

Григорьев П.Е., Килесса Г.В., Хорсева Н.И., Овсянникова Н.М.

ИНФОРМАЦИОННО-ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ КОМПЛЕКСНОГО МОНИТОРИРОВАНИЯ И ЭКСПРЕСС- ТЕСТИРОВАНИЯ ПСИХОФИЗИОЛОГИЧЕСКОГО СОСТОЯНИЯ ЧЕЛОВЕКА

Представлены архитектура и функции программно-информационного обеспечения, предназначенного для комплексного мониторинга состояния человека с помощью экспресс-методик оценки функциональной активности центральной и вегетативной нервной системы, психоэмоционального состояния, а также показателей гемодинамики. Разработанное программное обеспечение может быть использовано широким кругом пользователей без возрастных ограничений, программное обеспечение имеет возможность полной интеграции с разработанной авторами ранее информационной системой «Универсальный мониторинг экологического здоровья» www.umon.org.ua.

Введение

Для установления параметров индивидуальной чувствительности к изменчивым факторам внешней среды (экологическим, метеорологическим, космофизическим и др.) необходима длительная регулярная регистрация параметров функционального состояния человека. Однако лишь при условии накопления длительных временных рядов информативных показателей состояния человека можно эффективно прогнозировать индивидуальные особенности реагирования организма на факторы различной природы и разрабатывать адекватные мероприятия, направленные на здоровьесбережение [1].

При проектировании информационной системы (ИС) наиболее важен вопрос выбора ее функциональной, пространственной и информационной архитектуры. Функциональная архитектура характеризует состав функциональных компонентов системы, их назначение, взаимосвязи и способы взаимодействия; пространственная архитектура — размещение функциональных компонентов системы (по данному критерию системы разделяют на сосредоточенные и распределенные); информационная архитектура характеризует различные поддерживаемые в системе представления ее информационных ресурсов, их свойства и взаимосвязи [2].

Из существующих систем комплексного мониторинга состояния организма можно выделить следующие наиболее показательные решения:

1. Проект «Гелиомед» [3] — программно-аппаратный измерительный комплекс испытуемый — датчик — компьютер с использованием клиент-серверных технологий. Функциональное состояние испытуемого оценивается на основании анализа ЭКГ-сигнала в фазовом пространстве [4]. К сожалению, в рамках данной ИС отсутствует возможность реализации исследований без специфических датчиков; кроме того, получение выводов и рекомендаций не является автоматизированным процессом, что ограничивает область применения таких систем научно-исследовательскими задачами.

2. Распределенные клиент-серверные информационные системы «Мониторинг здоровья» (HealthMonitor) [5] и GMON Health Monitor [6], в которых пользователю необходимо на персональном компьютере установить специальное программное обеспечение, позволяющее проводить мониторинг

© Григорьев П.Е., Килесса Г.В., Хорсева Н.И., Овсянникова Н.М., 2012

ISSN 0452-9910. Кибернетика и вычисл. техника. 2012 Вып. 167

соматического здоровья человека, при этом хранилище может быть реализовано в пределах локальной сети организации или на одном ПК. Данные программные продукты обладают следующими недостатками: отсутствует централизованное глобальное хранилище, доступное через сеть Интернет, а также не предусмотрены функции анализа связи состояния участников-испытуемых с внешними факторами среды. Тем не менее, существуют системы, позволяющие осуществлять слежение за состоянием человека посредством сети Интернет через web-интерфейс. Примером такой технологии является сайт <http://www.monitorad.ru>, на котором возможен анализ показателей гемодинамики (артериального давления и пульса) с их последующей визуализацией и получением рекомендаций от специалистов. Несомненным преимуществом таких систем является их доступность для любого пользователя, которому в этом случае не требуется приобретать специальное оборудование (кроме тонометра).

Перечисленные выше виды информационных систем (в том числе, телемедицинских) имеют весьма ограниченный набор показателей состояния здоровья человека (соматического здоровья, в частности показатели сердечно-сосудистой системы) и не позволяют проводить комплексное мониторинговое различие показателей состояния организма либо требуют высокоспециального оборудования или программного обеспечения.

Таким образом, разработка информационно-программного обеспечения с локальным и/или централизованным хранилищем, доступного для использования любыми заинтересованными лицами и организациями, предоставляющего пользователю возможности комплексной диагностики его состояния, получения выводов и рекомендаций, актуальна для решения исследовательских и прикладных задач психологии, гигиены, медицины и экологии человека.

