Федеральное государственное бюджетное образовательное
учреждение высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра экономической математики, информатики и статистики
(ЭМИС)

Программирование

Методические указания
к выполнению курсового проекта для обучающихся по
направлению подготовки бакалавров
09.03.02 «Информационные системы и технологии» и
09.03.01 «Информатика и вычислительная техника»

2016

Зариковская Н.В.
Программирование. Методические указания к выполнению курсового проекта для обучающихся по направлению подготовки бакалавров 09.03.02 «Информационные системы и технологии» и 09.03.01 «Информатика и вычислительная техника». – Томск: Томский государственный университет систем управления и радиоэлектроники, 2016. – 21 с.

Методические рекомендации содержат описание целей и задач выполнения курсового проекта по дисциплине «Программирование» для обучающихся по направлению подготовки бакалавров 09.03.01 «Информатика и вычислительная техника» и дисциплине «Программирование на языках высокого уровня» для обучающихся по направлению подготовки бакалавров 09.03.02 «Информационные системы и технологии, задание на курсовой проект, требования к оформлению и защите, критерии оценки в условиях балльно-рейтинговой системы.

Содержание

1 Общие положения

Основные задачи и цели курсового проектирования - приобретение дополнительных навыков и изучение программирования, в частности алгоритмов и методов программирования сложных задач.

В пояснительной записке к курсовому проекту должна быть разработана тема в соответствии с заданием, одобренным кафедрой.

2 Общие требования к построению пояснительной записки (ПЗ)

2.1 Структура построения ПЗ

ПЗ к работе должна содержать следующие разделы:

1) титульный лист;
2) реферат;
3) задание на проектирование;
4) содержание;
5) основная часть работы;
6) заключение;
7) список литературы;
8) приложения.

Титульный лист

Титульный лист оформляется согласно ОС ТУСУР 01-2013, форма титульного листа приведена в приложении 1.

Реферат

Реферат - краткая характеристика работы с точки зрения

содержания, назначения, формы и других особенностей. Перечисляются ключевые слова работы, указывается количество страниц и приложений. Реферат размещают на отдельной странице. Заголовком служит слово «Реферат», написанное прописными буквами.

Задание на проектирование

Форма задания заполняется студентом в соответствии с полученным заданием. Форма задания приведена в приложении 2.

Содержание

Содержание включает наименования всех разделов, подразделов и пунктов, если они имеют наименование, а также список литературы и приложения с указанием номера страниц, на которых они начинаются.

Заключение

Заключение должно содержать краткие выводы по выполненной работе. Также следует указать, чему студент научился на примере решения этой задачи.

Список литературы

В список литературы входят все те, и только те источники литературы, на которые имеются ссылки в ПЗ. Список литературы оформляется в соответствии с ОС ТУСУР 01-2013.

Приложения

Приложения содержат вспомогательный материал: листинг программы и листинг тестов.

Программа должна быть самодокументированная, т.е.

- программа должна иметь простую и понятную структуру;
- в программе должны быть прокомментированы используемые структуры данных;
- для каждой подпрограммы должно быть указано, что она делает, что является входными данными и результатом,
- должен быть прокомментирован используемый алгоритм.

Основная часть курсового проекта

В основной части, должно быть решение поставленной задачи, в частности:

- постановка задачи
- анализ задачи;
- обоснование выбора алгоритма, алгоритм программы;
- обоснование выбора структур данных;
- описание алгоритма;
- реализация программы;

- описание основных процедур/функций реализованных в программе
- интерфейс программы;
- обоснование набора тестов.

Об анализе задачи

Разработка алгоритма представляет собой задачу на построение. Поэтому, как обычно в таких случаях (можно, например, вспомнить о методе решения геометрических задач на построение), необходим этап анализа задачи. Он позволяет установить, что является входом и выходом будущего алгоритма, выделить основные необходимые отношения между входными и выходными объектами и их компонентами, выделить подцели, которые нужно достичь для решения задачи, и как следствие этого, выработать подход к построению алгоритма. Результатом этапа анализа задачи должна быть спецификация алгоритма, т. е. формулировка в самом общем виде того, что (в рамках выбранного подхода) должен делать алгоритм, чтобы переработать входные данные в выходные.

