

УТВЕРЖДАЮ:

Проректор по учебной работе

_____ / О.В. Юсупова

" ____ " _____ 20__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.В.01.02 «Моделирование»

Код и направление подготовки (специальность)	09.03.04 Программная инженерия
Направленность (профиль)	Программная инженерия
Квалификация	Бакалавр
Форма обучения	Заочная
Год начала подготовки	2022
Институт / факультет	Институт автоматике и информационных технологий
Выпускающая кафедра	кафедра "Вычислительная техника"
Кафедра-разработчик	кафедра "Вычислительная техника"
Объем дисциплины, ч. / з.е.	144 / 4
Форма контроля (промежуточная аттестация)	Экзамен

Б1.В.01.02 «Моделирование»

Рабочая программа дисциплины разработана в соответствии с требованиями ФГОС ВО по направлению подготовки (специальности) **09.03.04 Программная инженерия**, утвержденного приказом Министерства образования и науки РФ от № 920 от 19.09.2017 и соответствующего учебного плана.

Разработчик РПД:

Профессор, доктор
технических наук, доцент

(должность, степень, ученое звание)

Б.В. Мартемьянов

(ФИО)

Заведующий кафедрой

А.В. Иващенко, доктор
технических наук, профессор

(ФИО, степень, ученое звание)

СОГЛАСОВАНО:

Председатель методического совета
факультета / института (или учебно-
методической комиссии)

Я.Г. Стельмах, кандидат
педагогических наук

(ФИО, степень, ученое звание)

Руководитель образовательной
программы

А.В. Иващенко, доктор
технических наук, профессор

(ФИО, степень, ученое звание)

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы	4
2. Место дисциплины (модуля) в структуре образовательной программы	5
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся	6
4. Содержание дисциплины (модуля), структурированное по темам (разделам), с указанием отведенного на них количества академических часов и видов учебных занятий	6
4.1 Содержание лекционных занятий	6
4.2 Содержание лабораторных занятий	7
4.3 Содержание практических занятий	8
4.4. Содержание самостоятельной работы	8
5. Перечень учебной литературы и учебно-методического обеспечения по дисциплине (модулю)	10
6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения	11
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных, информационно-справочных систем	11
8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)	12
9. Методические материалы	13
10. Фонд оценочных средств по дисциплине (модулю)	14

**1. Перечень планируемых результатов обучения по дисциплине (модулю),
соотнесенных с планируемыми результатами освоения образовательной
программы**

Наименование категории (группы) компетенций	Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения (знать, уметь, владеть, соотнесенные с индикаторами достижения компетенции)
Профессиональные компетенции			
Не предусмотрено	ПК-1 Владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения	ПК-1.1 Знает основы моделирования и формальные методы конструирования программного обеспечения	Знать 06.001 В - Проверка работоспособности и рефакторинг программного обеспечения: В/04.4 / 4 - Рефакторинг и оптимизация программного кода. 06.001 D - Разработка требований и проектирование программного обеспечения: D/03.6 / 6 - Проектирование программного обеспечения.
		ПК-1.2 Умеет использовать формальные методы конструирования программного обеспечения	Уметь 06.001 В - Проверять работоспособность и выполнять рефакторинг программного обеспечения: В/04.4 / 4 - Выполнять рефакторинг и оптимизацию программного кода. 06.001 D - Выполнять разработку требований и проектирование программного обеспечения: D/03.6 / 6 - Выполнять проектирование программного обеспечения.
		ПК-1.3 Владеет методами формализации и моделирования программного обеспечения	Владеть 06.001 В - проверкой работоспособности и выполнением рефакторинга программного обеспечения: В/04.4 / 4 - Рефакторингом и оптимизацией программного кода. 06.001 D - навыками разработки требований и проектирования программного обеспечения: D/03.6 / 6 - Проектирование программного обеспечения.
	ПК-3 Способность создавать программные интерфейсы	ПК-3.1 Знает способы создания программных интерфейсов	Знать D/03.6 / 6 - Проектирование программного обеспечения в части создания программных интерфейсов.
		ПК-3.2 Умеет создавать интуитивно понятные программные интерфейсы	Уметь D/03.6 / 6 - проектировать программное обеспечение в части создания программных интерфейсов.