Структура функционирования информационно-программного обеспечения. Ранее нами была разработана и реализована распределенная клиент-серверная ИС «Универсальный мониторинг экологического здоровья» («УМОН») [2, 7, 8] (комплекс экспресс-методик оценки функциональной активности центральной и вегетативной нервной системы, системы гемодинамики, психоэмоционального состояния), включающая Web-сайт www.umon.org.ua для организации доступа к проекту, программное обеспечение для тестирования, анализа полученных данных и выдачи результатов. Разработанное ранее пользовательское программное обеспечение (ПО) предоставляло лишь функции тестирования и отправки накопленных данных на удаленный сервер. Вывод результатов и рекомендаций осуществлялся только на Web-сайте в «личном кабинете», а это предусматривало необходимость подключения к сети Интернет для полноценного использования информационной технологии, что в определенной мере ограничивало возможности ИС.

Поэтому была поставлена задача разработать новое локальное информационно-программное обеспечение для психофизиологического мониторинга и экспресс-тестирования с расширенными функциями. Структура функционирования нового ПО в рамках ИС «УМОН» представлена на рис. 1. Следует отметить, что общая структура ИС осталась прежней, однако теперь результаты тестирования, после экспорта в файл формата CSV, можно отправить не только на сайт, но и специалисту или по электронной почте или на каком-либо носителе информации.

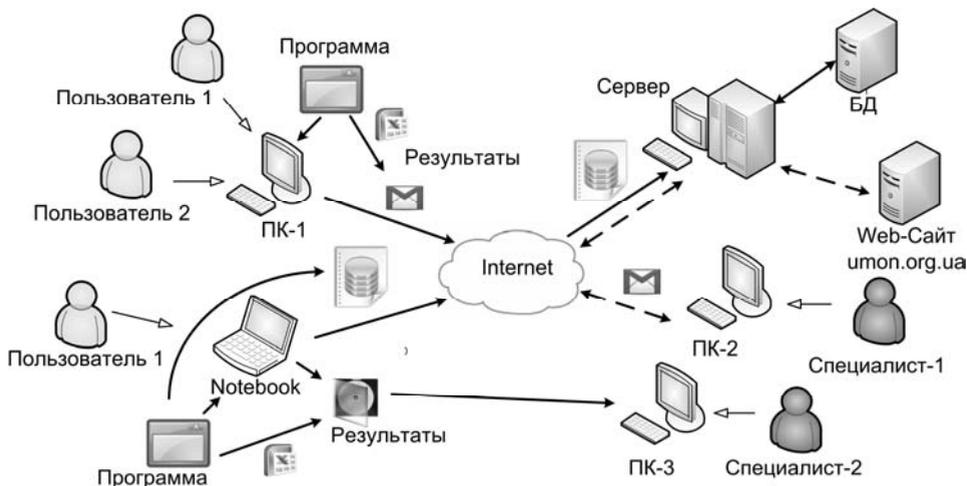


Рис. 1. Структура функционирования ПО в рамках ИС «УМОН» и механизм взаимодействия со специалистами

Программная реализация, архитектура и алгоритмы информационной системы. Информационная система реализована в программе, написанной на языке программирования C#, под программную платформу .NET Framework 2.0 для операционных систем семейства Windows. Несмотря на то что на момент разработки программы последней версией .NET Framework являлась 4.0 (поддерживается операционными системами Windows XP (Home/Professional Edition)/Windows Vista/Windows 7), мы использовали версию 2.0, позволяющую осуществлять совместимость с наиболее ранними версиями Windows, в частности с Windows 98, которая по-прежнему используется в некоторых школах и вузах.

Блок регистрации программного обеспечения. Для регистрации ПО используется уникальный идентификатор жесткого диска (не логического тома, а именно физического диска). Для получения ID доступных дисков используется бесплатная библиотека DiskId32.dll от WinSim Inc. [9], а каждому ID диска соответствует уникальный ключ продукта. Пользователю необходимо выбрать, к какому устройству будет осуществлена привязка ПО, сообщить ID диска нашему специалисту, и ввести ключ продукта, полученный от него. После ввода валидного ключа в системный реестр будет занесена соответствующая информация о регистрации ПО, и при следующем запуске программы пользователь сразу попадет в окно авторизации, где он сможет выбрать свою учетную запись или завести новую. Также в ПО предусмотрено 500 пробных запусков, при которых возможности программы никак не ограничиваются, а только процедура регистрации ПО будет инициироваться при каждом запуске, где клиенту нужно будет нажимать кнопку «Запуск», вместо ввода ключа продукта. Программа доступна для загрузки, например, с веб-страницы <http://dmfi.info/info.html>.

