

Е.Б. Галян

## Специализированный программный модуль технологии восстановления речи; архитектура и функциональное взаимодействие его компонентов

Разработана архитектура типа *клиент–сервер* программного модуля технологии тренировки движений кисти для восстановления речи. Описано функциональное поведение программного модуля в нотации *UML*. Представлены логическая модель управления доступом различных категорий пользователей на базе ролевой модели и модель взаимодействия основных функциональных компонентов модуля в терминологии классов объектно-ориентированного программирования.

The architecture as «client–server» type of the software module of the technology for training of hand movements to help in restoring speech is developed. The functional behavior of the software module in UML notation is described. The logical model of access control of different user categories based on the role-model and a logical model of the interaction of basic functional components of the module in terms of object-oriented programming classes are presented.

Розроблено архітектуру типу *клієнт–сервер* програмного модуля технології тренування рухів кисті для відновлення мовлення. Описано функціональну поведінку програмного модуля в нотатції *UML*. Представлено логічну модель керування доступом різних категорій користувачів на базі ролевої моделі та модель взаємодії основних функціональних компонентів модуля в термінології класів об'єктно-орієнтованого програмування.

**Введение.** В предыдущих исследованиях разработаны информационно-структурная и структурно-функциональная модели информационного компонента как необходимого инструмента повышения эффективности и оперативности практического использования новой технологии тренировки движений кисти для восстановления речи [1]. Показано, что информационный компонент представляет собой программный модуль и обеспечивает функции электронного справочника о технологии, поддержки принятия решений по формированию планов реабилитационных курсов, а также базы данных пациентов.

### Постановка задачи

Следующий этап исследований направлен на проектирование программного модуля, детализацию его функционального поведения, разработку логических моделей для последующей реализации.

Цель данной статьи – разработать архитектуру программного модуля информационной технологии тренировки тонкой моторики кисти пациента для восстановления речи с учетом специфики взаимодействия его функциональных компонентов. Для этого необходимо решить ряд задач:

- провести обоснование общей архитектуры программного модуля технологии восстановления речи, особенности алгоритмической и логической реализации выполняемых модулем функций;
- разработать логические модели отдельных компонентов программного модуля в терминологии классов объектно-ориентированного программирования для последующей реализации.

### Методы исследования

Для проектирования архитектуры программного модуля использован унифицированный язык моделирования (*Unified Modeling Language – UML*), предназначенный для визуализации, спецификации, конструирования и документирования артефактов программных систем [2]. Преимущества использования *UML* для выполнения поставленных задач в сравнении с другими способами моделирования:

- объектно-ориентированность языка моделирования, вследствие чего методы описания результатов анализа и проектирования семантически близки к методам программирования на языке *C Sharp*, выбранном для реализации программного модуля информационной технологии [1];

**Ключевые слова:** тренировка движений, восстановление речи, информационная технология, программный модуль, архитектура, *UML*, взаимодействие компонентов, логическая модель.

- многомодельность *UML*, что позволяет описать программный модуль практически со всех возможных сторон, а также аспекты его поведения с разным уровнем абстрагирования и детализации, начиная с концептуального уровня вплоть до физической реализации;

- диаграммы *UML* сравнительно просты для чтения после достаточно быстрого ознакомления с его синтаксисом.

## Результаты

**Обоснование общей архитектуры программного модуля технологии восстановления речи.** Анализ структурно-функциональной модели программного модуля [1] показал, что архитектуру модуля целесообразно проектировать по типу *клиент–сервер* (рис. 1), поскольку к работе с модулем должны иметь доступ лечащие врачи, младший медицинский персонал и другие специалисты отделения с разных компьютеров. При этом обеспечивается ведение единой базы данных пациентов и стандартизация протокола ведения реабилитации. Каждый пользователь взаимодействует с программным модулем через клиентское приложение, установленное на автоматизированном рабочем месте (АРМ) и реализующее графический интерфейс пользователя (*GUI*). Посредством запросов пользователи взаимодействуют с сервером, на котором располагаются база медицинских карт пациентов и компоненты, отвечающие за бизнес-логику, реализующую функциональные возможности программного модуля. Все АРМ и сервер объединены в сеть через сетевой коммутатор. Обработка запроса пользователя с АРМ при такой конфигурации программного модуля проводится при обращении через сервер к базе данных.

