*Министерство образования и науки*

*Республики Казахстан*

*Петропавловский колледж*

*железнодорожного транспорта*

*СПЕЦИАЛЬНОСТЬ*

*«Вычислительная техника и программное обеспечение»*

*КУРСОВОЙ ПРОЕКТ*

*по предмету: «Моделирование производственных и экономических процессов»*

*на тему: «Создание модели для решения задач статистического моделирования методом Монте - Карло»*

*Проверил: Bыполнил:  
Петрищев К.Р. Хайдаев Я.Е.  
 гр. ПО-21*

*2015*СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ 4
   1. АНАЛИТИЧЕСКАЯ ЧАСТЬ 6
      1. Постановка задачи 6
      2. Математическая реализация 8
   2. ПРАКТИЧЕСКАЯ ЧАСТЬ 9
      1. Описание программы 9
      2. Структура программы на языке С++ 12
2. ЗАКЛЮЧЕНИЕ 14
3. СПИСОК ЛИТЕРАТУРЫ
4. ПРИЛОЖЕНИЕ А

ВВЕДЕНИЕ

Статистика наука, зародившаяся еще в Шумерском царстве и применяемая во множестве аспектов человеческой жизни, при необходимости рассчитать и сопоставить данные.

В рамках данного курсового проекта будет рассматриваться тема статистических методов путём решения одной из задач подходящей для применения подобных расчётов.

Статистика — отрасль знаний, в которой излагаются общие вопросы сбора, измерения и анализа массовых статистических (количественных или качественных) данных; изучение количественной стороны массовых общественных явлений в числовой форме.

Статистические методы — методы анализа статистических данных. Выделяют методы прикладной статистики, которые могут применяться во всех областях научных исследований и любых отраслях народного хозяйства, и другие статистические методы, применимость которых ограничена той или иной сферой. Имеются в виду такие методы, как статистический приемочный контроль, статистическое регулирование технологических процессов, надежность и испытания, планирование экспериментов.

Статистические методы анализа данных применяются практически во всех областях деятельности человека. Их используют всегда, когда необходимо получить и обосновать какие-либо суждения о группе (объектов или субъектов) с некоторой внутренней неоднородностью.

Целесообразно выделить три вида научной и прикладной деятельности в области статистических методов анализа данных (по степени специфичности методов, сопряженной с погруженностью в конкретные проблемы):

1. Разработка и исследование методов общего назначения, без учета специфики области применения;
2. Разработка и исследование статистических моделей реальных явлений и процессов в соответствии с потребностями той или иной области деятельности;
3. Применение статистических методов и моделей для статистического анализа конкретных данных.

Кратко рассмотрим три только что выделенных вида научной и прикладной деятельности. По мере движения от 1. к 3. сужается широта области применения конкретного статистического метода, но при этом повышается его значение для анализа конкретной ситуации. Если работам вида 1. соответствуют научные результаты, значимость которых оценивается по общенаучным критериям, то для работ вида 3. основное - успешное решение конкретных задач той или иной области применения (техники и технологии, экономики, социологии, медицины и других). Работы вида 2. занимают промежуточное положение, поскольку, с одной стороны, теоретическое изучение свойств статистических методов и моделей, предназначенных для определенной области применения, может быть весьма сложным и математизированным с другой - результаты представляют не всеобщий интерес, а лишь для некоторой группы специалистов. Можно сказать, что работы вида 2. нацелены на решение типовых задач конкретной области применения.

При применении статистических методов в конкретных областях знаний и отраслях народного хозяйства получаем научно-практические дисциплины типа "статистические методы в промышленности", "статистические методы в медицине" и др. С этой точки зрения эконометрика - это "статистические методы в экономике". Эти дисциплины группы 2. обычно опираются на вероятностно-статистические модели, построенные в соответствии с особенностями области применения. Весьма поучительно сопоставить вероятностно-статистические модели, применяемые в различных областях, обнаружить их близость и вместе с тем констатировать некоторые различия. Так, видна близость постановок задач и применяемых для их решения статистических методов в таких областях, как научные медицинские исследования, конкретные социологические исследования и маркетинговые исследования, или, короче, в медицине, социологии и маркетинге. Они часто объединяются вместе под названием «выборочные исследования».

Отличие выборочных исследований от экспертных проявляется, прежде всего, в числе обследованных объектов или субъектов – в выборочных исследованиях речь обычно идет о сотнях, а в экспертных – о десятках. Зато технологии экспертных исследований гораздо изощрённее. Еще более выражена специфика в демографических или логистических моделях, при обработке нарративной (текстовой, летописной) информации или при изучении взаимовлияния факторов.

Одним из примеров таких приемов является Монте-Карло – это группа численных методов, основанных на получении большого числа реализаций стохастического (случайного) процесса, который формируется таким образом, чтобы его вероятностные характеристики совпадали с аналогичными величинами решаемой задачи. Используется для решения задач в различных областях физики, химии, математики, экономики, оптимизации, теории управления и других.

В рамках курсового проекта будет детально рассмотрен метод Монте-Карло, а также разработана модель для автоматизации статистических вычислений.

В качестве примера применения данной модели рассмотрим задачу по нахождению площади многоугольника. Разработка модели и приложения ведется на языке С++ в интегрированной среде разработки Microsoft Visual Studio 2013.

1. АНАЛИТИЧЕСКАЯ ЧАСТЬ.
   1. Постановка задачи

Целью курсового проекта является разработка модели, позволяющей, реализовывать системы решения статистических задач в областях, где применим метод Монте-Карло.

Для начала рассмотрим час метод. Из определения следует что принцип действия заключается в многократном проведении испытания со случайными процессами, такими как генерирование случайных величин. И произведении расчетов над ними. Также одним из принципов данного метода сопоставление некоторых групп с соседними, либо же некого множества с более крупным множества. Исходя из, вышесказанного ясно что для реализации вычислительной модели потребуется генератор (псевдо)случайных чисел. Он как раз и необходим для создания выше названых областей значений в заданных пределах. Затем путем нахождения, внутренних величин, соответствующих определённым критериям и сопоставления числа удовлетворяющих – общему числу можно получить, некоторые аналитические данные.

