Как видно, загрузка с сервера происходит только тогда, когда данное изображение отсутствует в браузере мобильного устройства, причем если памяти хватает, то каждое изображение с сервера загружается только один раз.

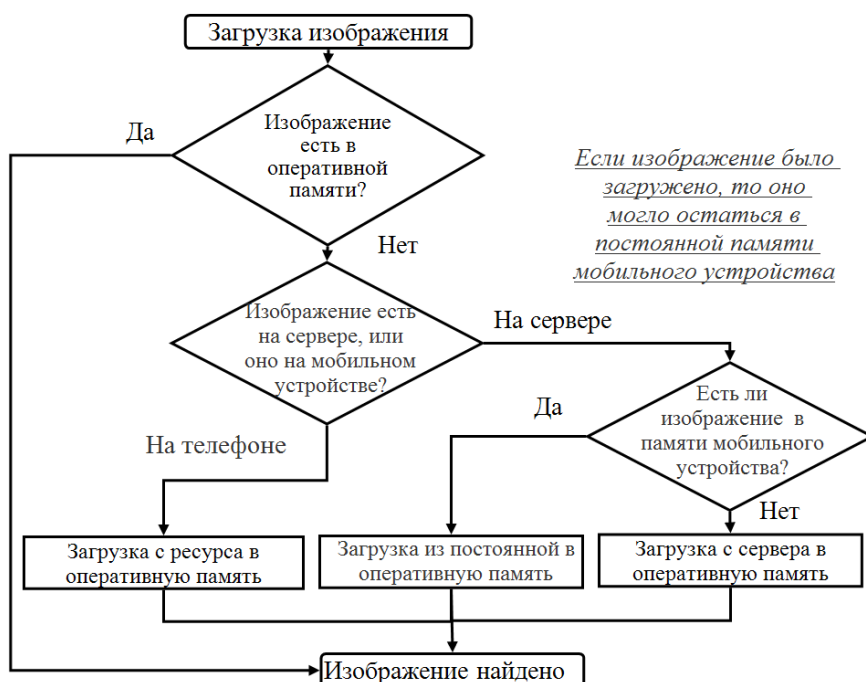


Рисунок 1 – Алгоритм загрузки изображений в системе управления персоналом

Как известно, чем чаще системе приходится копировать данные с сервера, тем на него больше нагрузка и тем медленнее работает программа, которая будет тратить время на подкачку данных вместо выполнения расчетов. Поэтому,

уменьшая обращения компьютера к ресурсам на сервере, мы увеличиваем общую производительность системы. Если памяти достаточно для выполнения приложения, то неиспользуемые ресурсы освобождаются только в конце выполнения программы. При этом менеджер «знает» обо всех ресурсах и не допускает их дублирования, что повышает быстродействие приложения.

Если ресурс захватит слишком много оперативной памяти и ее не хватает, это приводит к интенсивной передаче данных на сервер и резкому падению производительности всей системы. В этом случае производится поиск и удаление неиспользуемых данных, а малоиспользуемые данные в течение некоторого времени сохраняются на сервере или архивируются.

В программный класс хранения параметров ресурса (*ResourceParametres*) включаются много специфических параметров. Например: время загрузки ресурса; время его последнего использования; переменные для расчета рейтинга; флаги, указывающие путь извлечения информации (это позволяет не анализировать имя ресурса) и др. Варианты реализации класса *ResourceParametres* приведены в табл. 1.

Таблица 1 - Класс хранения параметров ресурса *ResourceParametres* в реализациях на языках программирования C++ (вариант 1), Java и C# (вариант 2)

Вариант 1	Вариант 2
<i>Компилятор</i> , поддерживающий указатели, например, C++.	<i>Интерпретатор</i> , не поддерживающий указателей, который, в основном, работает со ссылками на классы и все классы наследуют класс Object. Например, Java и C#.
<pre>ResourceParametres ResourceParametres(); int iCount; float fRating, fRatingAdd; void *cRes; .....</pre>	<pre>ResourceParametres ResourceParametres(); int iCount; float fRating, fRatingAdd; Object cRes; .....</pre>
<i>cRes</i> – указатель, связывающий класс параметров ресурса с ресурсом	<i>cRes</i> – ссылка, связывающая класс параметров ресурса с ресурсом
Освобождение памяти осуществляется в деструкторе класса менеджера ресурсов после синхронизации	Память освобождается стандартным методом после обнуления ссылок во внешних классах и в менеджере ресурсов