Сервер отвечает на клиентские запросы напрямую, при этом используя только собственные ресурсы и не вызывает сторонние сетевые приложения для выполнения какой-либо части запроса. При необходимости провести обработку информации о пациентах, хранящейся в базе данных медицинских карт, запущенное на компьютере пользователя клиентское приложение, работающее с базой данных, формирует

запрос на языке *SQL*. Сервер базы данных принимает запрос и обрабатывает его самостоятельно, при этом на компьютер пользователя передается только результат. Файл, хранящий данные, которые служат источником для обработки, никуда не передается. Такой подход позволяет оптимизировать нагрузку на сеть.



Рис. 1. Общая архитектура программного модуля технологии тренировки движений кисти для восстановления речи типа *клиент–сервер*

**Функциональное поведение программного модуля.** Для формирования общих требований к функциональному поведению программного модуля – набору пользовательских задач, выполняемых программой, а также способов по их выполнению, – разработана диаграмма вариантов использования программного модуля в нотации *UML* (рис. 2). В соответствии с диаграммой пользователи разрабатываемого программного модуля поделены на несколько категорий. Незарегистрированный пользователь может просматривать базу знаний технологии восстановления речи и регистрироваться. После регистрации он может проходить аутентификацию, редактировать профиль, сформировать запрос на изменение своей категории. Это общие варианты использования для всех категорий пользователей.

Рассмотрим подробно категории и их взаимодействие с программным модулем.

*Медсестра* – категория, представляющая младший медицинский персонал, занимающийся непосредственно проведением тренировок движений кисти. Пользователи данной категории имеют возможность просматривать медицинские карты пациентов, параметры назначенных тренировок, базу знаний технологии тренировки движений кисти. Кроме этого, по-