Генераторы случайных чисел делятся на несколько типов: физические, табличные и алгоритмические. Для данной задачи наилучшим вариантом будет последний. Существуют различные алгоритмы вычисления псевдослучайных чисел, среди них чаще всего применяются метод срединных произведений и метод срединных квадратов. Оба эти метода основываются на произведении, соответствующих математических операций над заранее заданными числами и извлечении, срединных цифр из результата вычислений.

Так как для проведения какого-либо анализа требуются исходные данные, необходимо разработать метод ввода данных, из файла и/или из командной строки в качестве параметров. Аналогичные требования к системе вывода данных.

В качестве примера применения модели вычислений методом Монте-Карло, как уже говорилось ранее, будет реализовано приложение для нахождения площади произвольного, многоугольника на плоскости. Из этого следует что необходима математическая модель работы с фигурами и точками на плоскости.

Систематизируем задачи:

1. Разработать и реализовать генератор случайных чисел;
2. Разработать модель для расчетов, связанных с фигурами на плоскости;
3. Разработать систему ввода данных: ввод из командной строки, чтение из указанного в параметрах файла;
   1. Математическая реализация

Первым делом следует рассмотреть подробнее способы получения псевдослучайных чисел:

Метод срединных произведений состоит в том, что на вход подаются 2 или более чисел, эти числа перемножаются друг на друга, затем из полученного произведения извлекаются средние цифры. Традиционно генераторы случайных чисел генерируют числа от 0 до 1, рассмотрим пример работы:

Имеются некоторые четырехзначные числа R0 и R1. Эти числа умножаются друг на друга и заносится в R2. Далее для следующей операции берется R1 и из R2 берется середина (четыре средних цифры), снова умножаются, и записывается в R3. Затем данная процедура проделывается некоторое количество раз для обеспечения большей распределённости случайных величин при каждом вызове генератора случайных чисел. Следует отметим, что на самом деле в качестве случайного числа необходимо брать не ghij, а 0.ghij — с приписанным слева нулем и десятичной точкой. Вышеописанный процесс проиллюстрирован на рисунке 2.

R0 \* R1 = R2

R1 \* R2\* = R3

R2\* · R3\* = R3

0.R2\*

0.R3\*

Случайные числа

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| e | f | g | h | i | j | k | l |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| e | f | g | h | i | j | k | l |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| e | f | g | h | i | j | k | l |

**Рисунок 1 –** иллюстрация метода срединных произведений

Метод срединных квадратов выглядит следующим образом, на вход подаётся число, это число возводится в квадрат, затем из полученного результата, аналогично предыдущему методу извлекаются средние цифры. Рассмотрим пример работы данного способа:

Имеется некоторое четырехзначное число R0. Это число возводится в квадрат и заносится в R1. Далее из R1 берется середина (четыре средних цифры) — новое случайное число — и записывается в R0. Затем процедура повторяется. Отметим, что на самом деле в качестве случайного числа необходимо брать не ghij, а 0.ghij — с приписанным слева нулем и десятичной точкой.

R0

R1 = R0 \* R0

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | c | d |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| e | f | g | h | i | j | k | l |

**Рисунок 2 –** иллюстрация метода срединных квадратов

Для гарантированного равномерного распределения случайных чисел, и не допущения повторных значений в качестве входных чисел будем применять системные миллисекунды, а так же числа от 1111 до 9999, сгенерированные стандартной функцией С++ – rand().

Перейдём к рассмотрению работы с фигурами. Задача предполагает нахождение площади фигуры методом Монте-Карло. На вход будут подаваться координаты фигуры(искомой). Затем её необходимо вписать в большую(контрольную) фигуру с известной площадью. Далее сгенерировать множество случайных точек, которые должны принадлежать контрольной фигуре. На следующем этапе необходимо вычислить количество, случайных точек, попавших в пространство искомой фигуры. Отношение числа точек принадлежащих, искомой фигуре в общему количеству, точек в контрольном многоугольнике, умноженная на площадь, контрольной фигуры будет приближенным значением, искомой площади. Путем проведения многократных испытаний и вычислением среднего значения получится наиболее точная величина площади.

Из вышесказанного следует, что при получении координат, необходимо выполнять проверку не корректных данных, например, координат прямой линии, так как линия не может иметь площадь. Это реализовывается следующим образом: Во-первых, многоугольник имеет как минимум три точки. Во-вторых, ни одна область, состоящая, из вышеназванных вершин, недолжна иметь площадь равную нулю. Значит для данной проверки разобьём многоугольник на области, состоящие из трёх точек и вычислим их площади, если ни одна из площадей не ровна нулю, то введены координаты многоугольника, иначе – прямой.

Необходим алгоритм проверки точки на принадлежность к области многоугольника. Для это из каждой случайной точки опустим лучи один параллельный оси абсцисс, другой – ординат. Далее если луч параллельный абсцисс будет пересекать четное количество сторон фигуры, больше нуля – точка расположена ниже фигуры, если количество пересечений равно нулю – точка выше фигуры, если же количество пересечений не четно, то координата точки по оси абсцисс находится в приделах фигуры. Аналогичные вычисления следует произвести с лучом параллельным оси ординат, если количество пересечений не четно, координата точки по ординатам находится в области многоугольника. В случае, соблюдения всех перечисленных условий точка располагается внутри данной фигуры. Данные операции следует произвести со всеми, сгенерированными, случайными точками. Рассмотрим пример определения принадлежности точки к фигуре:

На плоскости нарисован пятиугольник A(2; 4), B(2; 10), C(4; 14), D(6; 8), E(4; 4), а также точки M(1; 6), N(4; 4), K(5; 12). Проведем из каждой точки по два луча, один – вверх, другой – вправо. Как можно увидеть на рисунке 3 луч «a», выходящий из точки M не пересекает ни одну из сторон многоугольника ABCDE, но луч «b» пересекает две из них, значит точка M находится вне фигуры – левее. Луч «с», исходящий из точки N пересекает одну из сторон пятиугольника в точке C, а луч «d» – пересек сторону ED, следовательно, точка N расположена внутри области фигуры. Далее точка K, оба её луча не имеют ни одного пересечения, ни с одной из сторон ABCDE, так же на примере этой точки можно заметить, что если бы применялся метод сравнения координат случайных точек и координат вершин фигуры, то в отношении точек с подобным расположением не удалось бы получить верный ответ.