Об описании алгоритма

Прежде всего, нужно иметь в виду, что такое описание предназначено не для машины, а для человека. Другими словами, речь идет не о программе, а о некотором тексте (т. е. о словесном описании), по которому можно получить представление об общей структуре разрабатываемого алгоритма, о смысле его отдельных шагов и их логической взаимосвязи. Сохранение достаточно высокого уровня описания алгоритма также облегчает его обоснование. Поэтому шаги алгоритма должны описываться в терминах тех объектов и отношений между ними, о которых идет речь в формулировке задачи. Например, для "геометрической" задачи шаги алгоритма следует описывать как действия над точками, прямыми и т. п., как проверки свойств типа принадлежности трех точек одной прямой и т. п. Но не должно быть работы с кодами этих объектов, например, с матрицей координат точек некоторого множества.

Это, конечно, не исключает использование математической и иной удобной символики. Кроме того, структура алгоритма

станет более ясной, если шаги записывать в виде, напоминающем конструкции языков программирования, такие как, например, операторы цикла и условные операторы (псевдокод).

О выборе представления данных.

Данные, обрабатываемые алгоритмом, делятся на входные, промежуточные и выходные. Специфика входных и выходных данных состоит в том, что с ними имеет дело не только алгоритм, но и пользователь программы. Поэтому различают две формы представления таких данных - внешнее и внутреннее.

Основное требование к внешнему представлению состоит в его максимальном удобстве, понятности и естественности для пользователя, чтобы он мог достаточно легко подготовить данные для ввода и оценить результат по выходным данным. Выбор внутреннего представления определяется главным образом требованием эффективности того алгоритма, который нужно построить для решения задачи.

О выборе тестов

Нужно подобрать набор тестов, достаточный для демонстрации работы программы и ее реакции на экстремальные ситуации и неправильное обращение.

2.2 Правила оформления ПЗ к курсовой работе

ПЗ пишется в редакторе MS Word шрифтом Times New Roman, размером 12, на формате А4. Нумерация страниц должна быть сквозной, первой страницей является титульный лист. Номер страницы проставляется вверху справа. Заголовки разделов пишутся прописными буквами, начиная с красной строки текста. Заголовки подразделов пишутся с абзаца строчными буквами, кроме первой прописной. В заголовке не допускаются переносы слов. Точку в конце заголовка не ставят. Если заголовок состоит из двух предложений, то их разделяют точкой.

3 Темы курсовых работ

1) Задача Прима-Краскала («жадный» алгоритм)

Дана плоская страна и в ней n городов. Нужно соединить все города телефонной связью так, чтобы общая длина телефонных линий была минимальной.

Уточнение задачи. В декартовой системе координат положение i-го города, i = 1,...,n, задано парой координат (x[i],y[i]). d[i,j] - декартово расстояние между i-ым городом и j-ым городом , j=1,...,n. В задаче речь идет о телефонной связи, т. е. подразумевается *транзитивность* связи: если i-й город связан с j-ым, а j-ый с k-ым, то i-й связан с k-ым. Подразумевается также, что телефонные линии могут разветвляться только на телефонной станции, а не в чистом поле. Наконец, требование минимальности (вместе с транзитивностью) означает, что в искомом решении не будет циклов. В терминах теории графов задача Прима-Краскала выглядит следующим образом:

Дан граф с n вершинами; длины ребер заданы матрицей (d[i,j]), i,j = 1,...,n. Найти остовное дерево минимальной длины.

Как известно, дерево с n вершинами имеет n-1 ребер. Оказывается, каждое ребро надо выбирать жадно (лишь бы ни возникали циклы).

Алгоритм Прима-Краскала (краткое описание):

в цикле n-1 раз делай:

выбрать самое короткое еще не выбранное ребро при условии, что оно не образует цикл с уже выбранными.