Идентификация пользователя и запуск программ. На первом этапе был разработан общий алгоритм функционирования ПО, блок-схема которого приведена на рис. 2. После запуска программы происходит получение имеющихся данных о программе из системного реестра и определяется, зарегистрировано ли ПО на компьютере. В случае отсутствия регистрации пользователю предлагается пройти или процедуру регистрации, которая выполняется единожды, или он может воспользоваться пробным запуском.

На следующем этапе необходимо создать учетную запись индивидуальных данных (логин, пароль для входа в программу, Ф.И.О., возраст, пол, e-mail, пароль для доступа на сайт www.umop.org.ua). После входа в программу под своим логином пользователь попадает в главное окно, опции которого позволяют выбрать необходимые тесты, запустить процесс тестирования, просмотреть полученные результаты, экспортировать накопленные данные. Кроме того, для пользователя предусмотрен выбор режимов тестирования, в частности, предусмотрен режим регистрации психофизиологических показателей для постинсультных и постинфарктных состояний, для пациентов с отклонениями в развитии, неврологическими заболеваниями и т.д.

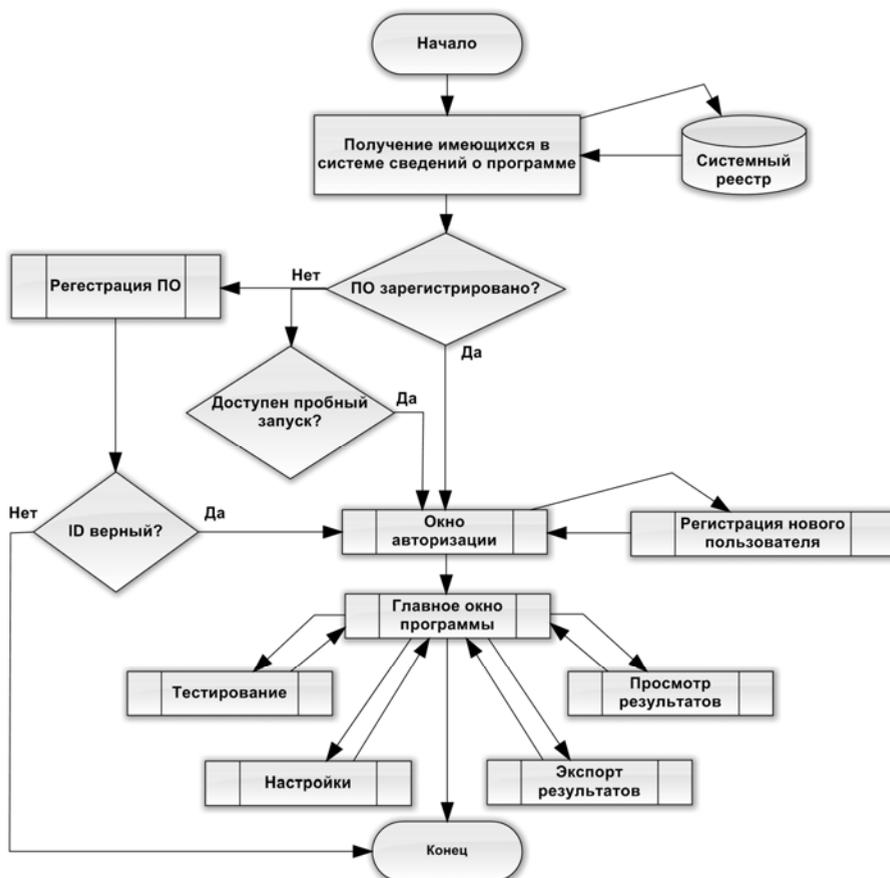


Рис. 2. Общий алгоритм функционирования ПО

Блок тестирования и просмотра результатов. На данный момент в разработанном программном обеспечении [7] доступен следующий набор методик, используемых для определения:

- слухо-моторной реакции;
- зрительно-моторной реакции;
- корректурной пробы (определение параметров устойчивости произвольного внимания, характеризующих физическую активность подкорковых структур головного мозга);
 - показателей мелкой моторики пальцев рук (хаотичный десятипальцевый теппинг, патент РФ №2314743), позволяющих определить уровень развития мелкой моторики рук, охарактеризовать функциональное

состояние и работоспособность моторных зон коры головного мозга, межполушарные взаимодействия, латеральные предпочтения;

- показателей уровня механической памяти;
- показателей уровня семантической памяти для оценки параметров смысловой памяти;
- показателей динамики психоэмоционального состояния;
- показателей гемодинамики (артериальное давление и частота пульса).