**Рисунок 3** – иллюстрация проверки точки на принадлежность к области фигуры

На основе выше описанных алгоритмов реализуется задача по нахождению площади фигуры. Что будет продемонстрировано, в следеющем примере:

Нахождение площади фигуры методом Монте-Карло. Определить методом Монте-Карло площадь пятиугольника с координатами углов (0, 0), (0, 10), (5, 20), (10, 10), (7, 0). Рисуется в двухмерных координатах заданный пятиугольник, вписываясь в прямоугольник, чья площадь, как нетрудно догадаться, составляет (10 – 0) • (20 – 0) = 200, рисунок 4.

**Рисунок 4** – иллюстрация нахождения площади

Используем таблицу случайных чисел для генерации пар чисел R, G, равномерно распределенных в интервале от 0 до 1. Число R будет имитировать координату X (0 ≤ X ≤ 10), следовательно, X = 10 · R. Число G будет имитировать координату Y (0 ≤ Y ≤ 20), следовательно, Y = 20 · G. Сгенерируем по 10 чисел R и G и отобразим 10 точек (X; Y) на рисунке 4 и в таблице 1

Таблица 1 - Решение задачи методом Монте-Карло

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер | R | G | X | Y | Точка (X; Y) попала в KLMN? | Точка (X; Y) попала в ABCDE? |
| 1 | 0,543374 | 0,554705 | 5,433739 | 11,0941 | Да | Да |
| 2 | 0,663769 | 0,32263 | 6,637686 | 6,452594 | Да | Да |
| 3 | 0,132668 | 0,325595 | 1,326676 | 6,511908 | Да | Да |
| 4 | 0,332327 | 0,490945 | 3,323272 | 9,818905 | Да | Да |
| 5 | 0,057314 | 0,112871 | 0,573144 | 2,257425 | Да | Да |
| 6 | 0,685227 | 0,856813 | 6,852274 | 17,13625 | Да | Нет |
| 7 | 0,868189 | 0,056411 | 8,681893 | 1,128222 | Да | Нет |
| 8 | 0,152591 | 0,888838 | 1,525913 | 17,77675 | Да | Нет |
| 9 | 0,635132 | 0,223465 | 6,351315 | 4,46931 | Да | Да |
| 10 | 0,880999 | 0,089966 | 8,809988 | 1,799325 | Да | Нет |
|  |  |  |  |  | 10 | 6 |

Статистическая гипотеза заключается в том, что количество точек, попавших в контур фигуры, пропорционально площади фигуры: 6:10 = S:200. То есть, по формуле метода Монте-Карло, получаем, что площадь S пятиугольника равна: 200 • 6/10 = 120. Проследим, как менялась величина S от опыта к опыту, таблица 2

Таблица 2 - Оценка точности ответа

|  |  |  |
| --- | --- | --- |
| Количество испытаний N | Оценка вероятности попадания случайной точки в испытуемую область | Оценка площади S методом Монте-Карло |
| 1 | 1/1 = 1.00 | 200 |
| 2 | 1/2 = 0.50 | 100 |
| 3 | 2/3 = 0.67 | 133 |
| 4 | 3/4 = 0.75 | 150 |
| 5 | 3/5 = 0.60 | 120 |
| 6 | 4/6 = 0.67 | 133 |
| 7 | 5/7 = 0.71 | 143 |
| 8 | 5/8 = 0.63 | 125 |
| 9 | 6/9 = 0.67 | 133 |
| 10 | 6/10 = 0.60 | 120 |

Точность результата растет с ростом количества испытаний, в этом и состоит метод Моте-Карло – проанализировать некоторое количество данных в диапазоне при некотором количестве испытаний и получить наиболее точные данные.

1. ПРАКТИЧЕСКАЯ ЧАСТЬ
   1. Описание программы

Первый элемент программы – генератор случайных чисел, он будет основан на совокупности нескольких источников псевдослучайных величин таких, как время, стандартный генератор случайных чисел и два вида алгоритмических.

Одним из алгоритмических методов генерирования случайных чисел будет – метод срединных произведений. В качестве первого множителя станет число, полученное путем сложения количества миллисекунд, полученных при двух запросах к системному времени. Второй множитель – число, сгенерированное стандартным генератором в диапазоне от 1111 до 9999. Количество шагов генерирования, также будет зависеть от количества миллисекунд.

Вторым алгоритмом станет метод срединных квадратов, в качестве начального числа, возводимого в квадрат, станет – сгенерированное стандартным ГСЧ, количество шагов генерации также будет зависеть от значения миллисекунд.

Далее все вышеперечисленные способы получения псевдослучайных чисел будет объединены, следующим образом: берется сумма системных миллисекунд, число, сгенерированное стандартным генератором, затем много кратно повторяется, следующие действия:

1. В генератор методом срединных произведений подаются выше полученные числа;
2. Генерируется число и подается в генератор срединных квадратов;
3. Последнее, в качестве второго операнда снова подаётся в генератор срединных произведений.

Последним действием полеченное четырехзначное число разделить на 10000. Таким образом можно получить величины, максимально распределенные в диапазоне от 0 до 1. Для получения случайных чисел в заданном диапазоне, сгенерированное число необходимо умножить на разность, максимального и минимального чисел диапазона, а затем, полученное произведение сложить, с максимальным числом диапазона.

Следующим звеном программы является математическая модель работы с фигурами. Любая фигура состоит из точек или вершин и сторон или ребер. Точки имеют лишь координаты на плоскости, но ребра, так как они являются линиями обладают коэффициентами углового и вертикального смещения. Коэффициенты должны вычисляться на этапе инициализации объекта ребро. Для вычисления данных коэффициентов следует вспомнить простое уравнение прямой

где

x, y – координаты точки;

k – коэффициент углового смещения;

b – коэффициент вертикального смещения

Множитель k для отрезка прямой вычисляется отношением разностей координат конечной и начальной точки, разность y1 и y0 делится на разность x1 и x0.