		ПК-3.3 Имеет навыки в создании современных программных интерфейсов	Владеть D/03.6 / 6 - проектированием программного обеспечения в части создания программных интерфейсов.
	ПК-9 Способность разрабатывать аппаратно-программные комплексы машинного и глубокого обучения	ПК-9.1 Знает методики разработки аппаратно-программных комплексов машинного и глубокого обучения	Знать D/03.6 / 6 - Проектирование программного обеспечения для комплексов машинного и глубокого обучения.
		ПК-9.2 Умеет разрабатывать аппаратно-программные комплексы машинного и глубокого обучения	Уметь D/03.6 / 6 - Проектировать программное обеспечение комплексов машинного и глубокого обучения.
		ПК-9.3 Имеет навыки разработки аппаратно-программных комплексов машинного и глубокого обучения	Владеть D/03.6 / 6 - Проектированием программного обеспечения для комплексов машинного и глубокого обучения.

2. Место дисциплины (модуля) в структуре образовательной программы

Место дисциплины (модуля) в структуре образовательной программы: **часть, формируемая участниками образовательных отношений**

Код компетенции	Предшествующие дисциплины	Параллельно осваиваемые дисциплины	Последующие дисциплины
ПК-1	Алгоритмы и структуры данных; Практико-ориентированный проект; Программирование на языке Java	Компьютерная вирусология; Программное обеспечение обработки визуальных данных; Проектирование и архитектура программных систем	Выполнение и защита выпускной квалификационной работы; Компьютерные средства искусственного интеллекта; Конструирование программного обеспечения; Производственная практика: преддипломная практика; Технологии машинного обучения
ПК-3	Графические системы компьютеров; Программирование на языке Java; Системное программное обеспечение; Теория автоматов и формальных языков	Программирование интернета вещей; Производственная практика: технологическая (проектно-технологическая) практика; Технологии виртуальной и дополненной реальности	Выполнение и защита выпускной квалификационной работы
ПК-9	Математическая логика и теория алгоритмов; Методы параллельных вычислений; Теория автоматов и формальных языков; Экспертные системы	Программирование интернета вещей; Технологии виртуальной и дополненной реальности	Выполнение и защита выпускной квалификационной работы; Компьютерные средства искусственного интеллекта; Системы поддержки принятия решений; Суперкомпьютерные вычислительные системы; Технологии машинного обучения

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Вид учебной работы	Всего часов / часов в электронной форме	7 семестр часов / часов в электронной форме
Аудиторная контактная работа (всего), в том числе:	12	12
Лабораторные работы	8	8
Лекции	4	4
Внеаудиторная контактная работа, КСР	4	4
Самостоятельная работа (всего), в том числе:	119	119
выполнение контрольных работ	10	10
подготовка к лабораторным работам	100	100
подготовка к экзамену	9	9
Контроль	9	9
Итого: час	144	144
Итого: з.е.	4	4

4. Содержание дисциплины (модуля), структурированное по темам (разделам), с указанием отведенного на них количества академических часов и видов учебных занятий

№ раздела	Наименование раздела дисциплины	Виды учебной нагрузки и их трудоемкость, часы				
		ЛЗ	ЛР	ПЗ	СРС	Всего часов
1	Основные понятия теории моделирования	1	0	0	22	23
2	Моделирование вычислительных процессов ¹	1	0	0	32	33
3	Статистическое моделирование	2	8	0	65	75
	КСР	0	0	0	0	4
	Контроль	0	0	0	0	9
	Итого	4	8	0	119	144

4.1 Содержание лекционных занятий

№ занятия	Наименование раздела	Тема лекции	Содержание лекции (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов / часов в электронной форме
7 семестр				
1	Основные понятия теории моделирования	Модель и моделирование	Модель и моделирование. Понятия модель, аналогия (прямая и косвенная, существенная и несущественная), свойства модели, моделирование. Адекватность модели объекту. Предмет и методы моделирования. Преимущества моделирования как метода исследования. Свойства моделей. Классификация моделей: материальные и идеальные, концептуальные и математические, аналоговые и имитационные.	1
2	Моделирование вычислительных процессов ¹	Моделирование компиляции алгебраических выражений	Три формы записи алгебраических выражений: инфиксная, постфиксная и префиксная. Моделирование преобразования инфиксного выражения в постфиксное. Стековая модель памяти и формализация процесса преобразования инфиксного выражения в постфиксное выражение по методу таблицы решений: алгоритм Дейкстры. Программная реализация некоторых функций алгоритма преобразования. Пример пошагового моделирования преобразования инфиксного выражения. Формализация вычисления значения выражения по его постфиксной форме записи.	1
3	Статистическое моделирование	Моделирование случайных величин	Математические модели специальных распределений с случайных величин (равномерное, нормальное, экспоненциальное, Пуассона). Математические методы моделирования псевдослучайных числовых последовательностей. Псевдослучайные числовые последовательности. Общий метод моделирования псевдослучайных числовых последовательностей. Моделирование равномерно распределенной случайной величины методом Лемера.	2
Итого за семестр:				4
Итого:				4