В тестах на слухо-моторную и зрительно-моторную реакции пользователю предъявляются в случайные моменты времени сигналы (соответственно звуковые или визуальные — белый квадрат на черном фоне) и ему необходимо, как можно быстрее реагировать на них, нажатием клавиши «Пробел». Программа при этом фиксирует время реакции испытуемого, и если оно $< 1,5$ секунды, то считается, что пользователь успел отреагировать на данный сигнал. При регистрации показателей учитывается и число ошибок — число пропущенных сигналов и число неверных нажатий (опережающие или лишние нажатия), а также вычисляется степень нестабильности реакции на сигнал.

В корректурной пробе пользователю предлагается таблица символов, по которой он может от символа к символу последовательно передвигаться вперед в течение двух минут, при этом буквы «К» и «Е» необходимо удалять из списка (клавишей «Пробел»). Программа автоматически регистрирует число просмотренных символов, число правильно удаленных букв, число пропущенных знаков, число ошибочно удаленных знаков. На основании этих показателей рассчитываются показатели продуктивности и точности выполнения теста.

В тесте мелкой моторики пальцев рук пользователю необходимо пройти четыре субтеста длительностью 30 секунд каждый: моторика двух рук при параллельном положении, моторика двух рук при перекрестном положении, моторика левой и правой рук по отдельности. Во всех четырех субтестах пользователю необходимо продемонстрировать картинку, на которой изображена требуемая схема расположения рук, а также обозначено, каким пальцем какую кнопку на клавиатуре нажимать. В ходе выполнения субтестов фиксируется время нажатия клавиш (относительно начала субтеста), коды клавиш, а также время их удержания в нажатом состоянии. По окончании теста для каждого субтеста рассчитывается более 30 показателей.

При выполнении теста «Механическая память» пользователю в течение 30 секунд предлагается запомнить 12 двузначных чисел. По истечении 30 секунд, испытуемый должен воспроизвести числа в том порядке, в котором он их запомнил, а не как они располагались на экране. Показателем уровня механической памяти является отношение количества правильно воспроизведенных чисел к общему количеству двузначных чисел, выраженное в процентах, — объем механической памяти.

В тесте на семантическую память испытуемому необходимо в течение 30 секунд запомнить 10 пар слов. По истечении времени пользователь должен из предложенных подсказок выбрать слова, которые он запомнил, и дополнить имеющуюся пару. Показателем смысловой памяти является коэффициент логического запоминания, определенный путем деления числа правильно воспроизведенных пар слов на количество представленных пар.

Методика для изучения динамики психоэмоционального состояния представляет собой опросный лист, где испытуемому необходимо выбирать

место на шкалах между определениями-антонимами в баллах (-3, -2, -1, 0, 1, 2, 3), которое наиболее соответствует его состоянию. По завершении теста определяются следующие показатели: самочувствие, активность, настроение, тревожность, агрессивность, степень благоприятности внешних обстоятельств (социум).

Для определения показателей гемодинамики пользователю необходимо провести три последовательных измерения артериального давления (САД и ДАД) и частоты сердечных сокращений (ЧСС), и внести данные в соответствующие поля. Полученные результаты являются основой для их дальнейшей обработки: на основании средних величин САД, ДАД и ЧСС вычисляется показатель пульсового давления и индекс Кердо.

Совокупность описанных выше методик позволяет проводить комплексную экспресс-диагностику состояния важнейших функциональных систем организма человека. По окончании последнего теста пользователь попадает в режим просмотра результатов, где отображаются результаты каждого выполненного тестового задания, а также выводы и рекомендации на основе сравнения полученных данных не только со среднegrupповыми показателями для данного возраста и пола, но и с индивидуальными нормами, полученными в результате многократного тестирования данного испытуемого.

Вывод результатов осуществляется в двух режимах: «однодневный» и «многодневный». В первом случае отображаются данные за один сеанс тестирования по выбранному тесту. В режиме «многодневный» результаты выводятся за определенный период времени, в этом случае строится динамика изменения того или иного параметра. Для визуализации полученных результатов в виде графиков используется бесплатная библиотека ZedGraph [10], написанная на языке программирования C# для рисования 2D графиков, гистограмм и круговых диаграмм. Также предусмотрена возможность формирования отчета с последующим сохранением в файл формата PDF. Для этой цели используется бесплатная библиотека iTextSharp [11] — порт Java-библиотеки iText с открытым исходным кодом для генерации PDF-документов.

Как было отмечено выше, для реализации информационно-программного обеспечения был выбран объектно-ориентированный язык программирования C#, и, как следствие, необходимо было разработать схему классов, позволяющую реализовать все выше описанные алгоритмы. На рис. 3 приведена схема классов, отвечающих за тестирование пользователей и вывод результатов.