Вычисление b происходит путем подстановки известных координат любой из точек, коэффициента k и преобразования уравнения прямой, следующим образом:

Таким образом получается сторона фигуры со всеми необходимыми параметрами для вычисления столкновений, её и луча исходящего из случайной точки.

Для выявления столкновения луча и прямой есть два подхода. Первый – вычислить все точки, принадлежащие прямой, а так все точки, принадлежащие лучу на данном отрезке координат проверить каждую точку луча и прямей на совпадение координат. Второй способ – зная координаты случайной точки и коэффициенты прямой, путем подстановки в уравнение прямой можно узнать координаты пересечения, проверяемого отрезка и ребра фигуры. Так как необходимо проверять лучи, направленные вверх и вправо – расположение, проверяемой точки с расположением точки пересечения следует сравнить, следующим образом:

1. Если координата X точки, испускающей луч вправо меньше ординаты места пересечения линий – луч имеет пересечение, иначе пересечения нет.
2. Если координата Y точки, испускающей луч вверх меньше абсциссы места пересечения отрезков – пересечение луча и ребра существует, иначе пересечения с вертикальным лучом нет.

Как можно заметить второй метод более рационален по потребности в ресурсах памяти и времени выполнения, с минимальным количеством вычислений и проверок.

Для решения задачи по нахождению площадей фигур необходимо проверять точки на принадлежность фигуре. Выше описанный метод позволит проверить каждый луч, каждой случайной точки на пересечение, с каждым ребром фигуры. При наличии нечетного количества пересечений у обоих лучей точки – последняя считается, принадлежащей к фигуре.

Важной частью приложения является система ввода вывода. Здесь будет присутствовать как ввод параметров из командной строки, как и файловый ввод свойств многоугольника, а также вывод результата в виде текстового файла или строки в окне команд.

* 1. Структура программы на языке С++

Каждое предприятие для получения прибыли ставит перед собой задачу производить что-либо или оказывать услуги. Веб-студия ставит задачу создавать веб-ресурсы для своих клиентов. При поступлении заказа задачи распределяются среди различных отделов, отдел продаж опрашивает клиента и разрабатывает техническое задания для дизайнера, после выполнения дизайна ставится задача перед техническим отделом на реализацию проекта.

Для успешной и продуктивной работы необходима хорошая техника с необходимыми возможностями и производительностью. А также техника должна быть настроена надлежащим образом. То есть иметь необходимый набор драйверов для корректной работы аппаратного комплекса и программ для выполнения поставленных задач перед сотрудником.

Для обеспечения корректной и производительной работы офисную технику необходимо обслуживать должным образом. В профилактических целях проводить диагностику программных средств на наличие вирусов ошибок и прочих неполадок. А так же проводить уборку в помещении для снижения загрязнения различных частей устройств

Для работы веб приложений следует применять языки текстовой разметки документов, с целью воссоздать необходимую структуру документа. В случаях, когда задача ставит изменение документа в зависимости от различных сложившихся ситуаций, не зависящих от пользователя, документа используются языки серверной разметки, на пример PHP, Perl или Python. Когда же стоит задача динамического реагирования страницы на поведение пользователя используются клиентские языки стилей такие, как СSS или программирования, как JavaScript.

При поступлении задания программист должен точно осознавать суть поставленной задачи, исходя из этого он выбирает наиболее оптимальные методы решения данных проблем.

При создании особых проектов не работа начинается с написания классов и функция различных объектов. Затем происходит сборка необходимых объектов во едино.

Примером подобного проекта может являться ресурс содержащий интерактивную таблицу химических элементов. Для такой задачи необходимы такие объекты как таблица, столбцы, ряды, ячейки. Все эти компоненты должны выстраиваться в закономерном порядке по средством специальных алгоритмов и наборов данных. Из подобных элементов может состоять любой проект, и в каждом из них будет присутствовать что-то особенное.

Выбор языка программирования зависит от поставленной задачи и предоставляемых ресурсов, данных и что главное у меняй программиста. Для построения веб приложений часто используют PHP или C#, в качестве серверных языков, а на клиентской стороне JavaScript или JQuery.

Для исправного функционирования вычислительной техники её следует проверять на наличие угроз безопасности, а так же следить за внешней целостностью. Таким образом при каждом начале или окончании рабочего дня необходимо прибирать рабочее место, это качается и влажной уборки, рекомендуется протирать внешние панели оргтехники влажной тканью.

При обнаружении не исправностей обращаться в знающим людям в данной области .

На современных предприятиях применяются сетевые технологии, для ускорения обменом информацией с заказчиками, поставщиками либо выполнения прямых производственных задач. По мимо аппаратного построения сетей под сетевыми технологиями подразумеваются также организация веб-ресурсов. Для осуществления работы сети интернет и размещения в ней различных данных применяются серверы, компьютеры с большой производительностью и расширенными сетевыми возможностями. Для удобства обслуживания и снижения риска получения доступа к таким машинам третьими лицами организуются Internet Data Center, такие центры создаются организациями, предоставляющими хостинг-пространство или связь на уровне государства. В данных помещения организуются специальные условия для обеспечения безопасности для устройств от перегрева или других факторов несущих вред технике.

Локальные сети предприятий чаще всего организуются сети с топологией звезда, так как это ни более надежное и простое решения для учреждений не специализирующихся на информационных технологиях. Соединение компьютеров все чаще осуществляется по средством WI-FI и все реже кабелями Ethernet, но второе решение более подходит при создании сетей большей протяженности, где беспроводные технологии не позволяют покрыть необходимый радиус.

ЗАКЛЮЧЕНИЕ

Производственно-технологическая практика позволила познакомиться с реальной ситуацией на рабочем месте, это дает понять учащемуся какие условия и задачи будут стоять перед ним при устройстве на работу. У каждой должности есть свои особенности, плясы и минусы, которые можно было увидеть и почувствовать во время выполнения заданий, устанавливаемых предприятием, а также при выполнении индивидуального задания. Удалось познакомиться с рабом технологий ускоряющих разработку веб-ресурсов, условиями, организуемыми при создании Internet Data Centr. Общение с профессионалами позволяет получить определенный опыт и знания от знающих специалистов.

