

РАЗРАБОТКА БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДИАГНОСТИКИ АТОПИЧЕСКОГО ДЕРМАТИТА

УДК 004.654:616.5

ПАНФЕРОВА Ирина Юрьевна

к.т.н., доцент кафедры информационных управляющих систем
Харьковского национального университета радиоэлектроники.

Научные интересы: проектирование баз данных, информационные системы.

ВЫСОЦКАЯ Елена Владимировна

к.т.н., профессор кафедры биомедицинской инженерии
Харьковского национального университета радиоэлектроники.

Научные интересы: медицинские информационные системы.

e-mail: diagnost@kture.kharkov.ua

ПОРВАН Андрей Павлович

к.т.н, с.н.с. кафедры биомедицинской инженерии
Харьковского национального университета радиоэлектроники.

Научные интересы: информационные технологии в медицине, численные методы оптимизации.

e-mail: borman_d@mail.ru

БЕСПАЛОВ Юрий Гаврилович

с.н.с. лаборатории моделирования адаптационных механизмов
Харьковского национального университета им. Каразина.

Научные интересы: моделирование адаптационных механизмов.

ТРУБИЦЫН Алексей Алексеевич

инженер кафедры Проектирования и эксплуатации электронных аппаратов
Харьковского национального университета радиоэлектроники.

Научные интересы: математическое моделирование в медицине.

e-mail: altr999@mail.ru

ВВЕДЕНИЕ

Атопический дерматит (АД) является одной из основных проблем детской дерматологии. За последнее время в мире, по данным разных авторов, АД страдают от 3 до 10 детей на каждую тысячу населения. Для Украины эта цифра еще больше (15 чел. на 1000 чел. населения).

Предупреждение развития и прогнозирование течения АД – это один из актуальных вопросов современной медицины, который связан с неуклонным ростом распространенности данного заболевания, склонностью к хроническому рецидивирующему течению, развитию осложнений. При этом информация, которой оперирует врач во время наблюдения и лечения пациента с АД (основные регистрационные данные анамнеза жизни больного и данные параклинического обследо-

дования, результаты клинического, биохимического и иммунологического анализов крови и т.д.) часто носит описательный характер. Учет и использование большого объема такой информации, а также необходимость ее структурирования и оптимизации доступа к ней, побуждает специалистов к разработке специализированных средств хранения информации [1, 2].

Следовательно, разработка базы данных (БД) информационной системы диагностики АД, позволяющей хранить и обрабатывать большие объемы указанной выше разнородной информации, представляет большой практический интерес.

АНАЛИЗ ТЕХНОЛОГИЙ РАЗРАБОТКИ

В медицинской практике для определения дерматопатологии используются такие медицинские информационные системы (МИС) как: «Аллергология» (Россия), «Дерматология, венерология» (Россия), «Анализ заболеваемости венерическими и кожными заразными заболеваниями» (Удмуртия) и др.

Все вышеперечисленные системы построенные на основе БД, организация которых основана как на работе реляционных моделей данных, так и объектно-ориентированных. Общим недостатком указанных информационных систем является отсутствие возможности диагностирования АД, а также использования локальных БД в качестве своего ядра [3].

Выбор системы управления базами данных (СУБД) при разработке информационной системы (ИС) диагностики АД, является одной из важнейших задач на этапе проектирования. В процессе построения схемы БД решаются задачи производительности, скорости реакции и масштабируемости будущей МИС. Схема данных МИС определяет способ и особенности хранения данных, отвечает за обработку данных, накладывает ограничения на функциональность в силу специфики архитектуры системы. На сегодняшний день, как правило, при проектировании схемы БД ИС используют два основных подхода – реляционный и объектно-ориентированный.

Объектно-ориентированный подход к моделированию систем и их составляющих рассматривает проектируемые классы, описывающие сразу все множество близких по своим свойствам объектов, обладающих внутренней структурой и подведением. Имея такое обобщенное описание, можно выбирать конкретный

элемент этого множества, необходимый для моделирования, создавая экземпляр класса и наделяя его конкретными значениями параметров. Разрабатываемое приложение сначала моделируется с помощью специального объектно-ориентированного языка моделирования, затем превращается в программный комплекс. Наиболее известным является унифицированный язык моделирования - UML. По мере разработки структуры классов и объектов с использованием UML моделируются данные, удовлетворяющие потребностям будущей системы. Обладая функцией наследования объектно-ориентированный подход дает возможность структурировать данные и распределить функции управления для разных классов объектов, что упрощает весь дальнейший процесс проектирования всей системы [4].