Из всех характеристик всего комплекса тестов мы выделили общие параметры (например, уникальный идентификатор теста для доступа к данным в локальной БД), и объявили их в абстрактном базовом классе TestBase. Как видно из рис. 3, в классе TestBase объявлен также ряд абстрактных функций: *ShowUI(Form form)* — функция выводит требуемые для данного теста элементы пользовательского интерфейса (User Interface) на форму, ссылка на которую передается в параметре *form*; *Start()* — вызов данной функции начинает выполнение подпрограммы данной тестовой методики (запускаются необходимые таймеры, включаются требуемые обработчики и т.д.); *Finish()* — функция, вызываемая при завершении теста, которая останавливает подпрограмму данной тестовой методики и вызывает виртуальную функцию класса TestBase *Calculate()*; *Refresh()* — при запросе пользователя начать выполнение теста заново вызывается данная функция,

которая очищает уже имеющиеся массивы данных; *KeyDown(Keys keyValue)* и *KeyUp(Keys keyValue)* — обработчики соответственно нажатия и отпускания клавиши на клавиатуре. Как было отмечено выше, в базовом классе еще определена виртуальная функция *Calculate()*, которая вызывает функцию *_result.Calculate()*, объявленную в классе *ResultBase*, а также проверяет, верно ли была пройдена тестовая методика; также имеется виртуальная функция *Save()*, которая вызывает функцию *_result.SaveToFile()*, объявленную в *ResultBase*. Каждая методика представлена своим классом, наследуемым от *TestBase* и реализующим все его абстрактные функции.

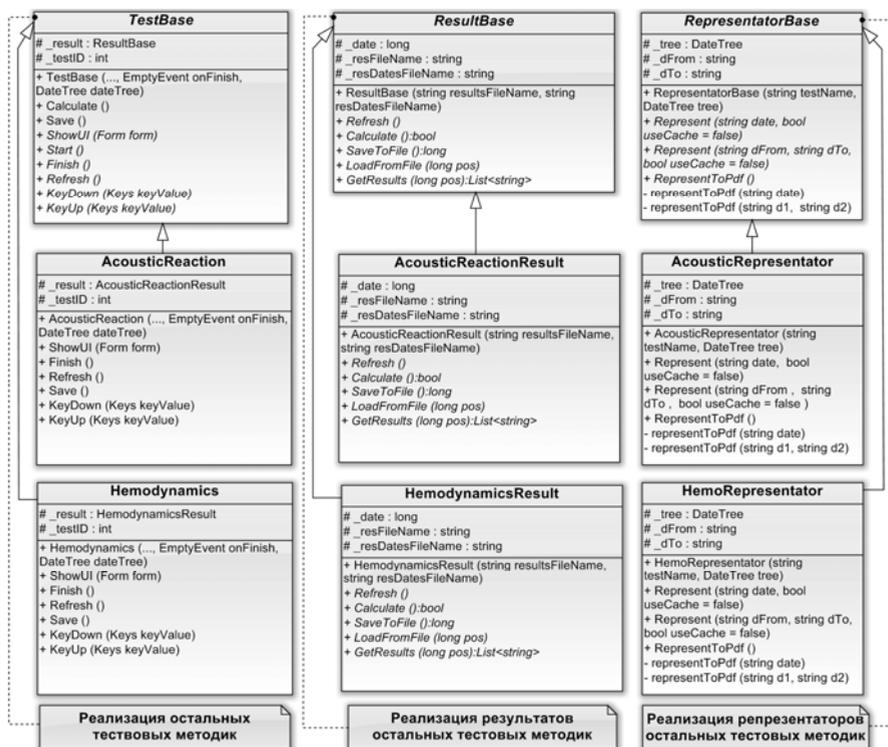


Рис. 3. Схема классов, отвечающих за блоки тестирования и вывод результатов

Аналогичным образом мы поступили в случае просмотра данных — для каждого теста есть свой репрезентатор, наследуемый от класса *RepresentatorBase*, который занимается визуализацией данных по соответствующей методике. На рис. 3 приведены наиболее важные функции класса *RepresentatorBase*, среди которых имеются абстрактные функции, реализуемые наследниками: *Represent(string date, bool useCache)*, *Represent(string dFrom, string dTo, bool useCache)*, *RepresentToPdf()*, выводящая результаты в PDF-документ, и виртуальные функции — *representToPdf (string date)* и *representToPdf (string d1, string d2)*, выводящие в PDF-файл общую информацию для всех тестов, отвечают за визуализацию данных на форме результатов соответственно в «однодневном» и «многодневном» режимах просмотра (параметр *useCache* отвечает за необходимость использования кэширования при обращении к базе данных).