В данном классе: *iCount* – количество классов, использующих ресурс; *fRating* – рейтинг ресурса, чем он выше, тем ресурс важнее; *fRatingAdd* – добавляемый рейтинг ресурса при его однократном использовании. Конструктор класса показан на рис. 2. Коэффициент *k* необходим для того, чтобы при первой загрузке ресурса его рейтинг не был занижен. К моменту загрузки нового ресурса, скорее всего другие используемые ресурсы будут иметь значение *fRating* больше чем *fRatingAdd* и, чтобы новый ресурс не был сразу малорейтинговым, коэффициент *k* должен быть больше единицы, например, равен двум.



Рисунок 2 – Блок-схема конструктора класса для хранения параметров ресурса

В менеджере управления ресурсами применен статистический метод поиска малоиспользуемой информации, когда рейтинг ресурсу назначается на основе частоты его использования. Чем больше значение переменной  $fRating$ , тем ресурс более важен и больше используется. Установки рейтинга ресурсу основаны на принципах изменения эффективности работы информационной системы:

- чем больше используется объект, тем он более важен, поэтому при каждом использовании объекта переменная  $fRating$  наращивается на величину переменной  $fRatingAdd$ ;

- чем дольше загружается объект, тем он важнее, поэтому этому ресурсу прибавляется рейтинг путем увеличения коэффициента, показывающего во сколько раз дольше идет загрузка с сервера, чем с мобильного устройства (изображения с сервера нужно хранить дольше);

- чем меньше объект, тем он важнее, поэтому если объект занимает мало физической памяти, то значение  $fRatingAdd$  увеличивается, так как один большой объект загружается быстрее, чем много маленьких, которые в сумме дают такой же размер (большие объекты легче перегружать в оперативную память, если они не используются много раз);

- чем больше нестабильность у объекта, тем он более важен (если объект нестабилен, это говорит о наличии множества ссылок на него извне).

На основе серии вычислительных экспериментов предложена формула для определения параметра  $fRatingAdd$

$$fRatingAdd = k_v \left( 2 - \frac{v - v_{min}}{v_{max} - v_{min}} \right) + k_V \left( 2 - \frac{V - V_{min}}{V_{max} - V_{min}} \right) + k_I \left( 2 - \frac{I - I_{min}}{I_{max} - I_{min}} \right), \quad (1)$$

где  $v$ ,  $v_{min}$ ,  $v_{max}$  – текущая, минимальная и максимальная скорости загрузки ресурсов;  $V$ ,  $V_{min}$ ,  $V_{max}$  – текущий, минимальный и максимальный размеры ресурсов программы;  $I$ ,  $I_{min}$ ,  $I_{max}$  – текущая, минимальная и максимальная нестабильность категории классов, имеющих отношение к ресурсам;  $k_v$ ,  $k_V$ ,  $k_I$  – управляющие коэффициенты от нуля до единицы, характеризующие относительные приоритеты и влияние относительно других рейтингов скорости загрузки ресурса, размеров ресурсов и нестабильности категории классов, соответственно.

Управляющие коэффициенты находятся в диапазоне  $[0; 1]$  и в сумме равны единице, то есть ограничения и допущения этой зависимости можно представить в виде

$$k_U[0; 1] + k_V[0; 1] + k_I[0; 1] = 1; \quad fRatingAdd = [1; 2] . \quad (2)$$

Переменная  $fRatingAdd$  имеет значение от одной до двух единиц и необходима для сравнительной характеристики ресурсов. По этой формуле хорошо выполняется сравнение относительных рейтингов ресурсов. Если используется программа с графическим интерфейсом на локальном компьютере и в ней не более двух-трех тысяч ресурсов, то этот пересчет можно осуществлять один раз в прорисовку или через несколько прорисовок по формуле  $fRating = 0,99 * fRating$ , которая хорошо себя показала при расчетах. Если менеджер обрабатывает ресурсы на сервере, то рациональнее выполнять пересчет раз в несколько минут или более по формуле  $fRating = 0,6 \div 0,9 * fRating$ .