ПРИЛОЖЕНИЕ A

index.php

<html>

<head>

<meta charset="utf-8"/>

<title>MyPHPpage 4</title>

<link rel="stylesheet" type="text/css" href="style.css"/>

</head>

<body>

<?php

include\_once 'php classes/stdafx.php';

//sql vars

$host = "localhost";

$dbName = "\*\*\*\*\*\*\*\*\*\*\*";

$user = "\*\*\*\*\*\*\*\*\*\*\*";

$pwd = "\*\*\*\*\*\*\*\*\*\*\*";

//------------------

$d = new DOMDocument('1.0', 'UTF-8');

//sql connect

/\*

$dbh = mysql\_connect($host,$user,$pwd)

or die("Не могу подключиться к MySQL");

mysql\_select\_db($dbName)

or die("Не могу подключиться к базе");

\*/

//-------------

//elements data arrays

$SQLSymbol = array(1,);//[118];

$SQLLatName = array(1,);//[118];

$SQLRusName = array(1,);//[118];

$SQLMass = array(1,);//[118];

$SQLGroup = array(1,);//[118];

$SQLsubGroup = array(1,);//[118];

$SQLPeriod = array(1,);//[118];

$SQLrow = array(1,);//[118];

//--------------------

$layout = new layout($d);

$layout->setCSSselectors("layout","");

//table cjmponent objects arrays and vars

$objTable = new table($d);// объект таблица

$objCol = array(1,);// [11] объект столбецы периодов, рядов, групп, электронных уровней

$objColLabelCell = array(1,);// [11] ячейки для шапки таблицы

$objColLabel = array(1,);// [11]

$objRowsInPeriod = array(1,);// [7] ряды в столбце периодов

$objCellsNumOfPeriod = array(1,);// [7] ячейки в рядах столбцов периодов

$objNumPerLabel = array(1,);// [7] надписи номеров

// ряды в столбце нумерации рядов

$objR\_GLabelCells = array(1,); // [2] ячейки ряд\группа

$objR\_GLabel = array(1,);// [2] надписи ряд\группа

$objNumRow = array(1,);// [11] ряды в столбце ряд

$objNumRowCell = array(1,);// [11] ячейки в рядах

$objNumRowLabels = array(1,);// [11] номера рядов в ячейках

// группы

$objRowsInGroup = array(1,);// [88] ряды в гуппах

$objSubGroupLabel = array(1,);// [16] подпись подгруппы

$objElementCell = array(1,);// [118] ячека с информацией об элементе

$objSymLabel = array(1,);// [118] обозначение

$objElNumberLabel = array(1,);// [118] периодический номер

$objMassLabel = array(1,);// [118] масса атома

$objNameParagraf = array(1,);// [118] наименование на латынии и на русском языке

$objElectronFormula = array(1,);// [118] электронная формула

$objElementElectronLvL = array(1,);// [118] распределение электронов по энерго уровням

$lvlK = array(1,);// [118]

$lvlL = array(1,);// [108]

$lvlM = array(1,);// [106]

$lvlN = array(1,);// [100]

$lvlO = array(1,);// [82]

$lvlP = array(1,);// [64]

$lvlQ = array(1,);// [32]

$objElementCellSeparat = array(236);

//электронные уровни

$objElFormatP;// в основной таблице

$objElectronLvlLabelCell;

$objLaAcElectronLabelCell;// в лантанойдах и актиноидах

$objLaAcRows = array(1,);// [12]

$objLaAcRow= array(1,);

$objLaAcElFormatP = array(1,);

$objZoomElectronCol;// в увеличении

$objZoomElFormatP;

$cellCount = 1;

$mass = 1.0079;

$exp = 2.7;

for($i=0; $i<10; $i++) // <10

{

if($i == 0)

{

$objCol[$i] = new field($d);

$objCol[$i]->setCSSselectors("ColPeriod", "Col");

$objColLabelCell[$i] = new cell($d);

$objColLabelCell[$i]->setCSSselectors("ColLabelPeriodCell", "ColLabelCell");

$objColLabel[$i] = new label($d, 'П');

$objColLabel[$i]->setCSSselectors("ColLabelPeriod", "ColLabel");

$objColLabelCell[$i]->DOMElement->appendChild($objColLabel[$i]->DOMElement);

$objCol[$i]->DOMElement->appendChild($objColLabelCell[$i]->DOMElement);

for($j=0; $j<1; $j++)

{

$objRowsInPeriod[$j] = new row($d);

$objRowsInPeriod[$j]->setCSSselectors("RowInPeriod{$j}", "RowInPeriod");

$objCellsNumOfPeriod[$j] = new cell($d);

$objCellsNumOfPeriod[$j]->setCSSselectors("CellNumPeriod{$j}", "CellNumPeriod");

$objNumPerLabel[$j] = new Label($d, $j+1);

$objNumPerLabel[$j]->setCSSselectors("NumPeriodLabel{$j}", "NumPeriodLabel");

$objCellsNumOfPeriod[$j]->DOMElement->appendChild($objNumPerLabel[$j]->DOMElement);

$objRowsInPeriod[$j]->DOMElement->appendChild($objCellsNumOfPeriod[$j]->DOMElement);

$objCol[$i]->DOMElement->appendChild($objRowsInPeriod[$j]->DOMElement);

}

}

if($i == 1)

{

$objCol[$i] = new field($d);

$objCol[$i]->setCSSselectors("R\_GCol","Col");

$objColLabelCell[$i] = new cell($d);

$objColLabelCell[$i]->setCSSselectors("R\_GColLabelCell","ColLabelCell");

for($j=0; $j<2; $j++)

{

$RGLN = $j+1;

$objR\_GLabelCells[$j] = new cell($d);

$objR\_GLabelCells[$j]->setCSSselectors("R\_GLabelCell{$RGLN}","R\_GLabelCells");

if($j == 0)

{

$objR\_GLabel[$j] = new Label($d, 'Г');

$objR\_GLabel[$j]->setCSSselectors("R\_GLabel{$RGLN}","R\_GLabel");

}

else

{

$objR\_GLabel[$j] = new Label($d, 'Р');