Расчетные параметры по каждому тесту также представлены отдельными классами, наследуемыми от базового *TestResult*, в котором объявлены все общие характеристики (дата прохождения, имена файлов с результатами и датами прохождения), а также ряд абстрактных функций: *SaveToFile()*,

LoadFromFile(log pos), GetResults(long pos), Calculate(), Refresh(). Ключевой функцией является *Calculate()* — именно здесь происходит расчет выходных показателей по данному тесту на основе входных данных, полученных в результате выполнения тестовой методики. Функция *GetResults(long pos)* используется при экспорте данных и возвращает результаты теста в виде списка строк, где каждая строка представляет какой-то выходной показатель данного теста.

Время отклика клавиатуры. В тесте мелкой моторики пальцев рук важными параметрами, по которым строятся ритм-диаграммы, характеризующие последовательность выполненных нажатий, и вычисляются некоторые выходные показатели, являются время удержания каждой клавиши в нажатом состоянии и время между двумя последовательными нажатиями. Именно последний параметр лежит в основе регистрации такого показателя, как синкинезии (содружественные движения — произвольные мышечные сокращения, сопровождающие выполнение какого-либо активного двигательного акта), в данном случае характеризующие уровень дифференцированности движений пальцев рук. В связи с этим была выполнена оценка времени отклика на разных клавиатурах и на разных ПК. Поскольку тест предусматривает последовательное выполнение четырёх субтестов (синхронный хаотичный теппинг обеими руками в обычном положении, каждой рукой отдельно и синхронный теппинг обеими руками в перекрещенном положении), в каждом субтесте каждой рукой выполнялось 20 одновременных нажатий двух клавиш и регистрировалась разность во времени между двумя одновременными нажатиями. Результаты, полученные на разных ПК, были в целом одинаковы (по средним значениям). Как видно из табл. 1, в которой приведены результаты одного из тестов, разница между двумя одновременными нажатиями варьируется от 0 до 16 мс, т.е. в пределах 16 мс невозможно точно различить последовательность нажатий. Таким образом, нажатие двух клавиш с разницей в 20 мс или меньше, что превышает время отклика в разработанной программе, может служить валидным основанием для регистрации синкинезии.

Следует особо отметить, что использование беспроводной клавиатуры может существенно исказить не только регистрацию данных теста мелкой моторики руки, но и теста «корректирующая проба». Поэтому в инструкциях к тестам указано: «Не использовать беспроводную клавиатуру».

Структура локальной базы данных. В первой реализации локальной программы была предусмотрена возможность лишь накопления результатов: данные хранились в обычном файле и при нажатии пользователем кнопки «Отправить» последовательно считывались из него, отправлялись на сервер и удалялись. Планировалось, что в новой версии программы пользователь сможет просматривать результаты, т.е. необходимо было реализовать быстрый доступ к данным по требуемому тесту за требуемую дату. Для решения этой задачи была разработана следующая схема хранения данных: для каждого теста создается два файла. В одном — хранятся результаты тестирования по данному тесту данным пользователем, а во втором — парами записаны целые 64-битные числа, первое число — сериализация объекта *DateTime* (дата прохождения) в 64-битовое двоичное значение, второе — позиции начала результатов за эту дату в первом файле. При входе в программу под своим логином происходит инициализация базы данных в оперативной памяти компьютера. Для каждого теста считывается второй файл, и происходит создание по нему дерева, где конечными узлами

выступают позиции в файле с результатами (position на рис. 4). При запросе результатов пользователем сначала из дерева достается позиция по требуемому тесту и за требуемую дату, а затем происходит считывание результатов из первого файла, начиная с полученной позиции. На рис. 4 приведен пример структуры локальной базы данных (БД) для двух тестов.

Таблица 1

№	Моторика – ПАРАЛЛ		Моторика – ОТДЕЛЬНО		Моторика – ПЕРЕКР	
	Левая рука	Правая рука	Левая рука	Правая рука	Левая рука	Правая рука
1	15	15	15	16	16	15
2	0	15	15	0	15	16
3	16	16	15	15	15	16
4	15	16	0	15	16	16
5	16	16	15	0	15	16
6	16	16	16	15	16	15
7	16	15	0	15	16	16
8	16	16	15	15	16	16
9	0	15	0	0	16	16
10	16	16	15	0	16	16
11	16	16	16	16	16	16
12	16	16	16	0	16	15
13	16	16	16	15	16	15
14	15	16	16	0	0	15
15	15	16	16	15	16	15
16	16	0	16	15	0	16
17	15	15	0	0	16	15
18	15	16	16	16	16	16
19	15	16	16	15	15	15
20	16	15	16	0	15	0
Средние значения						
1–20	14,05	14,9	12,5	9,15	14,15	14,8

Такая структура хранения данных позволяет использовать малый объем оперативной памяти для хранения сведений о результатах и добиться быстрого извлечения данных с жесткого диска. Данная схема также обеспечивает хорошую масштабируемость, т.е. скорость доступа к данным не зависит от количества тестовых методик в базе данных, и возможность кэширования данных, при которой в оперативной памяти в конечных узлах дерева хранятся еще и результаты, а не только их позиции в файлах.