Классические реляционные системы управления базами данных, несмотря на их большую распространенность, имеют ряд ограничений. Их применение в информационных системах, основанных на знаниях, часто является весьма затруднительным. Это, прежде всего, связано с тем, что в основе реляционной модели данных лежит их табличное представление, что является неэффективным при обработке больших объемов данных, т.к. необходимо разделять данные на несколько связанных между собой таблиц, которые перед использованием объединяются. Постоянные операции по разделению и объединению данных снижают общую производительность БД. Другим серьезным недостатком реляционных баз данных является проблема взаимодействия реляционных баз данных с объектными платформами для разработки интерфейса, например JAVA, или ++, что приводит к необходимости описания «мэппинга» – соответствия данных в объектных приложениях таблицам в реляционных базах данных. При этом не только сокращается производительность, но и усложняется разработка и сопровождение приложений. Кроме того, благодаря тому, что методы обработки данных находятся в непосредственной близости к хранилищу данных, сокращается объем данных, которыми обмениваются прикладная программа и СУБД, что приводит к выигрышу в производительности [5].

В процессе разработки схемы данных МИС предполагается использование структурного подхода. Сущность структурного подхода к разработке МИС заключается в ее декомпозиции (разбиении) на автоматизи-

руемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи, при этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны [6].

ЦЕЛЬ СТАТЬИ

Целью работы является разработка БД МИС диагностики АД у детей, которая позволяет хранить информацию о пациенте в едином структурированном виде и оптимизировать доступ к данным на различных этапах диагностики и лечения со стороны лечащего врача.

ОСНОВНОЙ МАТЕРИАЛ

Разработанная нами БД МИС диагностики АД у детей включает в себя 7 связанных между собой классов: *Chelovek*, *Doctor*, *Patient*, *Adres*, *Analizi_p*, *Anamnez_jizni*, *Poseshenie*. Между классами *Chelovek* и *Doctor*, *Chelovek* и *Patient* установлены отношения типа обобщение; между классами *Patient* и *Adres*, *Patient* и *Anamnez_jizni*, *Poseshenie* и *Analizi_p* установлены отношения агрегации; между классами *Patient* и *Poseshenie*, *Doctor* и *Poseshenie* установлены связи типа ассоциации. Разработка структуры БД в виде UML-диаграммы классов проводилась с использованием Case-средства Rational Rose, схема которой представлена на рис. 1.

Класс *Chelovek* является родительским классом, в котором указывается основная информация по которой можно идентифицировать человека. Он включает в себя атрибуты: *ID_Chelovek* типа *Integer* - код записи; *name* типа *String* - имя; *familija* типа *String* - фамилия; *otchestvo* типа *String* - отчество. Также класс *Chelovek* содержит функции модификации данных *dobavit_zapis*, *udalit_zapis*, *obnovit_zapis*.

Класс *Doctor* содержит информацию о лечащем враче, включает в себя атрибуты: *ID_Doctor* типа *Integer* - код записи; *Dolgnost* типа *String* - должность; *Kategorija* типа *String* - категория врача; *Kabinet* типа *Integer* - номер рабочего кабинета. Также класс *Doctor* содержит функции: статистики общего числа больных, которых ведет данный врач - *kol_bolnich*; больных с определенной формой атопического дерматита - *diagnozi*; периода наблюдения больного на момент последнего обращения - *dlitelnost_nabludenija*.

Класс *Patient* содержит основные регистрационные данные о пациенте и включает атрибуты: *ID_Patient* типа *Integer* - номер регистрационной карточки пациента; *Pol* типа *String* - пол пациента; *Data_rogdenija* типа *Data* - дата рождения; *Adres* типа *Adres* - атрибут отношения агрегации между классами *Patient* и *Adres*; *Anamnez* типа *Anamnez_jizni* - атрибут отношения агрегации между классами *Patient* и *Anamnez_jizni*.