Выбранные таким образом ребра образуют искомое остовное дерево. Напишите программу для решения задачи Прима-Краскала.

2) Шифр Цезаря (1)

Юлий Цезарь был, якобы первым, кто придумал собственно шифр. Алфавит размещается на круге по часовой стрелке (при этом в русском алфавите, после А идет Б, а после Я - А). Для зашифровки буквы текста заменяются буквами, отстоящими по кругу на заданное число букв дальше по часовой стрелке. Если, скажем, сдвиг на 3, то вместо i-й используется (i+3)-я буква, например, вместо А пишется Г а вместо Я пишется В. При

расшифровке наоборот берут букву на заданное число букв ближе, т. е. двигаясь против часовой стрелки.

Шифр Цезаря расшифровать легко. Известны вероятности букв p[i],i = 1,2,...,n, в языке сообщения (n - число букв в алфавите). подсчитаем частоты букв f[i] в зашифрованном сообщении. Если оно не очень короткое, то f[i] должны сравнительно хорошо согласовываться с p[i]: f[i] = p[i-s] для некоторого сдвига s. Затем начнем делать перебор по сдвигам. Когда сдвиг не угадан, общее различие между p[i] и f[i+s], равное

$$\Delta(s) = \Sigma \mid p[i] - f[i+s] \mid$$

(суммирование берется по всем i от 1 до n), будет велико, а когда сдвиг угадан - мало. Минимизация Δ(s) по всем s = 1,2,...,n дает ключ к расшифровке кода Цезаря.

Напишите и испытайте программу взлома шифра Цезаря.

3) Объектно-ориентированное программирование «Солнечная система»

Тема: объектно-ориентированное программирование астрономической модели солнечной системы. Модель описывает Солнце и планеты Меркурий, Венеру, Землю, Марс и их спутники. Программа работает следующим образом: на экране изображается Солнце и планеты со своими спутниками располагаются вокруг Солнца на своих астрономических местах. Планеты начинают вращаться вокруг Солнца по своим орбитам с правильным соотношением скоростей. В то же время спутники начинают вращаться вокруг своих планет по траекториям, складывающимся из двух вращательных движений: вращение планеты вокруг Солнца и вращение спутника вокруг планеты.

Чтобы обобщить определения разных небесных, определите объект Tbody. Планеты и спутники так же, как и Солнце, - это небесные тела. Их надо определить, как объекты-наследники от Tbody. Объекты-наследники должны содержать поля:

1) текущие координаты тела;
2) центр, вокруг которого тело вращается;
3) радиус орбиты;
4) список спутников;
5) скорость вращения;

6) размер;

7) цвет тела.

Вращение как планет, так и спутников вокруг центрального тела происходит по одним и тем же законом природы. Для планет телом, вокруг которого они вращаются, является Солнце, а для каждого спутника некоторая планета. Это движение для всех небесных тел можно определить одним методом – «Вращайся!». Идея метода состоит в осуществлении движения тела наращиванием углового перемещения с шагом в 10 градусов. Перемещение каждого тела вычисляется в виде относительной величины, зависящей от значения его скорости. При каждом изменении угла вычисляются новые координаты положения тела. Каждая планета, начав вращаться должна запустить соответствующий метод вращения для своих спутников.

Относительные параметры для планет и спутников

название	**радиус**	**скорость**	**размер**
Меркурий	58	0.416	3
Венера	108	0.416	5
Земля	150	0.1	6
Марс	228	0.053	4
Луна	15	1.3	2
Фобос	7	114.4	1
Деймос	12	30.4	1

4) Перенос слов

Как показывают многочисленные эксперименты, разбиение русского слова на части для переноса с одной строки на другую с большой вероятностью выполняются правильно, если пользоваться следующими простыми приемами:

– две идущие подряд гласные можно разделить, если первой из них предшествует согласная, а за второй идет хотя бы одна буква (буква й при этом рассматривается вместе с предшествующей гласной как единое целое);

– две идущие подряд согласные можно разделить, если первой из них предшествует гласная, а в той части слова, которая идет за второй согласной, имеется хотя бы одна гласная (буквы ь,

ъ вместе с предшествующей согласной рассматриваются как единое целое);

– если не удается применить пункты 1), 2), то следует попытаться разбить слово так, чтобы первая часть содержала более чем одну букву и оканчивалась на гласную, а вторая содержала хотя бы одну гласную.