Экспорт результатов. Поскольку разработанный программный продукт является локальным, т.е. данные хранятся на компьютере пользователя, то важным элементом ПО является механизм передачи данных специалисту в

электронном виде. В программе реализована возможность экспортирования результатов тестирования за выбранный промежуток дат в файл формата CSV (текстовый формат, предназначенный для представления табличных данных). После чего пользователь либо по электронной почте, либо на физическом носителе (флэш-память, оптический диск и т.п.) может предоставить результаты специалисту. Специалистам же предоставляется расширенная версия программного продукта, в которой реализована функция визуализации данных, где в качестве источника данных выступает CSV-файл, а не локальная база данных.

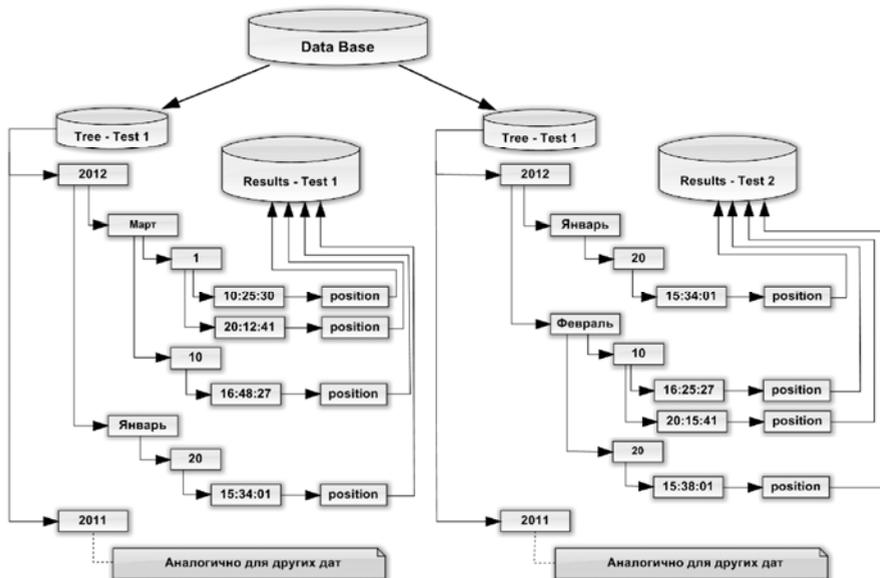


Рис. 4. Структура локальной базы данных

Как было отмечено выше, в ПО предусмотрены функции для отправления данных на сайт www.umon.org.ua, для чего разработан и реализован алгоритм передачи результатов из локальной программы на сервер (рис. 5).