Класс *Adres*, содержит информацию о месте проживания пациента, включает в себя атрибуты: *ID_Adres* типа *Integer* - код записи; *gorod/oblast* типа *String* - город(область) проживания; *ulitsa* типа *String* - улица; *kontaktn_telefon* типа *Integer* - контактный телефон. Класс *Adres* содержит функции модификации данных *dobavit_zapis*, *udalit_zapis*, *obnovit_zapis*.

Класс *Poseshenie*, содержит данные о дате и времени посещения пациентом лечащего врача, включает в себя атрибуты: *ID_Poseshenie* типа *Integer* - код посещения; *Diagnoz_Class* типа *String* - класс диагноза по МКБ 10; *Diagnoz_PClass* типа *String* - подкласс диагноза и симптома согласно МКБ 10; *Diagnoz_ICD* типа *String* - диагноз или симптом по МКБ 10; *Date_of_visit* типа *Integer* - дата визита; *Time_of_visit* типа *Integer* - время визита; *Pat* типа *Patient* - характеризует идентификацию относительно функций и методов БД; *tjagest_techenija_AD* типа *String* - тяжесть течения АД; *rasprostr* типа *String* - распространенность заболевания; *data_nachala_obostrenija* типа *Integer* - дата начала обострения; *kol_obostr_v_god* типа *Integer* - количество обострений в год; *L_zud* типа *Integer* - наличие зуда; *L_eritema* типа *Integer* - наличие эритемы на коже пациента; *L_papula* типа *Integer* - наличие папул на коже пациента; *L_korki* типа *Integer* - наличие корок на коже пациента; *L_ekskoriacii* типа *Integer* - наличие эксфолиаций на коже пациента; *L_son* типа *Integer* - степень нарушения сна; *L_lihenisatsija* типа *Integer* - наличие лихенизации на коже пациента; *L_summa_morf_el_sipi* типа *Integer* - сумма морфологических элементов сыпи кожных покровов; *L_suhost* типа *Integer* - наличие сухости кожных покровов; *Analizi_p* типа *Analizi_p* - связывает классы *Poseshenie* и *Analizi_p*. Класс *Poseshenie* содержит функции модификации данных *dobavit_zapis*, *udalit_zapis*, *obnovit_zapis*. Также класс *Poseshenie* содержит функцию подсчета

общего числа, обратившихся в течении года
chislo_visitov_za_god.