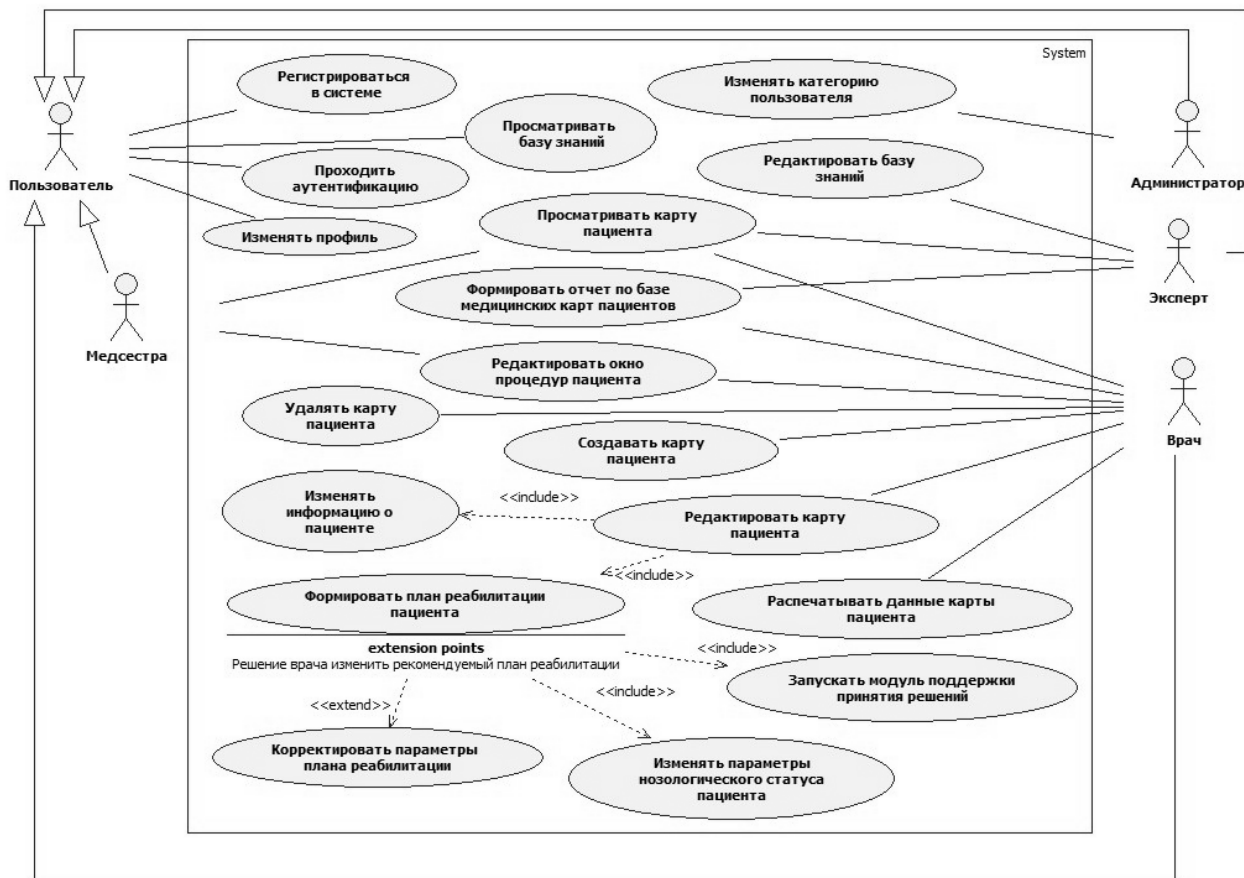


Рис. 2. Диаграмма вариантов использования программного модуля информационной технологии тренировки кисти для восстановления речи

сле выполнения тренировки медсестра фиксирует ее выполнение или записывает причины невыполнения.

*Врач* – категория медицинских специалистов, назначающих реабилитационный курс и определяющих его параметры: количество тренировок в одном курсе восстановительного лечения, набор тренируемых движений кисти и пальцев, параметры каждой тренировки. Соответственно, данная категория пользователей может создавать, редактировать и удалять медицинские карты пациентов, формировать реабилитационный курс, определять и корректировать его параметры и параметры отдельных тренировок. При невозможности присутствия медсестры, врач также может фиксировать выполнение тренировок или причины их невыполнения.

*Эксперт* – категория пользователей, имеющая доступ к формированию базы знаний по методике тренировки кисти для восстановле-

ния речи. Пользователи этой категории могут добавлять новые модули базы и редактировать существующие, проводить суммарный анализ полученных клинических результатов восстановительного лечения.

*Администратор* – пользователь, обслуживающий базу данных медицинских карт пациентов и подтверждающий права доступа другим пользователям.

Анализ диаграммы вариантов использования программного модуля показал, что некоторые варианты можно объединить в группы по выполняемым задачам и проектировать как отдельные компоненты модуля по методу компонентного проектирования [3]. Это дает возможность разрабатывать компоненты как самостоятельные единицы с завершенной функциональностью, а затем их объединять при помощи программных интерфейсов. Программный модуль технологии тренировки движений кисти для восстановления речи содержит компоненты:

- распределения прав доступа пользователей;
- базы данных медицинских карт пациентов;
- поддержки принятия решений по формированию планов реабилитационных курсов;
- электронной библиотеки по методам технологии тренировки движений кисти для восстановления речи;
- графического интерфейса пользователя (GUI).

