

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 28.01.2021 17:36:57
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРАЗОВАНИЯ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
Юго-Западный государственный университет
(ЮЗГУ)

Кафедра вычислительной техники



Вычислительные системы повышенной надёжности

Методические рекомендации по выполнению
лабораторных работ №1-4

по курсу “Вычислительные системы повышенной надёжности”
для студентов направления подготовки 09.03.01

Курск 2017

УДК 621.(076.1)

Составители: С.А. Дюбрюкс

Рецензент

Доктор технических наук, доцент Д.Б. Борзов

Вычислительные системы повышенной надёжности: методические рекомендации по выполнению лабораторных работ №1-4 по дисциплине «Вычислительные системы повышенной надёжности» / Юго-Зап. гос. ун-т; сост.: С.А. Дюбрюкс.- Курск, 2017. 17 с.: ил. 2.

Методические рекомендации содержат основные определения теории надёжности, основные причины и последствия отказов ПО, краткое описание аналитических моделей надёжности программ для ПЭВМ. Лабораторные работы посвящены практическому освоению методов расчёта надёжности вычислительных систем с использованием различных моделей: аналитической модели с дискретным увеличением времени наработки на отказ, экспоненциальной модели, модели Джелинского-Моранды, модели Маркова.

Предназначены для студентов направления подготовки 09.03.01 очной и заочной форм обучения «Информатика и вычислительная техника».

Текст печатается в авторской редакции

Подписано в печать _____ . Формат 60x84 1/16.

Усл. печ. л. Уч. – изд.л. Тираж 30 экз. Заказ. Бесплатно.

Юго-Западный государственный университет
305040, Курск, ул. 50 лет Октября, 94.

Содержание

Содержание	3
Введение	4
Цель работы	5
Основные теоретические сведения.	5
Лабораторная работа №1	7
Лабораторная работа №2	9
Лабораторная работа №3	11
Лабораторная работа №4	13
Оформление отчёта	16
Литература	17

Введение

Проблема надежности является одной из важнейших для современных сложных информационных систем (ИС), поскольку опыт их создания и применения показал, что **сбои и отказы функционирования ИС обусловлены их внутренними дефектами** и отражаются большим ущербом.

Для обеспечения надежности ИС в конкретных проектах применяются современные автоматизированные технологии и инструментальные средства, обеспечивающие предупреждение или исключение большинства видов дефектов и ошибок при создании и модификации систем.

При этом особую важность имеют методы и средства определения и прогнозирования характеристик надежности в жизненном цикле ИС с учетом их функционального назначения, сложности, структурного построения и технологии разработки. К их числу относятся методы математического моделирования, на практическое освоение которых и ориентированы данные лабораторные работы.

Цель работы. Лабораторные работы по учебной дисциплине имеют своей целью:

- закрепление, углубление и расширение теоретических знаний студентов в процессе решения конкретных задач по расчетам надежности ИС;

- развитие у студентов профессиональных навыков, а также практическое овладение методами расчета надежности сложных информационных систем;

- закрепление умений и навыков использования систем автоматизированного решения математических задач.

Выполнению работы предшествует опрос по теории работы и устное собеседование по методике ее выполнения.

Каждая работа оформляется студентом в виде отчета, который обязательно включает раздел, где анализируется и объясняется вся полученная информация.

Итогом работы является ее защита. Защита проводится устно, но обязательно индивидуально.

Основные теоретические сведения.

Характеристики надежности. *Надежность ИС* – это свойство сохранять заданные характеристики при определенных условиях эксплуатации. Надежность ИС определяется безотказностью и восстанавливаемостью соответствующего программного обеспечения (ПО).

Безотказность ПО – его свойство сохранять работоспособность в процессе обработки информации, которую можно определить вероятностью работы без отказов при определенных условиях внешней среды в течение заданного периода наблюдения.

Отказ программы – это недопустимое отклонение характеристик процесса функционирования программы от требуемых. Определенные условия внешней среды – это совокупность входных данных и состояния вычислительной системы.

Заданный период наблюдения обычно соответствует необходимому числу прогонов программы для решения задачи.

Безотказность ПО можно охарактеризовать также *средним временем между двумя отказами* в процессе выполнения программы. С точки зрения надежности принципиальное отличие программных средств от аппаратных состоит в том, что программы не изнашиваются и не подвержены физическому старению в процессе работы.