4.2 Содержание лабораторных занятий

№ занятия	Наименование раздела	Тема лабораторного занятия	Содержание лабораторного занятия (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов / часов в электронной форме
7 семестр				
1	Статистическое моделирование	Реализация метода Лемера построения генератора псевдослучайных числовых последовательностей с равномерным законом распределения.	Дискретные и непрерывные случайные величины Законы распределения случайных величин. Функции распределения случайных величин. Основные числовые характеристики случайных величин. Типовые распределения случайных величин.	2
2	Статистическое моделирование	Реализация метода Лемера построения генератора псевдослучайных числовых последовательностей с равномерным законом распределения.	Равномерное распределение случайных величин. Конгруэнтный метод Лемера построения генератора псевдослучайных числовых последовательностей с равномерным законом распределения.	2
3	Статистическое моделирование	Реализация метода Лемера построения генератора псевдослучайных числовых последовательностей с равномерным законом распределения.	Исследование основных статистических характеристик псевдослучайных последовательностей генерируемых на основе разработанной реализации метода Лемера.	2
4	Статистическое моделирование	Реализация метода Лемера построения генератора псевдослучайных числовых последовательностей с равномерным законом распределения.	Исследование основных статистических характеристик псевдослучайных последовательностей генерируемых на основе разработанной реализации метода Лемера.	2
Итого за семестр:				8
Итого:				8

4.3 Содержание практических занятий

Учебные занятия не реализуются.

4.4. Содержание самостоятельной работы

Наименование раздела	Вид самостоятельной работы	Содержание самостоятельной работы (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов
7 семестр			

<p>Основные понятия теории моделирования</p>	<p>Изучение теоретического материала</p>	<p>Объект, модель, моделирование. Особенность моделирования как метода исследования. Аналогия прямая и косвенная. Адекватность модели. Свойства моделей. Классификация моделей: материальные и идеальные, концептуальные и математические, непрерывные и дискретные, детерминированные и стохастические, функциональные и структурные,</p>	<p>22</p>
<p>Моделирование вычислительных процессов¹</p>	<p>Изучение теоретического материала</p>	<p>Три формы записи алгебраических выражений: инфиксная, постфиксная и префиксная. Моделирование преобразования инфиксного выражения в постфиксное. Концептуальная модель процесса преобразования инфиксного выражения в постфиксное. Неформальное преобразование инфиксного выражения в постфиксное и в префиксное. Стековая память и особенности ее моделирования. Формализация процесса преобразования инфиксного выражения в постфиксное выражение по методу таблицы решений: алгоритм Дейкстра. Концептуальная модель процесса вычисления значения выражения, представленного в постфиксной форме записи. Арифметический стек. Модель вычисления значения алгебраического выражения, представленного в постфиксной форме записи, на основе арифметического стека. Пример пошагового выполнения алгоритма вычисления значения выражения, представленного в постфиксной форме записи.</p>	<p>32</p>