$objR\_GLabel[$j]->setCSSselectors("R\_GLabel{$RGLN}","R\_GLabel");

}

$objR\_GLabelCells[$j]->DOMElement->appendChild($objR\_GLabel[$j]->DOMElement);

$objColLabelCell[$i]->DOMElement->appendChild($objR\_GLabelCells[$j]->DOMElement);

}

$objCol[$i]->DOMElement->appendChild($objColLabelCell[$i]->DOMElement);

for($j=0; $j<2; $j++)//11

{

$objNumRow[$j] = new Row($d);

$objNumRow[$j]->setCSSselectors("NumRow{$j}","NumRow");

$objNumRowCell[$j] = new cell($d);

$objNumRowCell[$j]->setCSSselectors("NumRowCell{$j}","NumRowCell");

$objNumRowLabel[$j] = new Label($d, $j+1);

$objNumRowLabel[$j]->setCSSselectors("NumRowLabel{$j}","NumRowLabel");

$objNumRowCell[$j]->DOMElement->appendChild($objNumRowLabel[$j]->DOMElement);

$objNumRow[$j]->DOMElement->appendChild($objNumRowCell[$j]->DOMElement);

$objCol[$i]->DOMElement->appendChild($objNumRow[$j]->DOMElement);

}

}

if($i >= 2 && $i <= 7)

{

$GN = $i-1;// номер группы для селектора id

$objCol[$i] = new field($d);

$objCol[$i]->setCSSselectors("Group{$GN}","Group Col");

$objColLabelCell[$i] = new cell($d);

$objColLabelCell[$i]->setCSSselectors("GroupLabelCell{$GN}","GroupLabelCell");

switch($i)

{

case 2:

$objColLabel[$i] = new label($d, "&#8544;");

break;

case 3:

$objColLabel[$i] = new label($d, "&#8545;");

break;

case 4:

$objColLabel[$i] = new label($d, "&#8546;");

break;

case 5:

$objColLabel[$i] = new label($d, "&#8547;");

break;

case 6:

$objColLabel[$i] = new label($d, "&#8548;");

break;

case 7:

$objColLabel[$i] = new label($d, "&#8549;");

break;

case 8:

$objColLabel[$i] = new label($d, "&#8550;");

break;

}

$objColLabel[$i]->setCSSselectors("ColLabel{$GN}","ColLabel");

$objSubGroupLabel[$i] = new label($d, 'A');

$objSubGroupLabel[$i]->setCSSselectors("subGroupLabel\_A{$GN}","subGroupLabel\_A");

$objSubGroupLabel[$i+1] = new label($d, 'B');

$objSubGroupLabel[$i+1]->setCSSselectors("subGroupLabel\_B{$GN}","subGroupLabel\_B");

$objColLabelCell[$i]->DOMElement->appendChild($objSubGroupLabel[$i]->DOMElement);

$objColLabelCell[$i]->DOMElement->appendChild($objColLabel[$i]->DOMElement);

$objColLabelCell[$i]->DOMElement->appendChild($objSubGroupLabel[$i+1]->DOMElement);

$objCol[$i]->DOMElement->appendChild($objColLabelCell[$i]->DOMElement);

for($j=0; $j<=1; $j++)//11

{

$objRowsInGroup[$j] = new row($d);

$objRowsInGroup[$j]->setCSSselectors("Row{$j}InGroup{$GN}","RowInGroup");

$objElementCell[$cellCount] = new cell($d);

$objElementCell[$cellCount]->setCSSselectors("ElementCell{$cellCount}","ElementCell");

$objElementCellSeparat[$cellCount] = new cell($d);

$objElementCellSeparat[$cellCount]->setCSSselectors("ElementCellSeparate{$cellCount}\_0","ElementCellSeparate0");

$objSymLabel[$cellCount] = new label($d, "Si");

$objSymLabel[$cellCount]->setCSSselectors("SymbolLabel{$cellCount}","SymbolLabel");

$objNameParagraf[$cellCount] = new paragraf($d, "Name RusNam");

$objNameParagraf[$cellCount]->setCSSselectors("ElementName{$cellCount}","ElementName");

$objElementElectronLvL[$cellCount] = new fparagraf($d, "2\n8\n4\n18");

$objElementElectronLvL[$cellCount]->setCSSselectors("ElectronLvLDiff{$cellCount}","ElectronLvLDiff");

$objElementCellSeparat[$cellCount+1] = new cell($d);

$objElementCellSeparat[$cellCount+1]->setCSSselectors("ElementCellSeparate{$cellCount}\_1","ElementCellSeparate1");

$objElNumberLabel[$cellCount] = new label($d, $cellCount);

$objElNumberLabel[$cellCount]->setCSSselectors("ElementNumberLabel{$cellCount}","ElementNumberLabel");

$objMassLabel[$cellCount] = new label($d, $mass+=$exp);

$objMassLabel[$cellCount]->setCSSselectors("MassElementLabel{$cellCount}","MassElementLabel");

$objElectronFormula[$cellCount] = new label($d, "2s23p4");

$objElectronFormula[$cellCount]->setCSSselectors("ElectronFormula{$cellCount}","ElectronFormula");

$objElementCellSeparat[$cellCount]->DOMElement->appendChild($objSymLabel[$cellCount]->DOMElement);

$objElementCellSeparat[$cellCount]->DOMElement->appendChild($objNameParagraf[$cellCount]->DOMElement);

$objElementCellSeparat[$cellCount]->DOMElement->appendChild($objElectronFormula[$cellCount]->DOMElement);

$objElementCellSeparat[$cellCount+1]->DOMElement->appendChild($objElNumberLabel[$cellCount]->DOMElement);

$objElementCellSeparat[$cellCount+1]->DOMElement->appendChild($objMassLabel[$cellCount]->DOMElement);

$objElementCellSeparat[$cellCount+1]->DOMElement->appendChild($objElementElectronLvL[$cellCount]->DOMElement);

if(

($cellCount >= 21 && $cellCount <= 30) ||

($cellCount >= 39 && $cellCount <= 48) ||

($cellCount >= 57 && $cellCount <= 80) ||

($cellCount >= 89 && $cellCount <= 112)