Вероятность правильного разбиения увеличивается, если предварительно воспользоваться хотя бы неполным списком приставок, содержащих гласные, и попытаться прежде всего выделить из слова такую приставку. Дано русское предложение. Выполнить разделение каждого слова из него на части для переноса.

5) Форматирование текста

Имеется текстовый файл, в конце каждой строки несколько неиспользованных позиций (т. е. правый край неровный).

Составить программу для выравнивания правого края за счет увеличения промежутков между словами. Количество пробелов в разных группах, располагающихся внутри одной и той же строки, должны отличаться не более чем на 1.

Программа должна выравнивать текст с любой длиной строки меньше <= 70.

7) Морской бой

На поле 10 на 10 позиций стоят невидимые вражеские корабли: 4 корабля по одной клетке, три корабля по 2 клетки, 2 корабля по 3 клетки, 1 корабль в 4 клетки. Позиции указываются русскими буквами от А до К (по строкам) и цифрами от 1 до 10 (по столбцам). Конфигурация и положение кораблей на поле выбираются с помощью датчика случайных чисел. Если клетка корабля угадана играющим верно, она отмечается крестиком; в противном случае точкой.

Написать программу для игры против компьютера в односторонний «морской бой». В программе реализуйте возможность показать расстановку кораблей в любой момент времени (это необходимо для отладки).

8) Построение таблицы имен (символов)

В текстовом файле записана (без ошибок) некоторая программа на языке программирования (паскаль или С). Напечатать в алфавитном порядке все различные введенные программистом идентификаторы этой программы, указав для каждого из них в возрастающем порядке номера всех строк текста программы. Учесть, что в идентификаторах одноименные прописные и строчные буквы отождествляются, что внутри литерных значений, строк – констант комментариев последовательности из букв и цифр не являются идентификаторами и что в записи вещественных чисел может встретиться буква E или e.

Для хранения идентификаторов и номеров строк используйте деревья.

9) Одномерные клеточные автоматы

Рассмотрим клетки, расположенные вдоль прямой. Каждая из клеток может иметь состояние 0 или 1. На каждом шаге по времени новое состояние клетки вычисляется из старого состояния клетки и состояний её соседей. На экране каждый горизонтальный ряд показывает состояния клеток с помощью соответствующего цвета в очередной момент времени. Временная ось идет сверху вниз и каждый ряд вычисляется на основе предыдущего ряда. Различные классы линейных клеточных автоматов могут быть определены в зависимости от того, как много различных значений (состояний) клетка может иметь (k), и как много соседей на каждой стороне клетки используются при вычислении нового состояния (r).

Правила для определения следующего состояния клетки используют сумму состояний самой клетки и клеток-соседей. Сумма состояний отображается правилом на новое состояние клетки. Если k=2, r=1, то только ближайшие соседи клетки используются. Если каждая из клеток имеет состояние 1, то максимальная сумма 1+1+1 равна 3 и сумма может меняться от 0 до 3, т. е. имеем четыре величины. Поэтому правило преобразования можно задать с помощью строки из четырех двузначных цифр. Например, правило 1010, начиная с правой цифры, задает отображение суммы 0 -> 0, 1 -> 1, 2 -> 0, 3 -> 1

(таким образом, если сумма равна 2, то новое состояние равно 0). Для k=2 и r=2 каждая клетка (на экране она изображается пикселем) новое значение получает в зависимости от значений двух соседних клеток с каждой стороны. Соответствующие правила должны иметь 6 двузначных цифр. Если k=3 и r=1, то каждая клетка может иметь значение 0, 1 или 2. Учитывается только единственный сосед с каждой стороны, поэтому результат преобразования задается правилом из 7 цифр и т. д.