**Логические модели компонентов и их взаимодействие в программном модуле.** Поскольку каждая категория пользователей имеет общие и уникальные права доступа к разным функциональным блокам программного модуля, разработана логическая модель распределения прав доступа в виде диаграммы классов (рис. 3). Диаграмма классов служит для представления статической структуры модели проектируемого модуля в терминологии классов объектно-ориентированного программирования и отражает различные взаимосвязи между

отдельными сущностями предметной области, описывает их внутреннюю структуру и типы отношений.

Распределение прав доступа пользователей основано на ролевой модели, при которой права доступа субъектов системы на объекты группируются с учетом специфики их применения, образуя так называемые роли [4].

Класс *User* описывает пользователя. Предполагается, что с программным модулем осуществляют работу пользователи, обладающие обязательными уникальными аутентификационными данными: именем пользователя и паролем, правами доступа соответственно категории пользователя. При этом категорию администратора может иметь только один пользователь и только он может менять категории других пользователей.

Пользователь может инициировать попытку аутентификации или зарегистрироваться с помощью методов *AuthAttempt* и *goRegistration* соответственно. Доступ к электронным справочникам пользователь имеет без авторизации

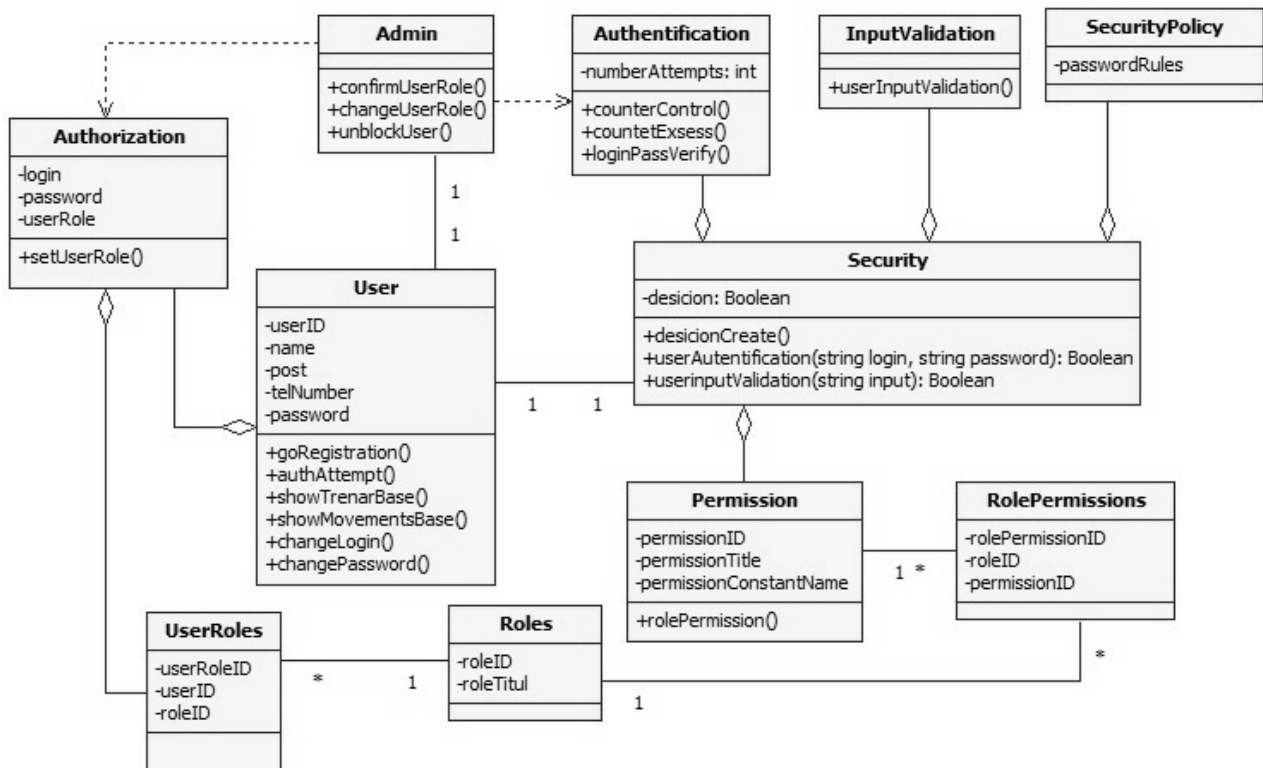


Рис. 3. Диаграмма классов распределения прав доступа пользователей программного модуля технологии тренировки движений кисти для восстановления речи

с помощью методов *startTrenarBase*, *startMovementsBase*.