Поэтому характеристики надежности ПО зависят от тщательности разработки и отладки, а также от условий хранения носителей программ. Безотказность ПО определяется его корректностью, а значит целиком зависит от наличия в нем ошибок, внесенных на этапе создания и хранения, в то время как безотказность аппаратных средств зависит в основном от случайных отказов, связанных с физическими изменениями параметров

элементов. Вид обрабатываемых данных не влияет на аппаратуру, но может привести к отказам ПО.

Интенсивность отказов ПО с течением времени уменьшается, так как в процессе эксплуатации обнаруживаются и устраняются его скрытые ошибки.

Восстанавливаемость программы может быть оценена сравнительной продолжительностью устранения ошибки в программе и восстановления ее работоспособности. Восстановление после отказа может заключаться в корректировке текста программы, исправлении данных, внесении изменений в организацию вычислительного процесса. Восстанавливаемость зависит от сложности структуры комплекса программ, от алгоритмического языка, от качества документации и т. д. Можно также говорить об *устойчивости* ПО, понимая под этим способность ограничивать последствия собственных ошибок и противостоять неблагоприятным условиям внешней среды. Устойчивость ПО может быть повышена с помощью разных форм структурной, информационной и временной избыточности, позволяющей иметь дублирующие модули программ, альтернативные пути для решения одной задачи, позволяющих осуществлять контроль за процессом исполнения программ (защелкивание, блокировка и т. д.).

Причины отказов ПО:

- ошибки, скрытые в самой программе;
- искажение входной информации, подлежащей обработке;
- неверные действия пользователя (могут быть связаны с некорректной документацией);
- неисправность аппаратуры, на которой реализуется вычислительный процесс.

Последствия и признаки появления ошибок в ПО. В зависимости от степени серьезности последствий ошибок в ПО отклонения выполнения программой заданных функций можно охарактеризовать следующим образом: полное прекращение выполнения функций на длительное или неопределенное время или кратковременное прекращение хода вычислительного процесса.

Симптомы проявления ошибки в программе:

- преждевременное окончание программы;
- увеличение времени выполнения программы (защелкивание);
- потери или искажение накопленных данных;
- нарушение порядка вызова отдельных программ.

Для устранения ошибок программы необходимо предусмотреть специальные средства диагностики типа кодов завершения, вводить в ПО контрольные точки, обеспечить возможность рестарта с контрольных точек.

Аналитические модели надежности программ. Аналитические модели дают возможность исследовать закономерности появления ошибок ПО, а также прогнозировать надежность его эксплуатации ПО. Модели строятся в предположении, что появление ошибок является случайным событием и имеет вероятностный характер.

Функцию надежности $P(t)$ можно определить как вероятность того, что ошибка появится в программе не ранее чем через время t . Обратная величина $Q(t)$ – вероятность того, что ошибка произойдет за время t .

Из этих характеристик можно вывести величину интенсивности отказов $\lambda(t)$, которая будет определяться плотностью вероятности возникновения отказа [1]:

$$\lambda(t) = -\frac{dP(t)}{dt} / P(t).$$

Для ПО характерно ступенчатое изменение $\lambda(t)$. Поэтому наиболее простая модель надежности ПО – это дискретная модель

$$\lambda(t) = K(M - i(t)) = \lambda_i,$$

где M – постоянная, характеризующая начальное число ошибок;

$i(t)$ – число отказов, устраняемое в момент времени t ;

K – эмпирический коэффициент, зависящий от характеристик системы.

Эти параметры можно найти на основании последовательности наблюдений интервалов между обнаружением ошибок по методу **максимального правдоподобия**, с помощью которого производится точечная оценка неизвестных параметров априорно известного закона распределения случайной величины. Суть этого метода – оценивание неизвестного параметра путем максимизации **функции правдоподобия** (на практике используется логарифмическая функция правдоподобия).

Если $i(t)$ постоянна – получаем простую линейную модель.

Лабораторная работа №1.

Расчет надежности ИС с использованием аналитической модели с дискретным увеличением времени наработки на отказ [1, 3, 5].

Цель работы: практическое освоение метода расчета надежности на основе использования модели с дискретным увеличением времени наработки на отказ.