Предложенный метод управления ресурсами программы имеет ряд существенных преимуществ и позволяет: проводить автоматическое наблюдение за динамической памятью (при переполнении находить данные с низким рейтингом и сохранять их на винчестере с последующим восстановлением); ускорить расчеты за счет группирования маленьких буферов; повысить удобство доработки программы с использованием одного интерфейса для загрузки разных данных, не разрабатывать повторного кода загрузки и кэширования данных в разных местах программы. В результате повышается вычислительная эффективность информационной системы по надежности, скорости прорисовки графических примитивов, используемой памяти и процессорному времени.

Таким образом, повышение продуктивности процесса вычислений достигается путем рейтингования ресурсов по частоте их использования с учетом параметра стабильности групп классов [7], при этом рейтинг малоиспользуемых ресурсов обратно пропорционален рейтингу более стабильных экземпляров класса, что позволяет в процессе выполнения программы выгружать малозначимые ресурсы и снижать ее ресурсоемкость.

Для защиты системы использована процедура входа с именем пользователя и паролем. Данные передаются по индивидуальному протоколу. Этот протокол предусматривает превращение классов в массив данных с последующим архивированием. Архивирование выполняется, чтобы уменьшить нагрузку на сеть и сделать передачу данных более быстрой.

Древовидная архитектура построения браузера дает возможность динамически добавлять и менять элементы управления, рис. 3. Например, можно легко поменять первую и вторую группу местами, при этом нет необходимости контролировать перемещение внутренних элементов. Это упрощает доработку элементов управления и логических элементов. Для добавления новой функциональности достаточно только дописать новый программный класс с реализацией нужного интерфейса. Покажем на примере, как создан элемент управления `CheckBox`, который указывает, включено или отключено какое-либо кон-

кретное условие. За основу взят класс кнопки, в который добавили переменную типа Boolean, а также обработчик события нажатия на кнопку и доработанную функцию рисования. В результате реализации мы получаем все функции кнопки: при нажатии кнопки меняется ее вид, выполняется скрипт (небольшая функция), появляется возможность использования текста и изображения.

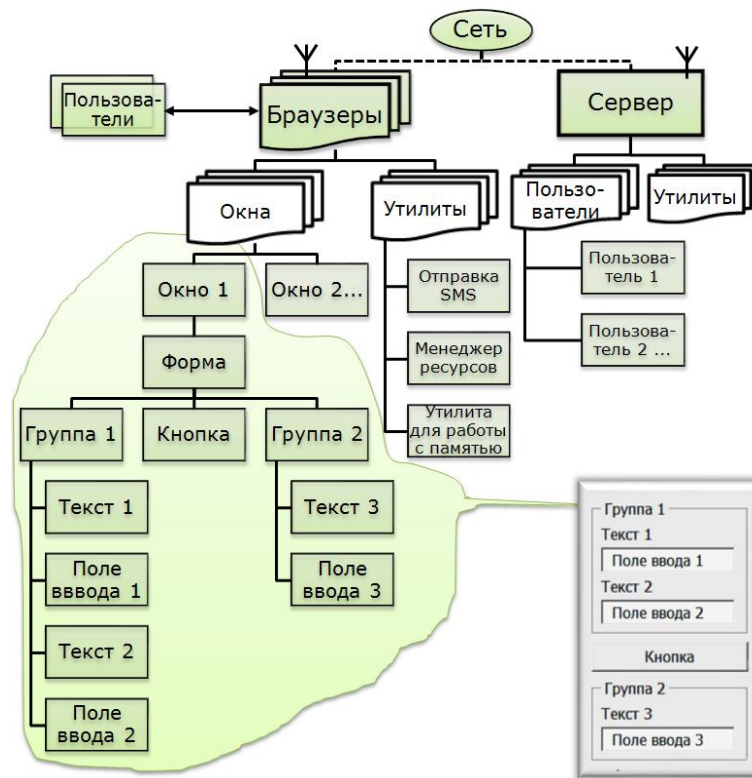


Рисунок 3 – Структурная схема построения системы управления персоналом

На сервере расположены утилиты, объединяющие множество пользователей и выполняющие взаимодействие между ними. Утилиты – это программы, отвечающие за выполнение определенных функций браузера, например, отсылка SMS-сообщений или сохранение данных в долговременную память мобильного устройства, считывание данных о системе и др. Взаимодействие элементов древовидной структуры между собой и утилитами производится с помощью внутренних сообщений, которые представляют собой программные классы, имеющие данные об адресате, о действии, которое адресат должен сделать, и другие параметры. Такая архитектура построения алгоритма дает возможность расширять функциональность программы, поскольку позволяет вызывать практически неограниченный набор функций через одну функцию.