**Класс Security.** Данный класс есть объединением всех реализуемых механизмов обеспечения безопасности, описание и определение которых приводится в классах *InputValidation*, *Authentication*, *Permission*. Класс *Security* содержит логическую переменную *Decision*, значение которой устанавливается методом *DecisionGenerate* на основании данных, генерируемых методами *UserAuthentication* и *UserInputValidation* в зависимости от пользовательской операции, безопасность которой проверяется. Если значение переменной *Security.Decision* – *true*, разрешается доступ к функциональной части приложения.

**Класс InputValidation.** Выполняет проверку корректности пользовательского ввода.

**Класс Authentication.** Содержащаяся в данном классе целочисленная переменная *NumOfAttempts* представляет собой счетчик попыток аутентификации пользователя и позволяет ограничить их количество. Управление счетчиком осуществляется в методе *CounterControl*. В случае превышения допустимого значения счетчика вызывается метод *CounterExcess*, блокирующий доступ к приложению. Разблокировку проводит администратор. Метод *LoginPassVerify* выполняет сравнение вводимых пользователем данных при попытке аутентификации с соответствующими сохраненными значениями.

**Класс Security Policy.** Содержит правила безопасности *PasswordRules*, согласно которым выбираются способы аутентификации, обеспечивающие различный уровень безопасности, а также ограничения на возможный пароль пользователя: минимальная длина, допустимые символы. С помощью соответствующих настроек пользователь может выбрать желаемый уровень защищенности.

**Класс Authorization.** Отвечает за авторизацию пользователя – предоставление пользователю прав на выполнение определенных действий.

**Класс Roles.** Определяет каждую категорию пользователя. Класс *UserRoles* содержит дан-

ные о категории отдельных пользователей и их правах доступа к операциям. Соответственно, классы *RolePermission* и *Permission* определяют набор прав каждой роли и непосредственно разрешенные операции.

Логическая модель взаимодействия остальных компонентов программного модуля представлена на рис. 4.

Данная логическая модель состоит из следующих сущностей:

- *Patient*. Включает всю информацию о паспортных данных пациента, его месте работы, социальном статусе;

- *Doctor*. Включает в себя информацию о врачах. Так как пациент наблюдается у одного лечащего врача и один и тот же врач может вести различных пациентов, то сущность *Doctor* образует с пациентом связь *один-ко-многим*;

- *Medsister*. Представляет пользователя с категорией «Медсестра». Данная сущность образует связь *один-к-одному* с сущностью *Procedure*, которая будет описана далее;

- *Gospitalization*. Определяет информацию о всех поступлениях в больницу конкретного пациента. Пациент с нарушениями речи может поступить в стационар на курс реабилитации несколько раз, поэтому сущность *Gospitalization* образует с сущностью *Patient* связь *один-ко-многим*;

- *NeurologicalStatus*. Несет информацию о текущем неврологическом статусе пациента, параметры которого неоднократно могут меняться и не только в течение одной госпитализации. Поэтому данная сущность образует связь *один-ко-многим* с сущностью *Gospitalization*. Так как значение параметров неврологического статуса влияет на выбор параметров реабилитационного курса в целом и отдельных его процедур [5], данная сущность образует связь *зависимость* в качестве класса-источника с сущностями *RehabilitationCourse* и *Procedure*;

- *Diagnosis*. Отвечает за представление информации о диагнозе пациента. Так как при госпитализации фиксируется текущий диагноз больного, сущность *Diagnosis* образует связь *один-к-одному* с сущностью *Gospitalization*;