Статистическое моделирование	Изучение теоретического материала. Подготовка к лабораторным работам и оформление отчетов по ним.	1. Элементы теории вероятностей и математической статистики. Дискретные и непрерывные случайные величины. Законы распределения случайных величин. Функции распределения случайных величин. Основные числовые характеристики случайных величин. Специальные распределения случайных величин: равномерное, экспоненциальное, Пуассона, нормальное. 2. Генераторы псевдослучайных числовых последовательностей. Моделирование псевдослучайной числовой последовательности с равномерным законом распределения алгоритмическим способом: базовые понятия; общий подход к получению псевдо-случайных последовательностей; метод Лемера. Построение генераторов с заданным законом распределения числовых последовательностей: универсальный подход; частные методы. Генератор последовательностей с нормальным законом распределения. Генератор последовательностей с экспоненциальным законом распределения. 3. Имитационное статистическое моделирование. Имитационные модели и имитационное моделирование. Метод Монте-Карло. Пример статистического моделирования: приближенное вычисление числа Пи. 4. Основы теории систем массового обслуживания. Простейшая система массового обслуживания. Классы моделей СМО. Модели потоков заявок. Простейший поток событий. События основные и вспомогательные. Моделирование событий. Модельное время и таймер модельного времени.	65
Итого за семестр:			119
Итого:			119

5. Перечень учебной литературы и учебно-методического обеспечения по дисциплине (модулю)

№ п/п	Библиографическое описание	Ресурс НТБ СамГТУ (ЭБС СамГТУ, IPRbooks и т.д.)
Основная литература		
1	Голубева, Н.В. Математическое моделирование систем и процессов : учеб.пособие / Н. В. Голубева.- СПб., Лань, 2013.- 191 с.	Электронный ресурс

2	Мартемьянов, Б.В. Моделирование : учеб. пособие / Б. В. Мартемьянов, А. Б. Мартемьянов; Самар.гос.техн.ун-т.- Самара, 2010.- 88 с.	Электронный ресурс
3	Морозов, В.К. Моделирование информационных и динамических систем : учеб.пособие / В. К. Морозов , Г. Н. Рогачев.- М., Академия, 2011.- 377 с.	Электронный ресурс
4	Орлов, С.П. Арифметика ЭВМ и логические основы переключательных функций : Учеб.пособие / С. П. Орлов, Б. В. Мартемьянов .- 3-е изд.,испр.и доп..- М., Машиностроение-1, 2005.- 255 с.	Электронный ресурс
Дополнительная литература		
5	Батищев, В.И. Основы теории систем : учеб. пособие / В. И. Батищев; Самар.гос.техн.ун-т.- Самара, 2012.- 147 с.	Электронный ресурс
6	Введение в математическое моделирование : учеб. пособие / под ред. П. В. Трусова.- М., Логос, 2014.- 439 с.	Электронный ресурс
7	Моделирование : метод.указания / Самар.гос.техн.ун-т, Вычислительная техника; сост. Б. В. Мартемьянов.- Самара, 2010.- 18 с..- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu elib 478	Электронный ресурс
8	Пугачев, А.И. Теория проектирования ЭВМ : лаборатор.практикум / А. И. Пугачев, Б. В. Мартемьянов; Самар.гос.техн.ун-т, Вычислительная техника.- Самара, 2010.- 50 с..- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu elib 541	Электронный ресурс

Доступ обучающихся к ЭР НТБ СамГТУ (elib.samgtu.ru) осуществляется посредством электронной информационной образовательной среды университета и сайта НТБ СамГТУ по логину и паролю.

6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения

При проведении лекционных занятий используется мультимедийное оборудование.

Организовано взаимодействие обучающегося и преподавателя с использованием электронной ин-формационной образовательной среды университета.

№ п/п	Наименование	Производитель	Способ распространения
1	Microsoft Windows 7 Profes-sional операционная системат	Microsoft (Зарубежный)	Лицензионное
2	Microsoft Visul Studio, Net2008, NetBeans	Microsoft (Зарубежный)	Свободно распространяемое

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных, информационно-справочных систем

№ п/п	Наименование	Краткое описание	Режим доступа
1	Электронно-библиотечная система IPRbooks	http://www.iprbookshop.ru/	Российские базы данных ограниченного доступа

2	Электронная библиотека изданий СамГТУ	http://irbis.samgtu.local/cgi-bin/irbis64r_01/cgiirbis_64.exe	Российские базы данных ограниченного доступа
3	РОСПАТЕНТ	http://www1.fips.ru/	Российские базы данных ограниченного доступа
4	ScienceDirect (Elsevier) - естественные науки, техника, медицина и общественные науки.	http://www.sciencedirect.com/	Зарубежные базы данных ограниченного доступа
5	Scopus - база данных рефератов и цитирования	http://www.scopus.com/	Зарубежные базы данных ограниченного доступа
6	Библиотека компьютерной литературы	http://it.eup.ru/	Ресурсы открытого доступа
7	Журнал Вестник СамГТУ. Серия «Технические науки».	http://vestnik-teh.samgtu.ru/	Ресурсы открытого доступа
8	Научная электронная библиотека «КиберЛенинка» (полные тексты научных статей из журналов)	http://cyberleninka.ru/search	Ресурсы открытого доступа
9	Ресурсы по информационным технологиям	http://compress.ru/article.aspx	Ресурсы открытого доступа
10	УИС РОССИЯ - Университетская информационная система РОССИЯ	http://www.cir.ru/index.jsp	Ресурсы открытого доступа