Для информационной системы управления персоналом разработан ряд библиотек, которые созданы с учетом возможности быстрого увеличения функций программного комплекса в случаях его расширения по требованию производителей. Эти библиотеки служат своеобразным многофункциональным конструктором, помогающим качественно и эффективно расширять возможности серверной и браузерной частей информационной системы. В табл. 2 показана



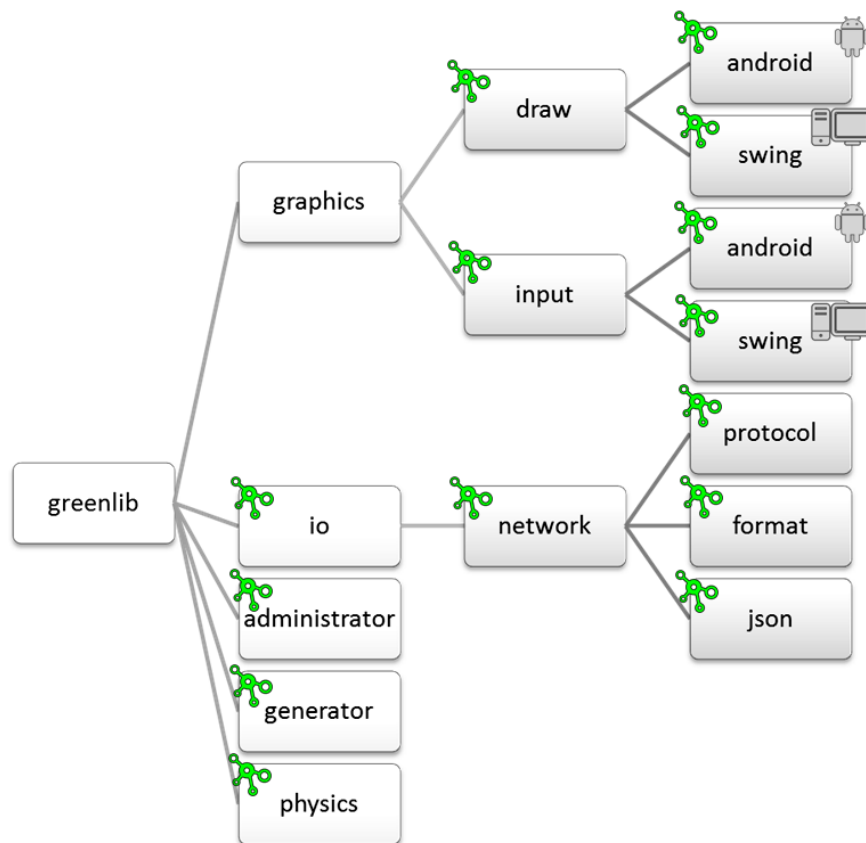
краткая характеристика функциональных возможностей программных библиотек, и их привязка к той части проекта, для которой они разработаны.

Таблица 2 – Краткая характеристика функциональных возможностей разработанных библиотек программного комплекса

Название модуля, составляющая браузера или сервера	Описание
greenlib.graphics.draw, браузер	Общие классы и интерфейсы, которые нужны для вывода графики. Упрощающие классы, которые дают возможность писать кроссплатформенный интерфейс пользователя
greenlib.graphics.draw.android, браузер	Платформозависимая привязка к операционной системе Android (OpenGL ES 2)
greenlib.graphics.draw.swing, браузер	Привязка к JOGL (Java OpenGL)
greenlib.graphics.input, браузер	Система для рекурсивной обработки событий мыши или сенсорного экрана (поиск компонентов, обозначенных на экране мышью и доставка событий компоненту)
greenlib.graphics.input.android, браузер	Привязка greenlib.graphics.input к стандартному вводу Android
greenlib.graphics.input.swing, браузер	Привязка greenlib.graphics.input к стандартному вводу Swing
greenlib.io, браузер, сервер	Система улучшенной сериализации (удобна в использовании и работает быстрее, чем стандартная)
greenlib.io.network, браузер, сервер	Система клиент-серверной передачи байтовых массивов или буферов по сети. Дополнительно реализована система расширений, с помощью которой можно архивировать, шифровать, или проверять данные на целостность
greenlib.io.network.protocol, браузер, сервер	Позволяет поверх простого протокола реализованного в greenlib.io.network надстроить свой протокол
greenlib.io.network.protocol.format, браузер, сервер	Реализация протокола с помощью greenlib.io.network.protocol и сериализации greenlib.io
greenlib.io.network.protocol.json, браузер, сервер	Реализация протокола с помощью greenlib.io.network.protocol и сериализации в Json
greenlib.administrator, сервер	Система мониторинга и консоль для серверов
greenlib.generator	Вспомогательный код для генерации исходного кода и упрощения рутинных операций
greenlib.physics, браузер	Математическая библиотека для работы с векторами матрицами и объектами

