

Кроме того, в системе реализована функция отслеживания пациентов по дате посещения, которая предоставляет информацию и том, кто должен прийти на прием в определенный день и кто пропустил день своего приема.

Модуль работы с пациентами, в зависимости от введенных данных, определяет корректные для данного возраста и пола опросники и предлагает их для заполнения.

Так же реализована функция просмотра и изменения уже имеющихся данных в опросниках. Так как, анамнестические данные постоянно обновляются, ввиду улучшения или ухудшения состояния пациента, то имеется функция добавления новой записи о пациенте, для отслеживания динамики лечения.

Также системе необходим контроль за последующими обращениями пациентов. Поэтому, создана функция подсчета даты следующего обращения пациента, благодаря которой легко отслеживать, в какой день пациенту следует обратиться на прием в следующий раз.

Еще одной важной функцией, реализованной в системе можно назвать функцию автоматического формирования заключения. В зависимости от введенной информации о пациенте происходит подсчет баллов по каждому критерию, в результате которого система предоставляет результирующее значение и предлагает наиболее подходящий вариант развития событий для пациента (например, необходимость обратиться к тому или иному врачу).

В рамках оптимизации кода были реализованы специализированные классы и библиотека функций:

- класс SQL-запросов к базе данных, основанный на технологии PHP Data Object;
- класс работы с csv-файлами;
- библиотека служебных функций (динамическое формирование интерфейсных форм, интерфейсные функции, алгоритмические функции расчета заключений).

В силу того, что разрабатываемая система предназначена для хранения медицинских данных, предназначенных в том числе для дальнейшего анализа, то в ней были реализованы следующие особенности:

- шифрование данных пользователей;
- проверка на корректность ввода данных;
- защита от повторного введения информации;
- защита от SQL-инъекций.

Таким образом, разработанная система должна позволить обеспечить снижение уровня «запущенных» заболеваний населения, без привлечения специализированного и высокотехнологичного медицинского оборудования. Благодаря своей структуре, в будущем система может получить поддержку методов искусственного интеллекта, а также математического анализа [4].

Библиографический список

1. Цхай А.А., Рыков Д.А., Сибиряков А.В., Шайдуров А.А. Информационно-моделирующая система мониторинга деятельности сельхозпроизводителей региона // Известия Алтайского государственного университета. – 2011. – №1/2. – С. 126–130.
2. Гарбуз С.С., Шайдуров А.А. WEB-интерфейс сбора и распределенной обработки данных для нейросетевого моделирования // Многоядерные процессоры, параллельное программирование, ПЛИС, системы обработки сигналов. – 2015. – №5. – С. 156–158.
3. Шайдуров А.А., Зацепин П.М. // Информационно-поисковые системы для INTRANET-сетей Известия Алтайского государственного университета. – 2000. – №1. – С. 061–062.
4. Шайдуров А.А., Шатохин А.С., Пиянзин А.И. и др. Нейросетевой диагностический комплекс с элементами автоматической модификации // Нейроинформатика и ее приложения: материалы XII всероссийского семинара. – Красноярск, 2004. – С. 171–172.

УДК 519.115

Генерация лабиринта с заданными позициями входа и выходов

Т.М. Тушкина, Н.В. Павлова
БТИ (филиал) АлтГТУ, г. Бийск

Авторы разрабатывают модель сборного лабиринта с одним входом и несколькими выходами. Лабиринт будет служить элементом технопарка для детей дошкольного и младшего школьного возраста. Для эффективности функционирования лабиринта необходимо обеспечивать его вариативность. Положим, что лабиринт реализуется на сетке размерности $m \times n$, узлам сетки соответствуют

опоры. На опорах крепятся стенки. Форма лабиринта меняется в зависимости от того, как установлены стенки на опорах. По периметру сетки выбирается 1 позиция входа и k позиций выходов. Положим, что внутри лабиринта расположено l стенок. В работе рассматриваются две взаимосвязанные задачи. Пересчет: сколько существует способов установить l стенок внутри лабиринта? Генерация: в рамках указанных допущений перебрать все возможные формы лабиринтов.