)

{

$objElectronFormula[$cellCount]->setCSSselectors("","A");

$objElementCell[$cellCount]->DOMElement->appendChild($objElementCellSeparat[$cellCount+1]->DOMElement);

$objElementCell[$cellCount]->DOMElement->appendChild($objElementCellSeparat[$cellCount]->DOMElement);

}

else

{

$objElectronFormula[$cellCount]->setCSSselectors("","B");

$objElementCell[$cellCount]->DOMElement->appendChild($objElementCellSeparat[$cellCount]->DOMElement);

$objElementCell[$cellCount]->DOMElement->appendChild($objElementCellSeparat[$cellCount+1]->DOMElement);

}

$objRowsInGroup[$j]->DOMElement->appendChild($objElementCell[$cellCount]->DOMElement);

$objCol[$i]->DOMElement->appendChild($objRowsInGroup[$j]->DOMElement);

$cellCount++;

}

}

if($i == 8)

{

$GN = $i-1;

$objCol[$i] = new field($d);

$objCol[$i]->setCSSselectors("Group{$GN}", "Group Col");

$objColLabelCell[$i] = new cell($d);

$objColLabelCell[$i]->setCSSselectors("GroupLabelCell{$GN}", "ColLabelCell");

$objColLabel[$i] = new label($d, $i-1);

$objColLabel[$i]->setCSSselectors("ColLable{$GN}","ColLabel");

$objColLabelCell[$i]->DOMElement->appendChild($objColLabel[$i]->DOMElement);

$objCol[$i]->DOMElement->appendChild($objColLabelCell[$i]->DOMElement);

$objColLabel[$i]->setCSSselectors("ColLabel{$GN}","ColLabel");

$objSubGroupLabel[$i] = new label($d, 'A');

$objSubGroupLabel[$i]->setCSSselectors("subGroupLabel\_A{$GN}","subGroupLabel\_A");

$objSubGroupLabel[$i+1] = new label($d, 'B');

$objSubGroupLabel[$i+1]->setCSSselectors("subGroupLabel\_B{$GN}","subGroupLabel\_B");

$objColLabelCell[$i]->DOMElement->appendChild($objSubGroupLabel[$i]->DOMElement);

$objColLabelCell[$i]->DOMElement->appendChild($objColLabel[$i]->DOMElement);

$objColLabelCell[$i]->DOMElement->appendChild($objSubGroupLabel[$i+1]->DOMElement);

$objCol[$i]->DOMElement->appendChild($objColLabelCell[$i]->DOMElement);

for($j=0; $j<=11; $j++)

{

$objRowsInGroup[$j] = new row($d);

$objRowsInGroup[$j]->setCSSselectors("Row{$j}InGroup{$GN}","RowInGroup");

if(

($j >= 0 && $j <= 3) ||

$j == 5 || $j == 7 || $j == 9

)

{

$objElementCell[$cellCount] = new cell($d);

$objElementCell[$cellCount]->setCSSselectors("ElementCell{$cellCount}","ElementCell");

$objElementCellSeparat[$cellCount] = new cell($d);

$objElementCellSeparat[$cellCount]->setCSSselectors("ElementCellSeparate{$cellCount}\_0","ElementCellSeparate0");

$objSymLabel[$cellCount] = new label($d, "Si");

$objSymLabel[$cellCount]->setCSSselectors("SymbolLabel{$cellCount}","SymbolLabel");

$objNameParagraf[$cellCount] = new paragraf($d, "Name RusNam");

$objNameParagraf[$cellCount]->setCSSselectors("ElementName{$cellCount}","ElementName");

$objElementElectronLvL[$cellCount] = new fparagraf($d, "2\n8\n4\n18");

$objElementElectronLvL[$cellCount]->setCSSselectors("ElectronLvLDiff{$cellCount}","ElectronLvLDiff");

$objElementCellSeparat[$cellCount+1] = new cell($d);

$objElementCellSeparat[$cellCount+1]->setCSSselectors("ElementCellSeparate{$cellCount}\_1","ElementCellSeparate1");

$objElNumberLabel[$cellCount] = new label($d, $cellCount);

$objElNumberLabel[$cellCount]->setCSSselectors("ElementNumberLabel{$cellCount}","ElementNumberLabel");

$objMassLabel[$cellCount] = new label($d, $mass+=$exp);

$objMassLabel[$cellCount]->setCSSselectors("MassElementLabel{$cellCount}","MassElementLabel");

$objElectronFormula[$cellCount] = new label($d, "2s23p4");

$objElectronFormula[$cellCount]->setCSSselectors("ElectronFormula{$cellCount}","ElectronFormula");

$objElementCellSeparat[$cellCount]->DOMElement->appendChild($objSymLabel[$cellCount]->DOMElement);

$objElementCellSeparat[$cellCount]->DOMElement->appendChild($objNameParagraf[$cellCount]->DOMElement);

$objElementCellSeparat[$cellCount]->DOMElement->appendChild($objElectronFormula[$cellCount]->DOMElement);

$objElementCellSeparat[$cellCount+1]->DOMElement->appendChild($objElNumberLabel[$cellCount]->DOMElement);

$objElementCellSeparat[$cellCount+1]->DOMElement->appendChild($objMassLabel[$cellCount]->DOMElement);

$objElementCellSeparat[$cellCount+1]->DOMElement->appendChild($objElementElectronLvL[$cellCount]->DOMElement);

if(

($cellCount >= 21 && $cellCount <= 30) ||

($cellCount >= 39 && $cellCount <= 48) ||

($cellCount >= 57 && $cellCount <= 80) ||

($cellCount >= 89 && $cellCount <= 112)

)

{

$objElectronFormula[$cellCount]->setCSSselectors("","A");

$objElementCell[$cellCount]->DOMElement->appendChild($objElementCellSeparat[$cellCount+1]->DOMElement);

$objElementCell[$cellCount]->DOMElement->appendChild($objElementCellSeparat[$cellCount]->DOMElement);

}

else

{

$objElectronFormula[$cellCount]->setCSSselectors("","B");

$objElementCell[$cellCount]->DOMElement->appendChild($objElementCellSeparat[$cellCount]->DOMElement);