Необходимость реализации быстрого расширения проекта обусловлена тем, что, как показала практика, исходное техническое задание на программное обеспечение служит только основой проекта и не всегда может удовлетворить растущие требования производителей в процессе работы с программными продуктами. Вследствие чего требуется добавление и создание новых функциональных возможностей системы.

В графической визуализации используются два пакета *android* для мобильных телефонов и планшетов, *swing* – для компьютеров. В библиотеку *draw* входят интерфейсы и функции для рисования на экраны планшетов и мобильных устройств. Прорисовка производится с помощью треугольников и текстур их заполняющих. Сервер хранит данные, принимает запросы и выдает ответы, а графическая прорисовка выполняется браузером. Соответственно три графических библиотеки и три библиотеки для ввода событий с клавиатуры и мыши находятся в *greenlib.graphics* и относятся к браузерной части проекта. Иерархия разработанных библиотек представлена на рис. 4.



- модуль (функционально законченный фрагмент программы);
- платформозависимый код для PC (Windows, Linux);
- платформозависимый код для Android (планшеты, телефоны, телевизоры)

Рисунок 5 – Иерархия библиотек для информационной системы управления персоналом

Библиотеки *greenlib.io* ... созданы для передачи данных по сети и сохранения данных на сервере, *greenlib.administrator* – для помощи при отладке и мониторинге сервера. С использованием библиотеки *greenlib.generator* было сгенерировано часть кода *greenlib.io*, что уменьшило количество потенциальных ошибок и ускорило разработку. Вспомогательная библиотека *greenlib.physics* была использована для математических расчетов в *greenlib.graphics.draw*. Все библиотеки написаны на языке Java, как пример представлена часть разработанного математического модуля *greenlib.physics*.

**Выводы.**

1. Эффективность работы урановых шахт зависит от слаженности функционирования не только технологической и технической подсистем, но и подсистемы управления персоналом. Анализ обстоятельств крупных аварий показал, что первое место среди их причин занимает человеческий фактор. Подсистема управления персоналом на горном производстве имеет свою специфику, связанную со значительным количеством работников, с управлением сложным оборудованием, большой вероятностью аварийных ситуаций, удаленностью руководителей высшего звена от производственных участков. Эти факторы требуют быстрого и качественного информационного взаимодействия руководителей и подчиненных, которое может быть обеспечено разработкой и внедрением на производстве современной информационной мобильной системы управления персоналом. Создание адаптированного к условиям производства программного обеспечения повышает безопасность работ и эффективность взаимодействия персонала, дисциплинарную ответственность, уровень контроля руководителей над производственными процессами.

2. Разработаны схемы, алгоритмы, ряд библиотек программного комплекса управления персоналом, который предназначен для: пересылки письменных сообщений от сервера к пользователю и обратно; передачи фотографий выполненных работ на сервер; передачи с сервера потребителю масштабированных или разбитых на части изображений (геологических разрезов, блок-схем и др.); автоматической регистрации и сохранения в базе данных сервера пересылаемой информации (для возможности последующего просмотра и анализа); генерации запроса о получении информации после ее пересылки с просьбой об обратном уведомлении и регистрацией на сервере факта получения сообщения; обязательного уведомления об особо важных сообщениях (сигнал работает до тех пор, пока пользователь не отреагирует на экстренное сообщение); автоматической настройки всей выводимой информации под размер экрана мобильного устройства и поворота изображения при повороте экрана.