Предмет и содержание работы. Модель надежности программ с дискретным увеличением времени наработки на отказ основана на предположении, что устранение ошибки приводит к увеличению времени наработки на отказ на некоторую случайную величину.

$$T_m = \frac{m(m+1)}{2} M[\Delta T],$$

где T_m – время наработки на отказ до возникновения m -го отказа;

$M[\Delta T]$ – математическое ожидание времени между двумя отказами;

$$M[\Delta T] = \frac{\sum_{i=1}^m t_i}{m}.$$

Для того чтобы подсчитать, какое время тестирования необходимо для обеспечения некоторой наработки на отказ, нужно сначала определить математическое ожидание времени между двумя отказами по существующим данным. Далее надо последовательно высчитывать по формуле время наработки на отказ до тех пор, пока оно не достигнет нужного значения. Просчитанные данные лучше заносить в таблицу. После достижения нужного значения все времена нужно сложить, получив в результате искомое время тестирования.

Задание: определить время тестирования, необходимое для достижения указанного времени наработки на отказ. Исходные данные: время первого отказа, время второго отказа, время наработки на отказ.

Оборудование, технические средства, инструмент: рабочая станция с установленной ОС, подключенная к Интернету, обозреватель Интернета (для доступа к Wolfram Alpha), Microsoft Word (текстовый процессор Open Office). В случае отсутствия доступа в Интернет задача может быть решена непосредственно на рабочей станции путем использования возможностей доступной на ней системы программирования.

Порядок (последовательность) выполнения работы:

- включение рабочей станции;
- проверка наличия связи с Интернетом;
- установление связи с Wolfram Alpha, выбор исходных данных для решения задачи;
- разработка алгоритма решения задачи;
- решение поставленной задачи;
- подготовка данных для отчета;
- сохранение этих данных;
- выключение компьютера.

Контрольные вопросы

1. Понятие надежности ИС.
2. Краткая характеристика аналитических моделей надежности.
3. Особенности аналитической модели с дискретным увеличением времени наработки на отказ.

Задания

1. Изучить свойства надежности ИС.
2. Изучить классификацию моделей надежности ИС.
3. Изучить особенности модели надежности программ с дискретным увеличением времени наработки на отказ.
4. Определить время тестирования, необходимое для достижения указанного времени наработки на отказ.

Исходные данные: время первого отказа, время второго отказа, время наработки на отказ.

5. Подготовить данные для отчета.

Лабораторная работа №2.

Расчет надежности ИС с использованием простой экспоненциальной модели [2, 3, 5].

Цель работы: практическое освоение метода расчета надежности на основе использования экспоненциальной модели.

Предмет и содержание работы. Экспоненциальная модель надежности ПО основана на предположении об экспоненциальном характере изменения числа ошибок во времени. В этой модели прогнозируется надежность программы на основе данных, полученных во время тестирования. В модели вводится суммарное время функционирования τ , которое отсчитывается с момента начала тестирования до контрольного момента оценки надежности.

Предполагается, что число ошибок в программе в каждый момент времени имеет пуассоновское распределение, а временной интервал между двумя ошибками распределен по экспоненциальному закону. Параметр этого распределения изменяется после распределения очередной ошибки. *Интенсивность отказов считается непрерывной функцией, пропорциональной числу оставшихся ошибок.*

Число ошибок m , выявленных за время τ будет определяться зависимостью (рисунок 1):

$$m = M(1 - \exp(-K\tau)), \quad (1)$$

где K – коэффициент пропорциональности, учитывающий быстродействие ЭВМ и число команд в программе.

M – число ошибок, имевшееся перед фазой тестирования.

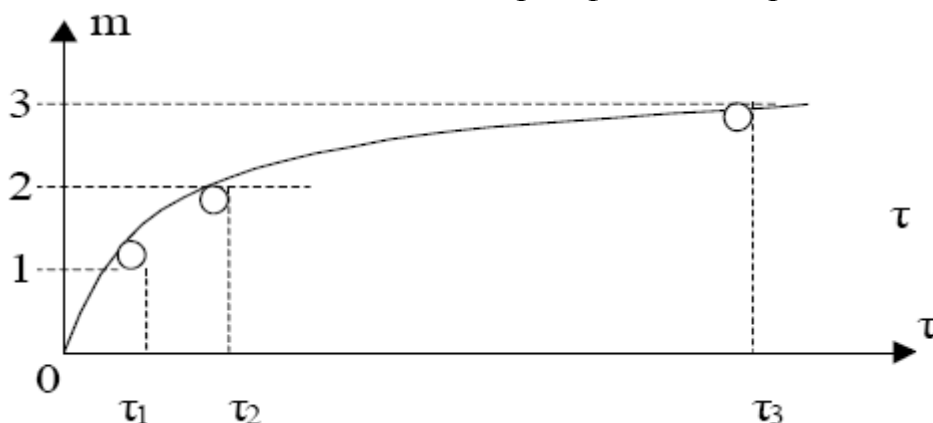


Рисунок 1 – Экспоненциальная модель надежности ПО.