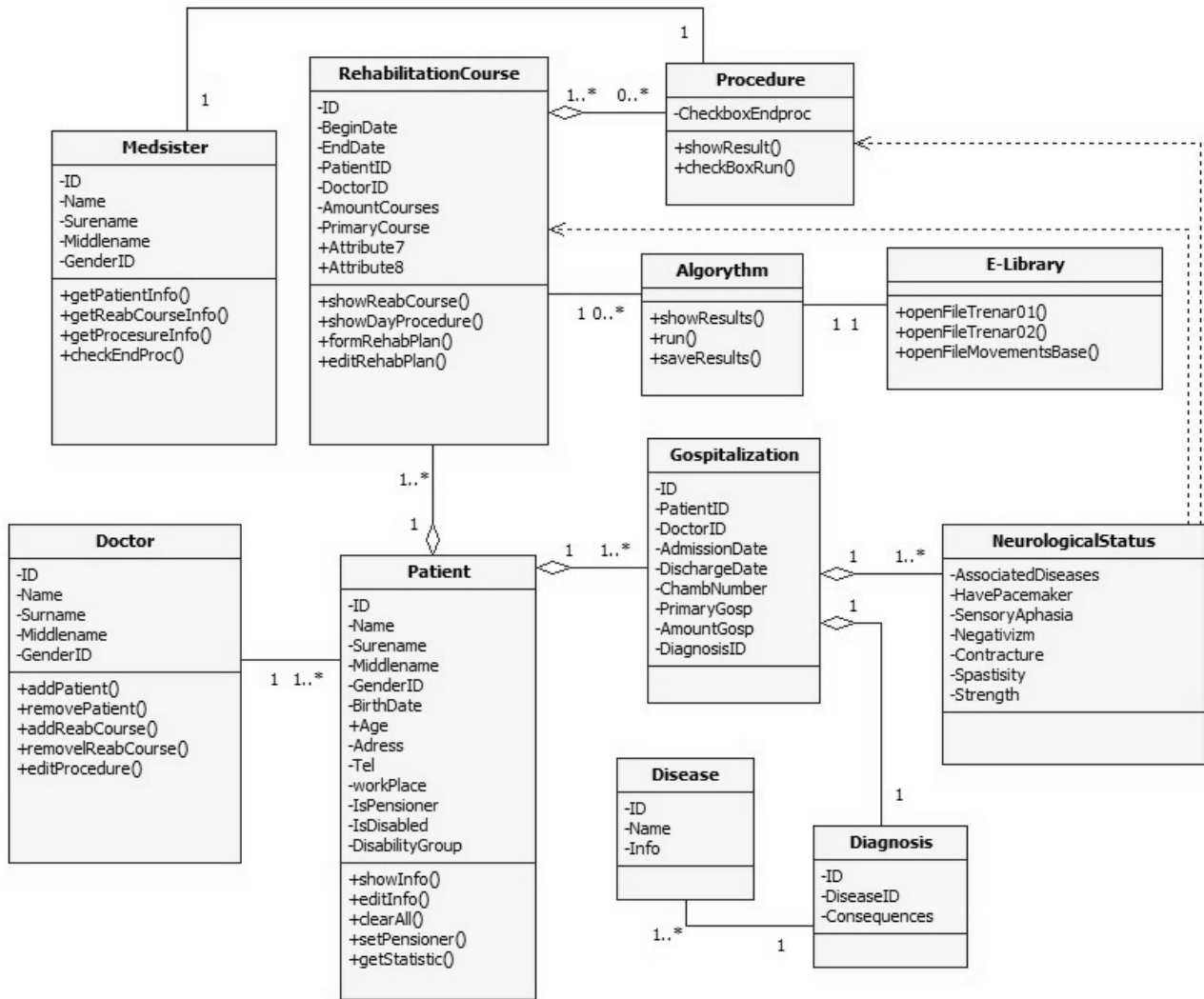


Рис. 4. Логическая модель взаимодействия компонентов программного модуля. Диаграмма классов