3. Повышение продуктивности процесса вычислений системы управления персоналом достигается путем рейтингования ресурсов по частоте их использования с учетом параметра стабильности групп классов проекта. При этом рейтинг малоиспользуемых ресурсов обратно пропорционален рейтингу более стабильных экземпляров класса, что позволяет в процессе выполнения программы выгружать малозначимые ресурсы и снижать ее ресурсоемкость. На основе вычислительных экспериментов предложена формула для определения переменной, характеризующей добавляемый рейтинг ресурса при его однократном использовании.

---

**СПИСОК ЛИТЕРАТУРЫ**

1. Геомеханічний моніторинг підземних геотехнічних систем : монографія / Відповід. ред. А.Ф. Булат / А.В. Анциферов, С.І. Скіпочка, А.О. Яланський [та ін.]. – Донецьк: «Ноулідж», 2010. – 253 с.

2. Ивашин, В.М. Как предотвратить аварийность на угольных шахтах Украины / В.М. Ивашин, Ю.В. Ивашин, Г.П. Штапаук// Науковий вісник НГУ. – Дніпропетровськ: РВК НГУ, 2009. - № 8. - С. 32-36.

3. Задачи научного обеспечения мониторинга и психофизических исследований в работе с кадрами угольной отрасли / Ю.И. Кияшко, В.Г. Шевченко, К.В. Присняков [и др.] / Геотехническая механика: Межвед. сб. науч. тр. / ИГТМ НАН Украины. – Днепропетровск, 2010. – Вып. 90. – С. 71-78.

4. Слащев, А.И. Разработка мобильной технологии управления персоналом на предприятиях горного профиля / А.И. Слащев // Геотехническая механика: Межвед. сб. науч. тр. / ИГТМ НАН Украины. – Днепропетровск, 2011. – Вып. 95. – С. 142-149.

5. Нортон, П. Полный справочник по Java : Пер. с англ. / П. Нортон, Г. Шилдт. – К.: Диалектика, 1997. – 592 с.

6. Слащев, А.И. Повышение вычислительной эффективности алгоритма визуализации результатов решения сложных геомеханических задач / А.И. Слащев // Г Геотехническая механика: Межвед. сб. науч. тр. / ИГТМ НАН Украины. – Днепропетровск, 2010. – Вып. 90. – С. 114-128.

7. Слащев, И.Н. Оптимизация информационной системы оперативного прогноза геомеханических процессов для поддержки принятия решений по безопасности шахт / И.Н. Слащев, В.Г. Шевченко, А.И. Слащев // Геотехническая механика: Межвед. сб. науч. тр. / ИГТМ НАН Украины. – Днепропетровск, 2013. – Вып. 112. – С. 129-143.

---

#### REFERENCES

1. Antsiferov, A.V., Skipochka, S.I. and Yalanskyu, A.A. (2010), *Geomekhanichnyy monitoring pidzemnykh geotekhnichnykh system* [Geomechanical underground geotechnical monitoring systems], Noulidzh, Donetsk, Ukraine.

2. Ivashin, V.M., Ivashin, Y.V. and Shtapauk, G.P. (2009), “How to prevent accidents at coal mines in Ukraine”, *Naukovyi visnyk Natsionalnoho hirnychoho universytetu*, no. 8, pp. 32-36.

3. Kiyashko, J.I., Shevchenko, V.G., Prisnyakov, K.V. and Padashulya, A.I. (2011), “Objectives of the monitoring and scientific research in psychophysical work with staff of the coal industry”, *Geotekhnicheskaya Mekhanika* [Geo-Technical Mechanics], no. 90, pp. 71-78.

4. Slashchov, A.I. (2011), “Development of mobile technology for personnel management at enterprises in the mining”, *Geotekhnicheskaya Mekhanika* [Geo-Technical Mechanics], no. 95, pp. 142-149.

5. Norton, P. and Shildt, G. (1997), *Polnyy spravochnik po Java* [Complete Reference Java], Dialectic, Kiev, Ukraine.

6. Slashchov, A.I. (2010), “Improving the computational efficiency of the algorithm results visualization solutions to complex geomechanical problems”, *Geotekhnicheskaya Mekhanika* [Geo-Technical Mechanics], no. 90, pp. 114-128.