Пусть среднее время наработки на отказ

$$T_{0\text{ср}} = 1 / \lambda(\tau).$$

Если ввести величину $T'_{0\text{ср}}$ – начальное значение средней времени наработки на отказ, то

$$T_{0\text{ср}} = T'_{0\text{ср}} \exp\left(\frac{\tau}{MT'_{0\text{ср}}}\right), \quad (2)$$

$$T'_{0\text{ср}} = \frac{1}{KM}.$$

где

Для того чтобы определить, какое время тестирования необходимо для обеспечения указанного времени наработки, необходимо рассчитать параметры закона распределения K и M . Зная эти параметры можно рассчитать прогнозируемое количество в любой момент времени. Следовательно, чтобы определить, какое время тестирования необходимо для определения необходимого времени наработки, нужно итеративно высчитывать промежутки времени между ошибками по формуле (2) до тех пор, пока необходимое время наработки не будет достигнуто. Другим способом решения этой задачи является аппроксимация функции вида (1) по заданным точкам.

Оборудование, технические средства, инструмент: рабочая станция с установленной ОС, подключенная к Интернету, обозреватель Интернета (для доступа к Wolfram Alpha), Microsoft Word (текстовый процессор Open Office). В случае отсутствия доступа в Интернет задача может быть решена непосредственно на рабочей станции путем использования возможностей доступной на ней системы программирования.

Порядок (последовательность) выполнения работы:

- включение рабочей станции;
- проверка наличия связи с Интернетом;
- установление связи с Wolfram Alpha;
- выбор исходных данных для решения задачи;
- разработка алгоритма решения задачи;
- решение поставленной задачи;
- подготовка данных для отчета;
- сохранение этих данных;
- выключение компьютера.

Контрольные вопросы

1. Понятие Web-страницы.
2. Основные показатели надежности ИС.
3. Понятия повреждения и отказа ИС.
4. Особенности экспоненциальной модели расчета надежности.

Задания

1. Изучить возможности интерфейса браузера.
2. Изучить особенности экспоненциальной модели надежности программ.
3. Уточнить особенности метода максимального правдоподобия применительно к рассматриваемой модели надежности.
4. Определить время тестирования, необходимое для достижения указанного времени наработки на отказ.

Исходные данные (время первого отказа, время второго отказа, время наработки на отказ) совпадают с данными, использованными при выполнении лабораторной работы № 1.

5. Сравнить результаты расчетов, полученных при выполнении данной лабораторной работы и лабораторной работы № 1.

6. Подготовить данные для отчета.

Лабораторная работа №3.

Расчет надежности ИС с использованием модели Джелинского–Моранды [2, 3, 5].

Предмет и содержание работы. Модель Джелинского–Моранды строится на основе ряда допущений:

- 1) интенсивность обнаружения ошибок $\lambda(t)$ пропорциональна текущему числу ошибок в программе, т. е. числу оставшихся ошибок;
- 2) все ошибки одинаково вероятны и их появления независимы;
- 3) каждая ошибка имеет один и тот же порядок серьезности;
- 4) время до следующего отказа (ошибки) распределено экспоненциально;
- 5) ПО функционирует в среде, близкой к реальной;
- 6) ошибки постоянно корректируются без внесения в ПО новых;
- 7) $\lambda(t) = \text{const}$ в интервале между двумя соседними ошибками.

В соответствии с этими допущениями интенсивность возникновения (обнаружения) ошибок в ПО можно представить в виде:

$$\lambda(t) = K [B - (i - 1)],$$

где t – произвольное время между обнаружением $(i - 1)$ и i -ой ошибок;

K – неизвестный коэффициент;

B – неизвестное общее число ошибок в ПО.