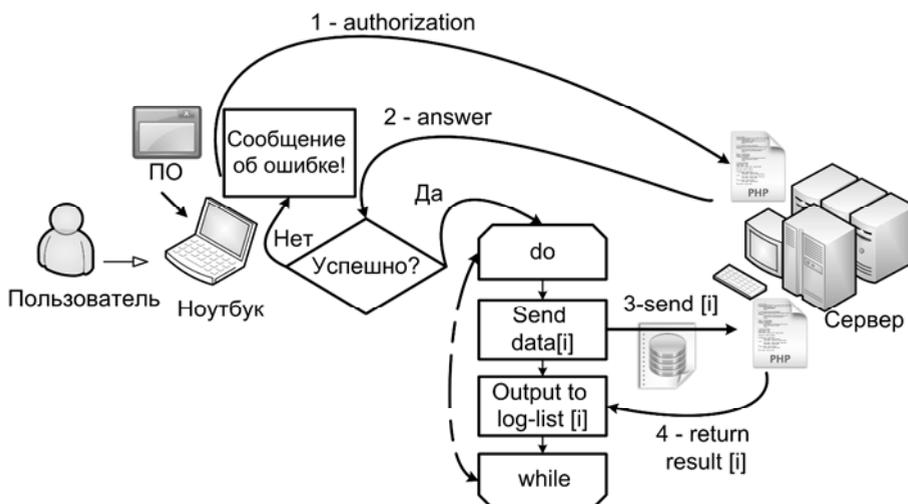


Рис. 5. Алгоритм передачи данных на сервер

В связи с тем что регистрационные данные пользователя на сайте (e-mail и пароль) могут отличаться от данных в локальной программе (login и пароль), в меню «Настройки» пользователю предлагается ввести свой e-mail и пароль, используемые им для входа на сайт. Эти данные используются на первом этапе отправки результатов — authorization. На сервере выполняется проверка, и её результат отправляется локальной программе (на рис. 5 шаг № 2 — answer). В случае отказа сервера пользователю в доступе к базе данных, в программе всплывает сообщение об ошибке. Если же проверка прошла успешно, то в ПО формируется массив, который необходимо отправить. Затем по циклу (за выбранный период времени) на каждом шаге происходит отправка [i]-го элемента сформированного массива (на рис. 5 шаг № 3 — send [i]). Сервер, получая данные, осуществляет транзакцию на добавление данных в базу данных и возвращает локальной программе её результат (на рис. 5 шаг № 4 — return result).

Выводы

1. Разработанное информационно-программное обеспечение, предназначенное для использования широким кругом пользователей без возрастных ограничений, позволяет осуществлять комплексное мониторирование психофизиологического состояния человека с помощью экспресс-методик оценки функционального состояния центральной и вегетативной нервной системы, психоэмоционального состояния, показателей гемодинамики, а также визуализировать результаты в виде удобных графиков и таблиц и формировать отчеты с последующим хранением их в формате PDF.

2. Созданный информационно-программный модуль функционирует в рамках ранее разработанной авторами информационной системы «Универсальный мониторинг экологического здоровья», предоставляя пользователю возможность отправлять результаты тестирования на сайт www.umon.org.ua или передать их в виде CSV-файла специалисту.

3. В результате тестирования испытуемого он автоматически получает выводы и рекомендации относительно психофизиологического состояния, которые формируются на основе сравнения полученных данных с нормативными значениями показателей для данного возраста и пола, а также с индивидуальными нормами, накопленными в результате многократного тестирования данного испытуемого.

1. Хорсева Н.И. Психофизиологические показатели как критерий оценки экологической безопасности места проживания // Климат, качество атмосферного воздуха и здоровье москвичей. Сб. научных трудов. — М.: АдамантЪ, 2006. — 246 с.
2. Серверная компонента информационной системы для определения зависимости состояния человека от космофизических факторов / А.В. Оленчук, Э.Э.Асанов, Г.В. Килесса и др. // Труды десятой ежегодной молодежной конф. ИБХФ РАН-вузы "Биохимическая физика". — Москва: РАН, 2010. — С. 98–100.
3. Проект "Гелиомед". — <http://www.geliomed.kiev.ua/>.
4. Файнзильберг Л.С. Информационные технологии обработки сигналов сложной формы. Теория и практика. — К.: Наукова думка, 2008. — 333 с.
5. Компьютерная программа «Мониторинг здоровья». — <http://www.healthmonitor.ru/>.
6. GMON Health Monitor // Medizin & Service GmbH. — <http://www.gmon.info/>.
7. Информационная система диагностики для определения зависимости состояния человека от действия космофизических факторов / П.Е. Григорьев, Г.В. Килесса, Н.И. Хорсева и др. // Клиническая информатика и телемедицина. — 2010. — 6, Вып. 7. — С. 83–89.

8. *Килесса Г.В., Асанов Э.Э., Григорьев П.Е., Хорсева Н.И.* Информационная технология мониторинга состояния человека для определения его зависимости от действия космофизических факторов // Труды десятой ежегодной молодежной конф. ИБХФ РАН-вузы "Биохимическая физика". — М.: РАН, 2010. — С. 94–98.
9. WinSim Inc. DiskId32 - A Win32 Application and Source Code for Reading Hard Drive Manufacturing Information.— <http://www.winsim.com/diskid32/diskid32.html>.
10. SourceForge. ZedGraph is a class library, user control, and web control for .net, written in C#, for drawing 2D Line, Bar, and Pie Charts. — <http://sourceforge.net/projects/zedgraph/>.
11. SourceForge. iText# (iTextSharp) is a port of the iText open source java library for PDF generation written entirely in C#. — <http://sourceforge.net/projects/itextsharp/>.

Крымский государственный медицинский
Университет имени С.И. Гиоргиевского, Симферополь
Таврический национальный университет
имени В.И. Вернадского, Симферополь
Институт биохимической физики имени
Н.М. Эмануэля РАН, Россия, Москва

Получено 10.05.2012