7. Slashchev, I.N., Shevchenko, V.G. and Slashchov, A.I. (2013), “Optimized information system for on-line predicting of geomechanical process behavior and ensuring proper decision-making on the mine safety”, *Geotekhnicheskaya Mekhanika* [Geo-Technical Mechanics], no. 112, pp. 129-143.

---

#### Об авторах

**Шевченко Владимир Георгиевич**, доктор технических наук, старший научный сотрудник, ученый секретарь института, Институт геотехнической механики им. Н.С. Полякова Национальной академии наук Украины (ИГТМ НАНУ), Днепропетровск, Украина, [V.Shevchenko@nas.gov.ua](mailto:V.Shevchenko@nas.gov.ua).

**Слащев Антон Игоревич**, магистр, аспирант, Институт геотехнической механики им. Н.С. Полякова Национальной академии наук Украины (ИГТМ НАН Украины), Днепропетровск, Украина, [SlashovAnton@ya.ru](mailto:SlashovAnton@ya.ru).

#### About the authors

**Shevchenko Vladimir Georgievich**, Doctor of Technical Sciences (D. Sc), Senior Researcher, Scientific Secretary of the Institute, M.S. Polyakov Institute of Geotechnical Mechanics under the National Academy of Sciences of Ukraine (IGTM, NASU), Dnepropetrovsk, Ukraine, [V.Shevchenko@nas.gov.ua](mailto:V.Shevchenko@nas.gov.ua).

**Slashchov Anton Igorevich**, Master of Science, Doctoral Student, M.S. Polyakov Institute of Geotechnical Mechanics under the National Academy of Science of Ukraine (IGTM, NASU), Dnepropetrovsk, Ukraine, [SlashovAnton@ya.ru](mailto:SlashovAnton@ya.ru).

**Анотація.** Стаття присвячена рішенням актуального завдання розробки інформаційного мобільного комплексу керування персоналом на уранових шахтах для підвищення безпеки робіт, ефективності взаємодії і рівня контролю керівниками виробничих процесів.

Розроблено схеми, алгоритми і бібліотеки програмного комплексу керування персоналом, що забезпечує пересилання письмових повідомлень від сервера до користувача і назад; передачу фотографій виконаних робіт на сервер; передачу із сервера масштабованих або розбитих на частині зображень; автоматичну реєстрацію і збереження в базі даних сервера інформації, що пересилається; генерацію запитів про одержання інформації після її пересилання; повідомлення про особливо важливі повідомлення; автоматичне настроювання всієї виведеної інформації під розмір екрана мобільного пристрою.

Підвищення продуктивності процесу обчислень системи керування персоналом досягається шляхом рейтингування ресурсів по частоті їхнього використання з урахуванням параметра стабільності груп класів проекту. При цьому рейтинг маловикористовуваних ресурсів обернено пропорційний рейтингу більше стабільних екземплярів класу, що дозволяє в процесі виконання програми вивантажувати малозначимі ресурси й знижувати її ресурсоемність. На основі обчислювальних експериментів запропонована формула для визначення змінної, що характеризує рейтинг, що додається, ресурсу при його однократному використанні.

**Ключові слова:** інформаційний мобільний комплекс, схеми і алгоритми, керування персоналом, уранові шахти, підвищення безпеки робіт.

**Abstract.** The article covers a pressing problem of designing a mobile information complex on human resource management for uranium mines with the view of improving safety of works, degree of control of and efficiency of interaction between the heads of production processes.

Schemes, algorithms and libraries have been designed for the HRM software complex which provides: feedback between server and user in the form of written messages; transmission of performed work photos to the server; transmission of scaled or split images from the server; automatic recording of received information and its saving in the server database; generating of requests on receiving information after its delivery; notification about the most important messages; automatic output information adjustment to the format of a mobile device screen.

Computation processes in the HRM system can be improved by establishing ratings of resources by frequency of their use with taking into account stability parameters of the project class groups. As low-activity resources are inversely proportional to the ratings of more stable classes it is possible to unload the low-activity resources and, therefore, reduce resource intensity of the program in whole. Basing on results of computational experiments a formula is proposed for determining a variable which characterizes an added rating of a resource at each its use.

**Keywords:** mobile information complex, schemes and algorithms, human resource management, uranium mines, improved safety of works.

*Стаття постуила в редакцію 27.10.2014*

*Рекомендовано к печати д-ром техн. наук Т.В. Бунько*