Если за время t было обнаружено $(i - 1)$ ошибок, то в ПО еще осталось $[B - (i - 1)]$ ошибок. Полагая $X_i = t_i - t_{i-1}$, где i изменяется от 1 до n , и учитывая допущения о том, что $\lambda(t) = \text{const}$ в интервале между $(i - 1)$ -й и i -й ошибками, можно считать, что X_i имеют экспоненциальное распределение.

Для получения оценок B и K используют следующие отношения.

Получение \hat{B} :

$$\hat{B} = m - 1,$$

где $m \geq n - 1$ – число прогнозируемых (пока не обнаруженных) ошибок.

Находят значения функций:

$$f_n(m) = \sum_{i=1}^n \frac{1}{m-i},$$

$$g_n(m, A) = \frac{n}{m-A},$$

$$A = \frac{\sum_{i=1}^n i \cdot X_i}{\sum_{i=1}^n X_i},$$

затем вычисляют значения разностей $f_n(m) - g_n(m, A)$, и, анализируя их (находят минимальную разность), определяют значение m как наилучшее целочисленное решение уравнения:

$$f_n(\hat{B}+1) = g_n(\hat{B}+1, A), \text{ при условии, что } A \geq (n+1)/2.$$

Получение \hat{K} :

$$\hat{K} = \frac{n}{(\hat{B}+1) \sum_{i=1}^n X_i - \sum_{i=1}^n i \cdot X_i}.$$

Определение оценок:

- интенсивность возникновения ошибок в ИС после того, как в ней уже обнаружена $(i-1)$ -я ошибка:

$$\hat{\lambda}(t) = \hat{K} [\hat{B} - (i-1)];$$

- среднее время до появления $(i+1)$ -й ошибки:

$$\hat{X}_{i+1} = \frac{1}{\hat{K} (\hat{B} - n)};$$

- время до окончания тестирования:

$$\hat{t}_K = \sum_{i=n}^{\hat{B}} X_i = \frac{1}{\hat{K}} \sum_{i=1}^{\hat{B}-n} \frac{1}{i}.$$

Оборудование, технические средства, инструмент: рабочая станция с установленной ОС, подключенная к Интернету, обозреватель Интернета (для доступа к Wolfram Alpha), Microsoft Word (текстовый процессор Open Office). В случае отсутствия доступа в Интернет задача может быть решена непосредственно на рабочей станции путем использования возможностей доступной на ней системы программирования.

Порядок (последовательность) выполнения работы:

- включение рабочей станции;
- проверка наличия связи с Интернетом;
- установление связи с Wolfram Alpha;
- выбор исходных данных для решения задачи;

- решение поставленной задачи;
- подготовка данных для отчета;
- сохранение этих данных;
- выключение компьютера.

Контрольные вопросы

1. Доминирующий фактор надежности ИС.
2. Виды избыточностей.
3. Особенности модели Джелинского–Моранды. Основное отличие данной модели от простой экспоненциальной модели.

Задания

1. Изучить особенности модели Джелинского–Моранды.
2. Определить: интенсивность возникновения ошибок в ИС после того, как в ней уже обнаружена $(i - 1)$ -я ошибка; среднее время до появления $(i + 1)$ -й ошибки; время до окончания тестирования. Исходные данные сведены в таблицу в виде интервалов времени X_i (дн.) между соседними ошибками (i – номер ошибки). Количество дней, в течение которых производилось тестирование – N , количество выявленных ошибок – m .
3. Подготовить данные для отчета.

Лабораторная работа №4.

Расчет надежности ИС с использованием марковской модели [1, 4, 5].

Цель работы: практическое освоение метода расчета надежности на основе использования марковской модели.

Предмет и содержание работы. Процесс поведения системы, при котором вероятность ее перехода в любое возможное состояние в каждый момент времени определяется только ее состоянием в предыдущий момент времени и не зависит от более ранней предыстории называется *марковским*. Такого рода процессы широко применяются для описания функционирования сложных ИС, состоящих из некоторого количества программных компонент – модулей.

В соответствии с этим выбор следующего модуля ИС зависит только от модуля, выполняемого в данный момент и не зависит от предыстории. Структуру управления системой по марковской модели можно представить в виде направленного графа (рисунок 2).