8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Лекционные занятия

Аудитории для лекционных занятий укомплектованы мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории (наборы демонстрационного обо-рудования (проектор, экран, компьютер/ноутбук), учебно-наглядные пособия, тематические иллюстрации).

Практические занятия

не предусмотрены

Лабораторные занятия

Для лабораторных занятий используются аудитория №314, корпус № 8, оснащенная следующим обо-рудованием:

компьютерами iRU i3-4160/4 GB с мониторами Samsung S20D300NH (11 шт.) с выходом в сеть Интернет. с лицензионным программным обеспечением, коммутатором D-Link DES, мультимедийным проектором DS 1700 , экраном переносным.

Для лабораторных занятий используются аудитория №309, корпус № 8, оснащенная следующим обо-рудованием:

компьютерами Intel Core 2 Duo E6300 с мониторами LG Flatron L1752S - SF17 (11 шт). с выходом в Интернет, с лицензионным программным обеспечением, коммутатором HUB Comrex PS16, мультимедий-

ным проектором NEC, экраном настенным проекционным.

Самостоятельная работа

Помещения для самостоятельной работы оснащены компьютерной техникой с возможностью под-ключения к сети «Интернет» и доступом к электронной информационно-образовательной среде СамГТУ:

- читальный зал НТБ СамГТУ (ауд. 200 корпус № 8; ауд. 125 корпус № 1; ауд. 41, 31, 34, 35 Главный корпус библиотеки, ауд. 83а, 414, 416, 0209 АСА СамГТУ; ауд. 401 корпус №10);
- компьютерные классы (ауд. 208, 210 корпус № 8).

9. Методические материалы

Методические рекомендации при работе на лекции

До лекции студент должен просмотреть учебно-методическую и научную литературу по теме лекции с тем, чтобы иметь представление о проблемах, которые будут разбираться в лекции.

Перед началом лекции обучающимся сообщается тема лекции, план, вопросы, подлежащие рассмотрению, доводятся основные литературные источники. Весь учебный материал, сообщаемый преподавателем, должен не просто прослушиваться. Он должен быть активно воспринят, т.е. услышан, осмыслен, понят, зафиксирован на бумаге и закреплен в памяти. Приступая к слушанию нового учебного материала, полезно мысленно установить его связь с ранее изученным. Следя за техникой чтения лекции (акцент на существенном, повышение тона, изменение ритма, пауза и т.п.), необходимо вслед за преподавателем уметь выделять основные категории, законы и определять их содержание, проблемы, предполагать их возможные решения, доказательства и выводы. Осуществляя такую работу, можно значительно облегчить себе понимание учебного материала, его конспектирование и дальнейшее изучение.

Конспектирование лекции позволяет обработать, систематизировать и лучше сохранить полученную информацию с тем, чтобы в будущем можно было восстановить в памяти основные, содержательные моменты. Типичная ошибка, совершаемая обучающимся, дословное конспектирование речи преподавателя. Как правило, при записи «слово в слово» не остается времени на обдумывание, анализ и синтез информации. Отбирая нужную информацию, главные мысли, проблемы, решения и выводы, необходимо сокращать текст, строить его таким образом, чтобы потом можно было легко в нем разобраться. Желательно оставить в рабочих конспектах поля, на которых можно будет делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. С окончанием лекции работа над конспектом не может считаться завершенной. Нужно еще восстановить отдельные места, проверить, все ли понятно, уточнить что-то на консультации и т.п. с тем, чтобы конспект мог быть использован в процессе подготовки к практическим занятиям, зачету, экзамену. Конспект лекции – незаменимый учебный документ, необходимый для самостоятельной работы.