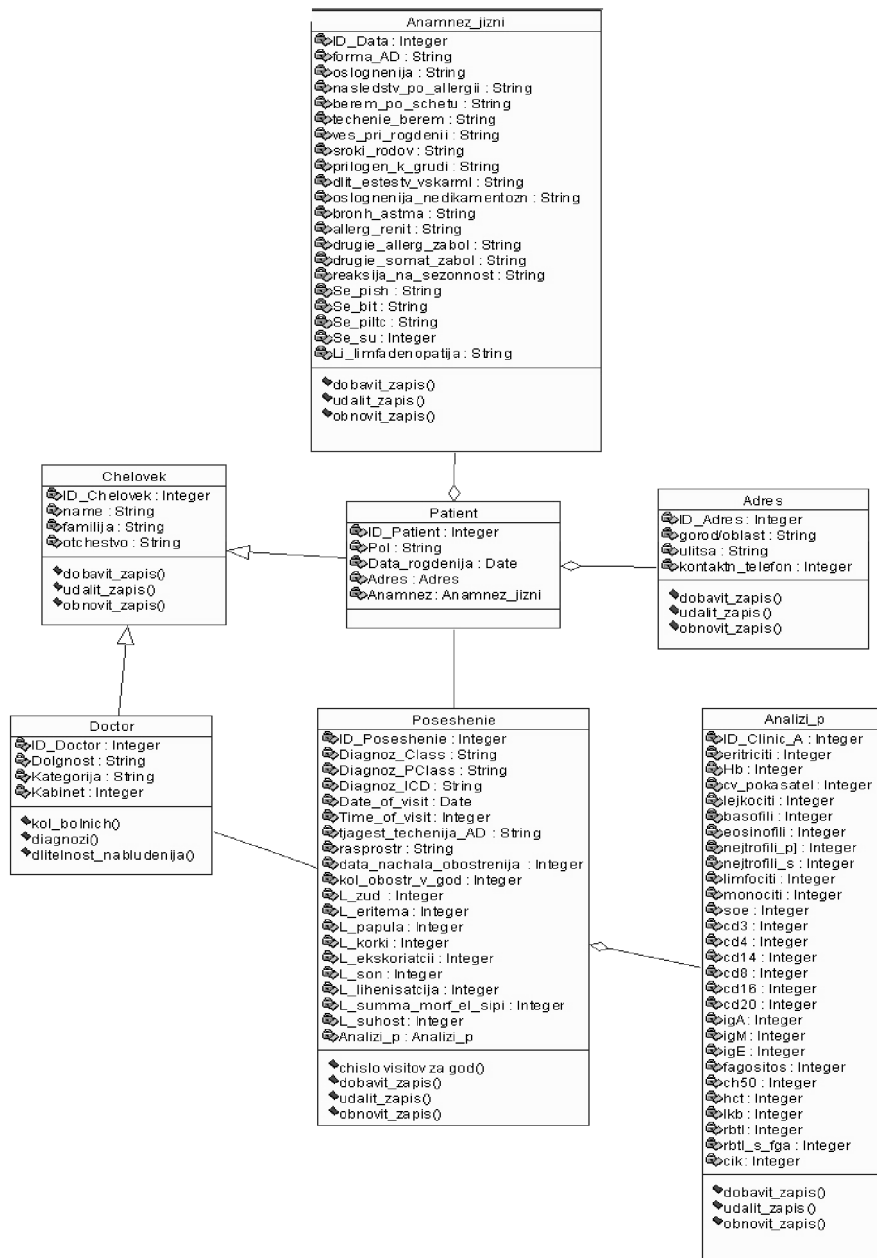


Рисунок 1 – UML-диаграмма классов объектно-ориентированной БД диагностики атопического дерматита

Класс Anamnes_jizni содержит данные, описывающие общее состояние пациента на момент поступления, включает в себя следующие атрибуты: ID_Data типа Integer - идентификатор класса Anamnes_jizni; forma AD типа String – форма АД; oslognenija типа String – наличие у больного осложнений после заболеваний; nasledstv_po_allergij типа String – наследственность связанная с наличием других аллергических заболева-

ний; berem_po_schetu типа Integer – беременность, по счету; techenie_berem типа String – характер течения беременности; ves_pri_rozdenii типа Integer – вес при рождении; sroki_rodov типа Integer – сроки родов; proligen_k_grudi типа Integer – срок, в течение которого больной был приложен к груди; dlit_estestv_vskarmli типа Integer – длительность естественного вскармливания; oslognenija_privivki типа String - наличие у боль-

ного осложнений после прививок; `oslognenija_medikamenti` типа `String` - наличие у больного аллергических реакций на медикаменты; `bronch_astma` типа `String` - наличие у больного бронхиальной астмы; `allerg_renit` типа `String` - наличие у больного аллергического ренита; `drugie_allerg_zabol` типа `String` - наличие у больного аллергических заболеваний; `somat_zabol` типа `String` - наличие у больного астматических заболеваний; `reaktsija_na_sezonnost` типа `String` - реакция больного на сезонность; `Se_pish` типа `String` - пищевая сенсидизация; `Se_bit` типа `String` - бытовая сенсидизация; `Se_piltc` типа `String` - пыльцевая сенсидизация; `Se_su` типа `Integer` - субъективные ощущения индивидуальной непереносимости; `Li_limfadenopatija` типа `String` - информация о состоянии лимфатических узлов пациента. Класс `Anamnes_jizni` содержит функции модификации данных `dobavit_zapis`, `udalit_zapis`, `obnovit_zapis`.