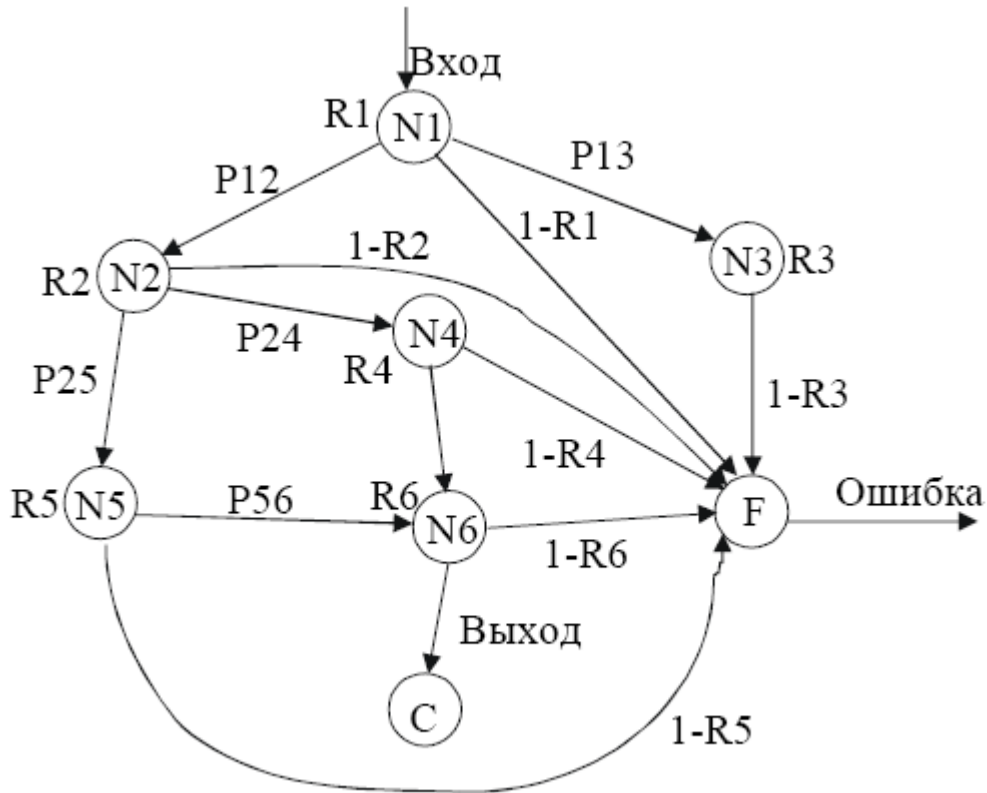


Рисунок 2 Граф надежности программного комплекса.

Каждое состояние системы может быть оценено вероятностью безотказной работы i -го модуля R_i . Вероятность перехода от i -го модуля к j -му представлена величиной P_{ij} .

Вероятность отказа равна $1 - P_{ij}$. Таким образом можно составить матрицу переходных вероятностей:

$$P = \begin{pmatrix} 0 & R_1 P_{12} & R_1 P_{13} & \dots & R_{ij} & 0 & 1 - R_1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & R_i P_{ij} & \dots & \dots & 0 & 1 - R_{n-1} \\ 0 & \dots & R_{n-1} P_{(n-1)j} & \dots & \dots & R_n & 1 - R_n \\ 0 & \dots & \dots & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Для каждого целого $k > 0$ $P^k(i, j)$ будет определять вероятность перехода из состояния i в состояние j за k шагов.

Таким образом, строки матрицы соответствуют «старым», а столбцы – «новым» состояниям системы. Очевидно, сумма элементов любой строки матрицы равна 1.

Тогда матрица

$$T = I + P + P^2 + \dots = \sum_{k=0}^{\infty} P^k$$

будет определять вероятность перехода из одного состояния в другое за произвольное число шагов (здесь I – единичная матрица).

Вычеркнув из матрицы P две последние строки и два последних столбца, соответствующие успеху C и отказу F , получаем матрицу Q .

$$S = I + Q + Q^2 + \dots = \sum_{k=0}^{\infty} Q^k.$$

Пусть Положив $W = I - Q$, имеем:

$$S = W^{-1} = (I - Q)^{-1}, \quad \text{откуда надежность информационной системы:}$$

$$R = S(1, n) R_n.$$

Оборудование, технические средства, инструмент: рабочая станция с установленной ОС, подключенная к Интернету, обозреватель Интернета (для доступа к Wolfram Alpha), MicroSoft Word (текстовый процессор Open Office). В случае отсутствия доступа в Интернет задача может быть решена непосредственно на рабочей станции путем использования возможностей доступной на ней системы программирования.