$objElementCell[$cellCount]->DOMElement->appendChild($objElementCellSeparat[$cellCount+1]->DOMElement);

}

$objRowsInGroup[$j]->DOMElement->appendChild($objElementCell[$cellCount]->DOMElement);

$objCol[$i]->DOMElement->appendChild($objRowsInGroup[$j]->DOMElement);

$cellCount++;

}

else

{

for($k=0; $k<3; $k++)

{

$objElementCell[$cellCount] = new cell($d);

$objElementCell[$cellCount]->setCSSselectors("ElementCell{$cellCount}","ElementCell");

$objSymLabel[$cellCount] = new label($d);

$objSymLabel[$cellCount]->setCSSselectors("SymbolLabel{$cellCount}","SymbolLabel");

$objElNumberLabel[$cellCount] = new label($d, $cellCount);

$objElNumberLabel[$cellCount]->setCSSselectors("ElementNumberLabel{$cellCount}","ElementNumberLabel");

$objMassLabel[$cellCount] = new label($d, $mass+=$exp);

$objMassLabel[$cellCount]->setCSSselectors("MassElementLabel{$cellCount}","MassElementLabel");

$objNameParagraf[$cellCount] = new paragraf($d);

$objNameParagraf[$cellCount]->setCSSselectors("ElementName{$cellCount}","ElementName");

$objElectronFormula[$cellCount] = new label($d);

$objElectronFormula[$cellCount]->setCSSselectors("ElectronFormula{$cellCount}","ElectronFormula");

$objElementElectronLvL[$cellCount] = new fparagraf($d);

$objElementElectronLvL[$cellCount]->setCSSselectors("ElectronLvLDiff{$cellCount}","ElectronLvLDiff");

if(

($cellCount >= 21 && $cellCount <= 30) ||

($cellCount >= 39 && $cellCount <= 48) ||

($cellCount >= 57 && $cellCount <= 80) ||

($cellCount >= 89 && $cellCount <= 112)

)

{

$objElementCell[$cellCount]->DOMElement->appendChild($objElNumberLabel[$cellCount]->DOMElement);

$objElementCell[$cellCount]->DOMElement->appendChild($objMassLabel[$cellCount]->DOMElement);

$objElementCell[$cellCount]->DOMElement->appendChild($objSymLabel[$cellCount]->DOMElement);

$objElementCell[$cellCount]->DOMElement->appendChild($objNameParagraf[$cellCount]->DOMElement);

$objElementCell[$cellCount]->DOMElement->appendChild($objElectronFormula[$cellCount]->DOMElement);

$objElementCell[$cellCount]->DOMElement->appendChild($objElementElectronLvL[$cellCount]->DOMElement);

}

else

{

$objElementCell[$cellCount]->DOMElement->appendChild($objSymLabel[$cellCount]->DOMElement);

$objElementCell[$cellCount]->DOMElement->appendChild($objElNumberLabel[$cellCount]->DOMElement);

$objElementCell[$cellCount]->DOMElement->appendChild($objMassLabel[$cellCount]->DOMElement);

$objElementCell[$cellCount]->DOMElement->appendChild($objNameParagraf[$cellCount]->DOMElement);

$objElementCell[$cellCount]->DOMElement->appendChild($objElectronFormula[$cellCount]->DOMElement);

$objElementCell[$cellCount]->DOMElement->appendChild($objElementElectronLvL[$cellCount]->DOMElement);

}

$objRowsInGroup[$j]->DOMElement->appendChild($objElementCell[$cellCount]->DOMElement);

$objCol[$i]->DOMElement->appendChild($objRowsInGroup[$j]->DOMElement);

$cellCount++;

}

}

}

}

if($i == 9)

{

$objCol[$i] = new field($d);

$objCol[$i]->setCSSselectors(" ElectronLvlCol","Group Col");

$objElectronLvlLabelCell= new cell($d);

$objElectronLvlLabelCell->setCSSselectors("ElectronLvlLabelCell","ElectronLvLLabelCell");

$objElectronLvlLabel = new Label($d, "Electron LvL");

$objElectronLvlLabel->setCSSselectors("ElectronLvlLabel","ElectronLvLLabel");

$objElectronLvlLabelCell->DOMElement->appendChild($objElectronLvlLabel->DOMElement);

$objCol[$i]->DOMElement->appendChild($objElectronLvlLabelCell->DOMElement);

for($j=0; $j<11; $j++)

{

$objElectronLvlRows[$j] = new row($d);

$objElectronLvlRows[$j]->setCSSselectors("ElectronLvlRow{$j}","ElectronLvLRow");

switch($j)

{

case 0:

$objElectronLabelfparagraf[$j] = new fparagraf($d, "K");

break;

case 1:

$objElectronLabelfparagraf[$j] = new fparagraf($d, "L\nK");

break;

case 2:

$objElectronLabelfparagraf[$j] = new fparagraf($d, "M\nL\nK");

break;

case 3:

case 4:

$objElectronLabelfparagraf[$j] = new fparagraf($d, "N\nM\nL\nK");

break;

case 5:

case 6:

$objElectronLabelfparagraf[$j] = new fparagraf($d, "O\nN\nM\nL\nK");

break;

case 7:

case 8:

$objElectronLabelfparagraf[$j] = new fparagraf($d, "P\nN\nM\nL\nK");

break;

case 9:

case 10:

$objElectronLabelfparagraf[$j] = new fparagraf($d, "Q\nP\nN\nM\nL\nK");

break;

}

$objElectronLabelfparagraf[$j]->setCSSselectors("ElectronLabelfparagraf{$j}","ElectronLabelfparagraf");

$objElectronLvlRows[$j]->DOMElement->appendChild($objElectronLabelfparagraf[$j]->DOMElement);

$objCol[$i]->DOMElement->appendChild($objElectronLvlRows[$j]->DOMElement);

}

}

$objTable->DOMElement->appendChild($objCol[$i]->DOMElement);

}

$layout->content->DOMElement->appendChild($objTable->DOMElement);

$d->appendChild($layout->DOMElement);

echo $d->saveHTML();

?>

</body>

</html>