Напишите программу для построения линейных клеточных автоматов следующих классов kr = 21, 31, 41, 22, 32, 42.

k	r	количество цифр в правиле	пример правила
2	1	4	1010
3	1	7	1211001
4	1	10	3311100320
2	2	6	0110110
3	2	11	21212002010
4	2	16	2300331230331001

Начальная строка задается либо случайным образом, либо пользователем.

Как ведут себя клеточные линейные автоматы? Происходит ли переход к однородному состоянию независимо от начальных данных? Существуют ли классы автоматов с локализованными стационарными или периодическими конфигурациями? Есть ли автоматы с хаотическим временным поведением? Есть ли автоматы, которые ведут себя по-разному при различных начальных данных? Необходимо ответить на эти вопросы.

Смотрите программу-аналог Fractint.

10)Программа для алгебраических вычислений (2)

Объекты, с которыми необходимо работать – это многочлены от нескольких переменных, представленных в символьном виде с вещественными коэффициентами. Многочлены должны изображаться как арифметические выражения, так умножение изображается знаком ‘*’, а возведение в степень - знаком ‘^’.

Для манипуляций с многочленами нужны некоторые команды, чтобы пользователь мог получать ответы на вопросы,

на которые не удается ответить с помощью традиционных языков программирования. Для этого вам понадобится обозначать многочлены идентификаторами. Команды выполняют некоторые операции над своими операндами и помещают результат в качестве значения некоторого имени многочлена.

Список команд.

– ввести многочлен и записать его под некоторым именем.

– образовать алгебраическую сумму (разность, произведение) двух многочленов и записать полученный многочлен под некоторым именем.

– возвести данный многочлен в целую степень и результат записать под некоторым именем.

– заменить каждое вхождение некоторой переменной в многочлене на данный многочлен и результат записать под некоторым именем.

– вычислить производную многочлена по переменной и результат записать под некоторым именем.

Напечатать данный многочлен.

Многочлен представлять в виде суммы членов, включающих только операции умножения и возведения в степень. В каждом таком одночлене все константы перемножены и образуют числовой коэффициент (первый сомножитель), переменные упорядочены по алфавиту и все степени одной переменной объединены так, что каждая переменная встречается лишь один раз. Следует приводить подобные члены, т. е. объединять одночлены, имеющие одинаковые наборы переменных и степеней, с соответствующим изменением коэффициентов. Для представления многочленов в памяти используйте списковые структуры.

11) Оптимальные стратегии для игры с угадыванием «Быки и коровы»

Правила игры «Быки и коровы» крайне просты. Один из игроков, **загадывающий**, записывает секретную комбинацию из любых четырех цифр от 1 до 6 (повторение допускаются), называемую **кодом**. Второй игрок, **отгадывающий,** пытается раскрыть код, высказывая разумные предположения, называемые

пробами. Каждая проба, как и код, представляет собой произвольную комбинацию из четырех цифр в диапазоне от 1 до 6. Отгадывающий игрок сообщает пробу загадывающему, и тот должен ответить, сколько цифр в пробе совпадает с цифрами кода, как по положению, так и по величине (это **«быки»**) и сколько из остальных цифр пробы входят в код, но стоят на другом месте (это **«коровы»**). Так, на пробу 1123 при коде 4221 будет получен ответ: «Один бык и одна корова». Тур игры продолжается до тех пор, пока отгадывающий не назовет пробу, в точности совпадающую с кодом, т. е. пока не отгадает код. Хотя здесь не последнюю роль играет везение, тем не менее, игрок, систематически делающий правильные умозаключения из получаемой информации, должен иметь лучшие результаты по итогам нескольких партий. Практически вы должны пытаться выводить из ответов на ваши пробы отрицательные следствия относительно того, какие коды невозможны; психологические тесты показывают, что для многих людей – это оказывается совсем не просто.