Порядок (последовательность) выполнения работы:

- включение рабочей станции;
- проверка наличия связи с Интернетом;
- установление связи с Wolfram Alpha;
- разработка алгоритма решения задачи (если нет доступа к Wolfram Alpha);
- решение поставленной задачи;
- подготовка данных для отчета;
- сохранение этих данных;
- выключение компьютера.

Контрольные вопросы

1. Понятие и виды избыточностей в ИС.
2. Понятия исправности, неисправности, работоспособности, неработоспособности ИС.
3. Особенности *марковской* модели расчета надежности.

Задания

1. Изучить особенности марковской модели надежности программ.
2. Выполнить расчет надежности с использованием представленного на рисунке 2 графа надежности со следующими параметрами: $R1 = 0.9$; $R2 = 0.2$; $R3 = 0.94$; $R4 = 0.85$; $R5 = 0.95$; $R6 = 0.93$; $P12 = 0.7$; $P13 = 0.2$; $P25 = 0.6$; $P24 = 0.29$; $P56 = 0.95$.
3. Сравнить результаты расчетов, полученных при выполнении данной лабораторной работы и лабораторной работы 1.
4. Подготовить данные для отчета.

3 ОФОРМЛЕНИЕ ОТЧЕТА

После выполнения каждой из лабораторных работ студентом оформляется отчет и представляется преподавателю для проверки с последующей защитой (выполнение отчета и защита работы проводится каждым студентом индивидуально).

Работа оформляется в последовательности, приведенной в методических указаниях.

На первой странице пишется заглавие, указывается цель и объем работы в часах, Ф.И.О. студента, группа, дата выполнения.

Текст работы оформляется на ПЭВМ шрифтом Times New Roman с использованием средств текстового процессора и выводится на принтер на листах формата А4 (210 * 297 мм) с соблюдением ГОСТ 2.105-95, ГОСТ 8.417-2002 и ГОСТ 7.1-2003.

В отчете по проделанной работе должны быть включены следующие структурные элементы:

- а) титульный лист;
- б) цель работы;
- в) основная часть, содержащая постановку задачи и полученные результаты, а также отражающая процесс выполнения работы;
- г) выводы.

Перенос слов на титульном листе и в заголовках текста не разрешается. Точка в конце заголовка не ставится.

Защита лабораторных работ осуществляется по результатам выполненного задания, в процессе защиты выполняется дополнительная проверка (с использованием контрольных вопросов) усвоения студентом материала.

ЛИТЕРАТУРА

1. Громов, Ю.Ю. Надежность информационных систем: учебное пособие / Ю.Ю. Громов, О.Г. Иванова, Н.Г. Мосягина, К.А. Набатов – Тамбов: Изд-во ГОУ ВПО ТГТУ, 2010. – 160 с. Информационная система «Единое окно доступа к образовательным ресурсам» <http://window.edu.ru/resource/090/73090>
2. Лаврищева, Е. Методы и средства инженерии программного обеспечения [Электронный ресурс] / Е. Лаврищева, В. Петрухин. – М.: Интуит.ру, 2008. URL: http://www.intuit.ru/studies/professional_retraining/944/courses/237/info (Интернет-университет информационных технологий).
3. Котляров, В. Основы тестирования программного обеспечения [Электронный ресурс] / В. Котляров. – М.: Интуит.ру, 2005. – URL: <http://www.intuit.ru/studies/courses/48/48/info> (Интернет-университет информационных технологий).
4. Нечаев, Д.Ю. Надежность информационных систем [Электронный ресурс] / Д.Ю. Нечаев, Ю.В. Чекмарев. – М.: «ДМК Пресс», 2012. – 64 с. http://e.lanbook.com/books/element.php?p11_cid=25&p11_id=3030 (ЭБС «Лань»).
5. Острейковский, В.А. Теория надежности: учебник для вузов / В.А. Острейковский. – М.: Абрис, 2012. – 463 с. <http://www.biblioclub.ru/index.php?page=author&id=33844> ЭБС«Университетская библиотека ONLINE»).