- *Disease*. Данный класс содержит информация о заболеваниях, которые либо могут выступать причиной нарушений речи, например, острое нарушение мозгового кровообращения, либо – сопутствующие, например, дерматит. Так как при постановке диагноза может быть отмечено несколько заболеваний, сущность *Diagnosis* имеет отношение к сущности *Disease* *один–ко–многим*;

- *RehabilitationCourse*. Описывает параметры реабилитационного курса, назначенного пациенту в стационаре.

- *Algorithm*. Отвечает за запуск и выведение результатов алгоритма по выбору параметров реабилитационного курса целенаправленной тренировки движений кисти для восстановления речи, представленного в [1]. При этом ал-

горитм может не запускаться вообще или запускаться несколько раз в течение одного курса реабилитации в связи со сменой неврологического статуса пациента. Следовательно, сущность *Algorithm* образует связь *ноль–ко–многим* с сущностью *RehabilitationCourse*;

- *Procedure*. Содержит информацию об одной процедуре тренировки движений кисти. Данная сущность связана отношением *один–ко–многим* с сущностью, отвечающей за реабилитационный курс, а также с сущностью *Medsister* отношением *один–к–одному*, так как процедуру отпускает одна медсестра;

- *E-Library*. Содержит информацию о доступе к файлам электронной библиотеки, содержащей инструкции по работе с техническими подсистемами технологии тренировки движений для

восстановления речи, справочник по методикам тренировки. В результате работы алгоритма по выбору параметров реабилитационного курса пользователь может обратиться к базе знаний для просмотра электронных инструкций по эксплуатации аппаратов для электростимуляции, справочнику по методикам наложения электродов для получения требуемых движений. Данная сущность образует связь *один-к-одному* с сущностью алгоритма.

Взаимодействие пользовательских классов и классов, реализующих функционал программного модуля, происходит через интерфейс *GUI*. Такой подход основан на паттерне проектирования *MVVM (Model-View-ViewModel)*, предложенного Джоном Госсманом (*John Gossman*) в 2005 г. как модификации паттерна *Presentation Model* [6]. Данный паттерн предполагает разделение компонентов программного модуля, обеспечивающих его функциональность и визуального представления. Пример медицинской карты пациента с нарушениями речи показан на рис. 5.

Клиническая карта пациента

КЛИНИЧЕСКАЯ КАРТА ПАЦИЕНТА №: \_\_\_\_\_

Сведения о пациенте

Фамилия: \_\_\_\_\_ Домашний адрес: \_\_\_\_\_  
 Имя: \_\_\_\_\_ Телефон: \_\_\_\_\_  
 Отчество: \_\_\_\_\_ Место работы, занимаемая должность: \_\_\_\_\_  
 Пол: \_\_\_\_\_ Возраст: \_\_\_\_\_  
 Дата рождения: 15 октября 2014 г.  Пенсионер  Инвалид Группа: \_\_\_\_\_

Сведения о восстановительном лечении моторной афазии

| № | Госпитализация | Выписка | Лечащий врач | Палата |
|---|----------------|---------|--------------|--------|
| * |                |         |              |        |