Написать программу, имитирующую роль загадывающего, не составляет труда. Вам же надо написать программу, чтобы машина могла, как загадывать коды, так и отгадывать их (что более сложно). Реализуйте стратегию отгадывания Боба Кули, заключающуюся в следующем.

Центральное место в стратегии занимает идея **пространства решений**. Начальное пространство решений P(0) состоит из всех возможных кодов (и имеет, следовательно 6*6*6*6 элементов); после i-й пробы G(i) пространство P(i) состоит из всех тех членов пространства P(i-1), которые не опровергающая ответом R(i). Иными словами, пространство P(i) - это множество тех комбинаций, которые все еще могут быть кодом; задача отгадывающего - свести пространство к одному элементу.

Пробой G(1) пусть будет любая случайно выбранная комбинация с одной повторяющейся цифрой, например 4311, 6552 или 1335. Выполните эту пробу и постройте пространство P(1) на основе ответа R(1). Новая проба G(i+1) ищется по пространству P(i), i больше или равно 1, путем поочередного сравнения всех комбинаций C из P(i) с пробой G(i). В качестве следующей пробы выбирается *наименее* похожая на G(i)

комбинация С. Мерой сходства служит число точных совпадений, а в случае равенства - число коров. Так, среди трех комбинаций 2641, 2356 и 1345 наиболее похожей на 2345 будет 1345, а 2641 - наименее похожей. Если имеется несколько наименее похожих комбинаций, то можно выбрать любую кандидатуру случайным образом. Тур прекращается, когда будет получен ответ «4 быка», и, разумеется, в случае пространства из одного элемента в качестве следующей пробы всегда надо брать этот элемент. Как показывают эксперименты, размеры пространства решений сокращаются после пробы примерно в 4 раза и никогда не требуется более шести проб.

12) Поиск узоров из простых чисел (3)

Основная трудность в теории простых чисел обусловлена тем, что простые числа распределены среди всех целых чисел по закону, носящему заведомо не вероятностный характер, но тем не менее упорно не поддающемуся всем попыткам установить его. Закономерности появляются, когда целые числа отображаются на плоскость (или в пространство). Однажды математику Станиславу М. Уламу пришлось присутствовать на одном очень длинном и очень скучном, по его словам, докладе. Чтобы как-то развлечься, он начертил на листке бумаги вертикальные и горизонтальные линии и хотел было заняться составлением шахматных этюдов, но потом передумал и начал нумеровать пересечения, поставив в центре 1 и двигаясь по спирали против часовой стрелки. Без всякой задней мысли он обводил все простые числа кружками. Вскоре, к его удивлению, кружки с поразительным упорством стали выстраиваться вдоль прямых. Так выглядит спираль со ста первыми числами (от 1 до 100):

100	99	98	***97***	96	95	94	93	92	91
65	64	63	62	***61***	60	***59***	58	57	90
66	*37*	36	35	34	33	32	***31***	56	***89***
67	38	***17***	16	15	14	***13***	30	55	88
68	39	18	***5***	4	***3***	12	***29***	54	87
69	40	***19***	6	***1***	***2***	***11***	28	***53***	86
70	***41***	20	***7***	8	9	10	27	52	85
71	42	21	22	***23***	24	25	26	51	84

72	***43***	44	45	46	***47***	48	49	50	***83***
73	74	75	76	77	78	***79***	80	81	82

Напишите программу, которая отображает целые числа на плоскость некоторым регулярным образом и отмечает на рисунке места, где находятся простые числа. Выведите формулы, описывающие прямые линии на вашем рисунке, и напечатайте те из них, которые особенно изобилуют простыми числами; печатайте также долю простых чисел на этих прямых. Обеспечьте высокую эффективность ваших программ проверки целых чисел на простоту, так чтобы вам хватило времени для анализа весьма отдаленных отрезков натурального ряда.

13)«Сбей самолет»

Напишите программу для аркадной игры. По экрану летят вражеские самолеты. Цель – сбить их. Пусковая установка находиться внизу экрана. Пусковую установку можно перемещать в горизонтальном направлении вперед и назад.