Класс `Analizi_p` содержит результаты анализов пациента, включает в себя атрибуты: `ID_Clinic_A` типа `Integer` - идентификатор класса `Analizi_p`; `eritrociti` типа `Integer` - количество эритроцитов; `Hb` типа `Integer` - уровень гемоглобина; `sv_pokasatel` типа `Integer` - цветной показатель; `lejkcociti` типа `Integer` - количество лейкоцитов; `basofili` типа `Integer` - количество базофилов; `esinofili` типа `Integer` - количество эозинофилов; `nejtrociti_p` типа `Integer` - количество нейтрофилов палочкоядерных; `nejtrociti_s` типа `Integer` - количество нейтрофилов сегментоядерных; `limfociti` типа `Integer` - количество лимфоцитов; `monociti` типа `Integer` - количество моноцитов; `SOE` типа `Integer` - скорость оседания эритроцитов (`СОЭ`) - неспецифический индикатор пато-

логического состояния организма; `CD3` типа `Integer` - общее количество Т-лимфоцитов; `CD4` типа `Integer` - количество Т-хелперов; `CD8` типа `Integer` - количество цитотоксических супрессоров; `CD16` типа `Integer` - количество NK-клеток; `CD20` типа `Integer` - общее количество В-лимфоцитов; `IgA` типа `Integer` - концентрация сывороточных иммуноглобулинов класса А; `IgM` типа `Integer` - концентрация сывороточных иммуноглобулинов класса М; `IgE` типа `Integer` - концентрация сывороточных иммуноглобулинов класса Е; `fagocitos` типа `Integer` - количество фагоцитов; `Ch50` типа `Integer` - уровень гемолитической активности классического комплемента; `hct` типа `Integer` - показатель гематокрита; `Ikb` типа `Integer` - содержание лизосомального катионного белка; `rbtn` типа `Integer` - реакция бластной трансформации лимфоцитов; `rbtn_s_fga` типа `Integer` - реакция бластной трансформации фагоцитов; `sik` типа `Integer` - показатель циркулирующих иммунных комплексов. Класс `Analizi_p` содержит функции модификации данных `dobavit_zapis`, `udalit_zapis`, `obnovit_zapis`.

ВЫВОДЫ

Таким образом, разработанная БД позволяет хранить информацию в едином структурированном виде, оптимизировать доступ к данным на различных этапах диагностики и лечения АД со стороны лечащего врача, а также уменьшить бумажный документооборот и сократить общее время обработки данных во время проведения диагностических и профилактических процедур у детей с АД и, при своевременном врачебном вмешательстве, предотвращать развитие патологии.

ЛИТЕРАТУРА:

1. Шуленіна О.В. Клініко-епідеміологічні аспекти atopічного дерматиту в дорослих хворих в екологічно несприятливому промисловому регіоні [Текст] /О.В. Шуленіна //Питання експериментальної та клінічної медицини. – 2009. – Вип.13, Т.1 – С.71-75.
2. Проценко Т.В. Клініко-епідеміологічні особливості atopічного дерматиту у хворих раннього дитячого віку [Текст] /Т.В. Проценко, І.Є. Мілус, О.В. Шуленіна //Дерматологія та венерологія. – 2007. – №2 (36). – С.29-32.
3. Гусев А.В. Обзор медицинских информационных систем на отечественном рынке в 2005 году [Текст] /А.В. Гусев, Ф.А. Романов, И.П. Дуданов //Мед. академ. ж. – 2005. – Т.5, №3. – С.72-84.
4. Роберт Дж. Мюллер. База данных и UML. Проектирование [Текст] /Роберт Дж. Мюллер. перевод с англ. – Landon, NY, М.: Издательство «ЛОРИ», 2007. – 420 с.
5. Алишов Н.И. Концепция и архитектурные особенности реализации объектно-ориентированной среды разработки приложений [Текст] /Н.И. Алишов, Н.А. Назаров //Комп'ютерні засоби, мережі та системи. – 2002. – №1. – С.112-118.
6. Соколова Н.А. Проектирование базы данных информационной системы прогнозирования развития нейробластомы на основе объектно-ориентированного подхода [Текст] /Н.А. Соколова, В.Э. Орел, А.А. Селезнева, А.В. Гусынин //Вестник ХНТУ. Серия «Информационные технологии» – 2012. – №1 (44). – С.86-90.