Общее количество стенок лабиринта размерности $m \times n$ определяется посредством выражения $2mn+m+n$, при этом внешних (ограничивающих) стенок будет $2(m+n)$, остальные $2mn-m-n$ будут внутренними. Пронумеруем все элементарные стенки сетки числами от 1 до $2mn+m+n$. Выберем позиции входа и выходов. Их будет $1+k$.

Введем обозначения: 0 – стенка есть, 1 – стенки нет. Тогда каждому лабиринту будет ставиться в соответствие число, записанное $2mn+m+n$ символами из множества $\{0, 1\}$, причем в разрядах, соответствующих выбранным позициям входа и выходов, будут записаны 1, а в разрядах, соответствующих остальным ограничивающим стенкам – 0. Разряды двух указанных видов будем называть зарезервированными. Далее нужно расставить в оставшихся незарезервированных $2mn-m-n$ разрядах символы таким образом, чтобы было l символов 0, а остальные 1. Переход от одного из возможных вариантов к другому получается путем перестановки 0 и 1 в незарезервированных разрядах. Число всевозможных вариантов расстановки стенок на опорах будет определяться числом перестановок с повторением $(2mn-m-n)$ элементов, где l элементов первого класса (число 0) и $(2mn-m-n-l)$ элементов второго класса (число 1), т.е. определяется отношением:

$$\frac{(2mn-m-n)!}{l!(2mn-m-n-l)!}$$

Например, используя сетку 4×5 , положив, что выходов будет 4 (в фиксированных позициях), а стенок внутри лабиринта 12, получим, что стенки внутри лабиринта можно расставить:

$$\frac{(2 \cdot 4 \cdot 5 - 4 - 5)!}{12!(2 \cdot 4 \cdot 5 - 4 - 5 - 12)!} = \frac{31!}{12!19!} = 141120525 \text{ способами.}$$

Задача перечисления лабиринтов решается алгоритмически. На начальном этапе вводятся данные: количество строк (m) и столбцов лабиринта (n), количество выходов из лабиринта (k), количество стенок внутри лабиринта (l). В результате определяется число $2mn+m+n$ разрядов числа с цифрами 0 и 1, строится каркас лабиринта с пронумерованными стенками. Далее из предложенного перечня выбираются позиции входа и выходов. Символам в соответствующих разрядах присваиваются значения 1.

На следующем этапе происходит генерация лабиринта. Для этого в незарезервированных позициях записывается число i в двоичной системе счисления ($i = 0 \dots 2^{2mn-m-n} - 1$). Если арифметическая сумма символов во всех разрядах числа i равна $2mn-m-n-l+k+1$, то код лабиринта выводится на экран, в противном случае $i := i+1$, переход к началу цикла. В результате этих действий определяются двоичные числа (коды лабиринтов), удовлетворяющие всем поставленным условиям. Для каждого кода лабиринта на экране появляется соответствующая ему схема. Предполагается продемонстрировать посетителям технопарка работу программы генерации лабиринта, а также снабжать их схемой реализованного в настоящий момент лабиринта для обучения в игровой форме ориентированию на местности по карте.

УДК 519.23

Сильная согласованность в задачах восстановления зависимостей по данным с интервальной неопределённостью

С.П. Шарый

*Институт вычислительных технологий СО РАН,
г. Новосибирск*

Предметом нашей работы является развитие методов анализа данных, имеющих интервальную неопределённость.

Мы рассматриваем задачу восстановления линейной зависимости вида

$$b = a_1 x_1 + a_2 x_2 + \dots + a_n x_n,$$

в которой неизвестные коэффициенты x_1, x_2, \dots, x_n должны быть определены на основе ряда измерений значений a_1, a_2, \dots, a_n (входных переменных) и b (выхода). Измерения неточны, и мы