Методические рекомендации при работе на лабораторном занятии

Проведение лабораторной работы делится на две условные части: теоретическую и практическую.

Необходимыми структурными элементами занятия являются проведение лабораторной работы, проверка усвоенного материала, включающая обсуждение теоретических основ выполняемой работы.

Перед лабораторной работой, как правило, проводится технико-теоретический инструктаж по использованию необходимого оборудования. Преподаватель корректирует деятельность обучающегося в процессе выполнения работы (при необходимости). После завершения лабораторной работы подводятся итоги, обсуждаются результаты деятельности.

Возможны следующие формы организации лабораторных работ: фронтальная, групповая и индивидуальная. При фронтальной форме выполняется одна и та же работа (при этом возможны

различные варианты заданий). При групповой форме работы выполняется группой (командой). При индивидуальной форме обучающимися выполняются индивидуальные работы.

По каждой лабораторной работе имеются методические указания по их выполнению, включающие необходимый теоретический и практический материал, содержащие элементы и последовательную инструкцию по проведению выбранной работы, индивидуальные варианты заданий, требования и форму отчетности по данной работе.

Методические рекомендации по выполнению самостоятельной работы

Организация самостоятельной работы обучающихся ориентируется на активные методы овладения знаниями, развитие творческих способностей, переход от поточного к индивидуализированному обучению с учетом потребностей и возможностей обучающегося.

Самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса.

Самостоятельная работа реализуется:

- непосредственно в процессе аудиторных занятий;
- на лекциях, практических занятиях;
- в контакте с преподавателем вне рамок расписания;
- на консультациях по учебным вопросам, в ходе творческих контактов, при ликвидации задолженностей, при выполнении индивидуальных заданий и т.д.;
- в библиотеке, дома, на кафедре при выполнении обучающимся учебных и практических задач.

Эффективным средством осуществления обучающимся самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к учебным планам, рабочим программам дисциплин (модулей), практик, к изданиям электронных библиотечных систем.

10. Фонд оценочных средств по дисциплине (модулю)

Фонд оценочных средств представлен в приложении № 1.

Приложение 1 к рабочей программе дисциплины
Б1.В.01.02 «Моделирование»

**Фонд оценочных средств
по дисциплине
Б1.В.01.02 «Моделирование»**

Код и направление подготовки (специальность)	09.03.04 Программная инженерия
Направленность (профиль)	Программная инженерия
Квалификация	Бакалавр
Форма обучения	Заочная
Год начала подготовки	2022
Институт / факультет	Институт автоматизации и информационных технологий
Выпускающая кафедра	кафедра "Вычислительная техника"
Кафедра-разработчик	кафедра "Вычислительная техника"
Объем дисциплины, ч. / з.е.	144 / 4
Форма контроля (промежуточная аттестация)	Экзамен

**Перечень планируемых результатов обучения по дисциплине (модулю),
соотнесенных с планируемыми результатами освоения образовательной
программы**

Наименование категории (группы) компетенций	Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения (знать, уметь, владеть, соотнесенные с индикаторами достижения компетенции)
Профессиональные компетенции			
Не предусмотрено	ПК-1 Владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения	ПК-1.1 Знает основы моделирования и формальные методы конструирования программного обеспечения	Знать 06.001 В - Проверка работоспособности и рефакторинг программного обеспечения: В/04.4 / 4 - Рефакторинг и оптимизация программного кода. 06.001 D - Разработка требований и проектирование программного обеспечения: D/03.6 / 6 - Проектирование программного обеспечения.
		ПК-1.2 Умеет использовать формальные методы конструирования программного обеспечения	Уметь 06.001 В - Проверять работоспособность и выполнять рефакторинг программного обеспечения: В/04.4 / 4 - Выполнять рефакторинг и оптимизацию программного кода. 06.001 D - Выполнять разработку требований и проектирование программного обеспечения: D/03.6 / 6 - Выполнять проектирование программного обеспечения.
		ПК-1.3 Владеет методами формализации и моделирования программного обеспечения	Владеть 06.001 В - проверкой работоспособности и выполнением рефакторинга программного обеспечения: В/04.4 / 4 - Рефакторингом и оптимизацией программного кода. 06.001 D - навыками разработки требований и проектирования программного обеспечения: D/03.6 / 6 - Проектирование программного обеспечения.
	ПК-3 Способность создавать программные интерфейсы	ПК-3.1 Знает способы создания программных интерфейсов	Знать D/03.6 / 6 - Проектирование программного обеспечения в части создания программных интерфейсов.
		ПК-3.2 Умеет создавать интуитивно понятные программные интерфейсы	Уметь D/03.6 / 6 - проектировать программное обеспечение в части создания программных интерфейсов.