14 «Советник по транспорту»

Некоторая железнодорожная компания обслуживает n станций S1,S2,...,Sn. Она предлагает улучшить информационное обслуживание клиентов с помощью информационных терминалов, управляемых вычислительной машиной. Клиент набирает название своей станции направления Sa и станции назначения Sd, и ему должна выдаваться схема расписаний поездов с минимальным общим временем поездки.

Разработайте программу для вычисления нужной информации. Пусть расписание (представляющее собой ваш банк данных) изображено соответствующей структурой данных, содержащей время отправления и прибытия всех имеющихся поездов. Разумеется, не все станции связаны непрерывными линиями.

15 «Обратный тетрис»

Тетрис наоборот. Игрок выбирает фигуру и бросает, компьютер пытается установить ее в стакан. Цель игры – «завалить» компьютер.

16 Эмулятор Total Commander.

Реализовать эмуляцию перемещения по некоторой файловой системе. Реализовать эмуляцию большинства функциональных клавиш Total Commander, включая сочетание с клавишами «Ctrl», «Alt» и «Shift».

17 Касса аэрофлота

Расписание: номер рейса, маршрут, пункты промежуточной посадки, время отправления, дни отлета. Количество свободных мест на каждом рейсе. Выбор ближайшего рейса до заданного рейса (при наличии свободных мест), оформление заданного числа билетов по согласованию с пассажиром (с уменьшением числа свободных мест), оформление посадочной ведомости.

18 База данных «классный журнал»

Создается программа для ведения базы данных по учету успеваемости студентов. Эта программа должна выполнять следующие функции:

- создает базу данных;
- производит корректировку записей;
- вывод базы данных на печать и экран.

Структура записи содержит:

- фамилию;
- имя;
- отчество;
- год рождения;
- оценка.

19 «Тир слов»

Составить программу для заучивания слов иностранного языка. Предусмотреть режим обучения.

20 Круги на воде

Экран изображает бассейн с водой, в который бросили камень (в заданных координатах). От камня пошли круги, которые, дойдя до стенок бассейна, отражаются от них.

Реализовать эту динамическую картину (в графике или псевдографике).

21 Игра «Нажми Клавишу»

Случайно выводится изображение клавиши и необходимо как можно быстрее нажать соответствующую клавишу на клавиатуре. При этом программа измеряет время между выводом изображения и соответствующим нажатием. Перед началом игры вводится имя игрока и определяется количество попыток. Результаты игры вносятся в специальную таблицу <имя игрока> <результат>

ПРИЛОЖЕНИЕ 1

ФОРМА ТИТУЛЬНОГО ЛИСТА К ПОЯСНИТЕЛЬНОЙ ЗАПИСКЕ ПО КУРСОВОМУ ПРОЕКТУ

Федеральное государственное бюджетное образовательное учреждение высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра экономической математики, информатики и статистики (ЭМИС)

Игра «Жизнь»
Пояснительная записка к курсовому проекту по дисциплине «Программирование»

Выполнил
Студент гр. _____
_____С. С. Лавров
_____ (дата)

Проверил
к.ф.-м.н., доцент кафедры ЭМИС
__ Н.В. Зариковская
_____ (дата)

2016

ПРИЛОЖЕНИЕ 2

ФОРМА ЗАДАНИЯ ДЛЯ КУРСОВОГО ПРОЕКТА

Федеральное государственное бюджетное образовательное учреждение высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра экономической математики, информатики и статистики (ЭМИС)

УТВЕРЖДАЮ
заведующий кафедрой ЭМИС
д.ф.м.н., профессор И.Г.Боровской
_____

ЗАДАНИЕ
на курсовое проектирование по дисциплине
«Программирование»

студенту_____
группа_____ факультет вычислительных систем

1. Тема проекта: Игра «Жизнь»
2. Срок сдачи студентом законченной работы _____
3. Исходные данные к проекту (здесь должен быть текст задания) _____
_____
_____
_____
_____
_____
4. Дата выдачи задания: _____ г.

Задание принял к исполнению
_____(ФИО)