Общие сведения | Неврологический статус | Реабилитационный курс | Процедура

При поступлении

|                                                                                    |                                                                                    |                                                                                    |
|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| 1. Сознание                                                                        | При выписке                                                                        | 1. Сознание                                                                        |
| 2. Контакт с пациентом                                                             | 2. Контакт с пациентом                                                             | 2. Контакт с пациентом                                                             |
| 3. Эмоциональная сфера                                                             | 3. Эмоциональная сфера                                                             | 3. Эмоциональная сфера                                                             |
| 4. Память                                                                          | 4. Память                                                                          | 4. Память                                                                          |
| 5. Речь                                                                            | 5. Речь                                                                            | 5. Речь                                                                            |
| <input type="checkbox"/> моторная афазия <input type="checkbox"/> сенсорная афазия | <input type="checkbox"/> моторная афазия <input type="checkbox"/> сенсорная афазия | <input type="checkbox"/> моторная афазия <input type="checkbox"/> сенсорная афазия |
| Степень: _____                                                                     | Степень: _____                                                                     | Степень: _____                                                                     |
| 6. Сила в руке, баллы                                                              | 6. Сила в руке, баллы                                                              | 6. Сила в руке, баллы                                                              |
| 7. Тонус                                                                           | 7. Тонус                                                                           | 7. Тонус                                                                           |
| 8. Комментарий                                                                     | 8. Комментарий                                                                     | 8. Комментарий                                                                     |

Рис. 5. Медицинская карта пациента с нарушениями речи

Карта содержит основные сведения о пациенте и его госпитализациях. Сведения о каждой из них, неврологическом статусе пациента, параметрах реабилитационного курса и отдельных процедур представлены в виде отдельных вкладок.

**Заключение.** Обоснованием проектирования программного модуля технологии тренировки движений кисти для

восстановления речи по типу *клиент-сервер* считается то, что такая архитектура позволяет осуществлять одновременный совместный доступ пользователей программного модуля к базе данных медицинских карт пациентов, обеспечивает уменьшение нагрузки на сеть за счет обработки запросов непосредственно на сервере, делает возможным масштабирование системы, реализующей программный модуль, предполагает невысокие финансовые затраты на покупку оборудования для ее реализации.

Применение метода компонентного проектирования программного модуля позволило при разработке отдельных компонентов модуля использовать типовые компоненты уже разработанных интеллектуальных систем, специализируя их под задачи данного исследования.

Разработанная логическая модель компонента распределения прав доступа пользователей программного модуля с использованием ролевой модели управления доступом позволяет решить такие задачи, как аутентификация и авторизация пользователей, валидация ввода, обработка исключений, что обеспечивает безопасную конфигурацию программного модуля от несанкционированного доступа.

Разработка логической модели взаимодействия отдельных компонентов программного модуля в терминологии классов объектно-ориентированного программирования позволила выделить основные сущности предметной области модуля (классы), их атрибуты и выполняемые операции, определить логику их взаимосвязей для последующей их реализации в виде классов на языке *C Sharp*.

1. *Галян Е.Б.* Информационный компонент технологии тренировки движений кисти для восстановления речи // Биомедицинская инженерия и электроника. – 2014. – № 2. – [www.es.rae.ru/biofbe/199-958](http://www.es.rae.ru/biofbe/199-958) (дата обращения: 18.08.2014).
2. *Буч Г., Рамбо Д., Якобсон И.* Введение в *UML* от создателей языка. – М.: ДМК Пресс, 2010. – 496 с.
3. *Соммервилл И.* Инженерия программного обеспечения. – М.: Вильямс, 2002. – 624 с.
4. *Role-Based Access Control Models / R. Sandhu, E. Coync, H. Feinstein et al.* // *IEEE Comp. (IEEE Press) Aug. 1996. –29 (2). – P. 38-47.*
5. *Вовк М.И., Галян Е.Б., Подпригора Е.Н.* Информационная технология управления движениями кисти при восстановлении моторного компонента речи // Кибернетика и выч.техн. – 2014. – **175**. – С. 20-30.
6. *Gossman J.* Introduction to Model/View/ViewModel pattern for building WPF apps. – <http://blogs.msdn.com/b/johngossman/archive/2005/10/08/478683.aspx> (дата обращения: 18.09.2014).

Поступила 20.10.2014  
 Тел. для справок: +38 044 503-9560, 234-6428 (Киев)  
 E-mail: [galevbor@mail.ru](mailto:galevbor@mail.ru)  
 © Е.Б. Галян, 2014