		ПК-3.3 Имеет навыки в создании современных программных интерфейсов	Владеть D/03.6 / 6 - проектированием программного обеспечения в части создания программных интерфейсов.
	ПК-9 Способность разрабатывать аппаратно-программные комплексы машинного и глубокого обучения	ПК-9.1 Знает методики разработки аппаратно-программных комплексов машинного и глубокого обучения	Знать D/03.6 / 6 - Проектирование программного обеспечения для комплексов машинного и глубокого обучения.
		ПК-9.2 Умеет разрабатывать аппаратно-программные комплексы машинного и глубокого обучения	Уметь D/03.6 / 6 - Проектировать программное обеспечение комплексов машинного и глубокого обучения.
		ПК-9.3 Имеет навыки разработки аппаратно-программных комплексов машинного и глубокого обучения	Владеть D/03.6 / 6 - Проектированием программного обеспечения для комплексов машинного и глубокого обучения.

Матрица соответствия оценочных средств запланированным результатам обучения

Код индикатора достижения компетенции	Результаты обучения	Оценочные средства	Текущий контроль успеваемости	Промежуточная аттестация
Основные понятия теории моделирования				
ПК-1.1 Знает основы моделирования и формальные методы конструирования программного обеспечения	Знать 06.001 В - Проверка работоспособности и рефакторинг программного обеспечения: В/04.4 / 4 - Рефакторинг и оптимизация программного кода. 06.001 D - Разработка требований и проектирование программного обеспечения: D/03.6 / 6 - Проектирование программного обеспечения.	тесты	Да	Да
		Отчеты по лабораторным работам	Да	Нет
ПК-1.2 Умеет использовать формальные методы конструирования программного обеспечения	Уметь 06.001 В - Проверять работоспособность и выполнять рефакторинг программного обеспечения: В/04.4 / 4 - Выполнять рефакторинг и оптимизацию программного кода. 06.001 D - Выполнять разработку требований и проектирование программного обеспечения: D/03.6 / 6 - Выполнять проектирование программного обеспечения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-1.3 Владеет методами формализации и моделирования программного обеспечения	Владеть 06.001 В - проверкой работоспособности и выполнением рефакторинга программного обеспечения: В/04.4 / 4 - Рефакторингом и оптимизацией программного кода. 06.001 D - навыками разработки требований и проектирования программного обеспечения: D/03.6 / 6 - Проектирование программного обеспечения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да

ПК-3.1 Знает способы создания программных интерфейсов	Знать D/03.6 / 6 - Проектирование программного обеспечения в части создания программных интерфейсов.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-3.2 Умеет создавать интуитивно понятные программные интерфейсы	Уметь D/03.6 / 6 - проектировать программное обеспечение в части создания программных интерфейсов.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-3.3 Имеет навыки в создании современных программных интерфейсов	Владеть D/03.6 / 6 - проектированием программного обеспечения в части создания программных интерфейсов.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-9.1 Знает методики разработки аппаратно-программных комплексов машинного и глубокого обучения	Знать D/03.6 / 6 - Проектирование программного обеспечения для комплексов машинного и глубокого обучения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-9.2 Умеет разрабатывать аппаратно-программные комплексы машинного и глубокого обучения	Уметь D/03.6 / 6 - Проектировать программное обеспечение комплексов машинного и глубокого обучения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-9.3 Имеет навыки разработки аппаратно-программных комплексов машинного и глубокого обучения	Владеть D/03.6 / 6 - Проектированием программного обеспечения для комплексов машинного и глубокого обучения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
Моделирование вычислительных процессов¹				
ПК-1.1 Знает основы моделирования и формальные методы конструирования программного обеспечения	Знать 06.001 В - Проверка работоспособности и рефакторинг программного обеспечения: В/04.4 / 4 - Рефакторинг и оптимизация программного кода. 06.001 D - Разработка требований и проектирование программного обеспечения: D/03.6 / 6 - Проектирование программного обеспечения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-1.2 Умеет использовать формальные методы конструирования программного обеспечения	Уметь 06.001 В - Проверять работоспособность и выполнять рефакторинг программного обеспечения: В/04.4 / 4 - Выполнять рефакторинг и оптимизацию программного кода. 06.001 D - Выполнять разработку требований и проектирование программного обеспечения: D/03.6 / 6 - Выполнять проектирование программного обеспечения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да

ПК-1.3 Владеет методами формализации и моделирования программного обеспечения	Владеть 06.001 В - проверкой работоспособности и выполнением рефакторинга программного обеспечения: В/04.4 / 4 - Рефакторинг и оптимизацией программного кода. 06.001 D - навыками разработки требований и проектирования программного обеспечения: D/03.6 / 6 - Проектирование программного обеспечения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-3.1 Знает способы создания программных интерфейсов	Знать D/03.6 / 6 - Проектирование программного обеспечения в части создания программных интерфейсов.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-3.2 Умеет создавать интуитивно понятные программные интерфейсы	Уметь D/03.6 / 6 - проектировать программное обеспечение в части создания программных интерфейсов.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-3.3 Имеет навыки в создании современных программных интерфейсов	Владеть D/03.6 / 6 - проектированием программного обеспечения в части создания программных интерфейсов.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-9.1 Знает методики разработки аппаратно-программных комплексов машинного и глубокого обучения	Знать D/03.6 / 6 - Проектирование программного обеспечения для комплексов машинного и глубокого обучения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-9.2 Умеет разрабатывать аппаратно-программные комплексы машинного и глубокого обучения	Уметь D/03.6 / 6 - Проектировать программное обеспечение комплексов машинного и глубокого обучения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-9.3 Имеет навыки разработки аппаратно-программных комплексов машинного и глубокого обучения	Владеть D/03.6 / 6 - Проектированием программного обеспечения для комплексов машинного и глубокого обучения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
Статистическое моделирование				
ПК-1.1 Знает основы моделирования и формальные методы конструирования программного обеспечения	Знать 06.001 В - Проверка работоспособности и рефакторинг программного обеспечения: В/04.4 / 4 - Рефакторинг и оптимизация программного кода. 06.001 D - Разработка требований и проектирование программного обеспечения: D/03.6 / 6 - Проектирование программного обеспечения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да

ПК-1.2 Умеет использовать формальные методы конструирования программного обеспечения	Уметь 06.001 В - Проверять работоспособность и выполнять рефакторинг программного обеспечения: В/04.4 / 4 - Выполнять рефакторинг и оптимизацию программного кода. 06.001 D - Выполнять разработку требований и проектирование программного обеспечения: D/03.6 / 6 - Выполнять проектирование программного обеспечения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-1.3 Владеет методами формализации и моделирования программного обеспечения	Владеть 06.001 В - проверкой работоспособности и выполнением рефакторинга программного обеспечения: В/04.4 / 4 - Рефакторингом и оптимизацией программного кода. 06.001 D - навыками разработки требований и проектирования программного обеспечения: D/03.6 / 6 - Проектирование программного обеспечения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-3.1 Знает способы создания программных интерфейсов	Знать D/03.6 / 6 - Проектирование программного обеспечения в части создания программных интерфейсов.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-3.2 Умеет создавать интуитивно понятные программные интерфейсы	Уметь D/03.6 / 6 - проектировать программное обеспечение в части создания программных интерфейсов.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-3.3 Имеет навыки в создании современных программных интерфейсов	Владеть D/03.6 / 6 - проектированием программного обеспечения в части создания программных интерфейсов.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-9.1 Знает методики разработки аппаратно-программных комплексов машинного и глубокого обучения	Знать D/03.6 / 6 - Проектирование программного обеспечения для комплексов машинного и глубокого обучения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-9.2 Умеет разрабатывать аппаратно-программные комплексы машинного и глубокого обучения	Уметь D/03.6 / 6 - Проектировать программное обеспечение комплексов машинного и глубокого обучения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да
ПК-9.3 Имеет навыки разработки аппаратно-программных комплексов машинного и глубокого обучения	Владеть D/03.6 / 6 - Проектированием программного обеспечения для комплексов машинного и глубокого обучения.	Отчеты по лабораторным работам	Да	Нет
		тесты	Да	Да

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К КУРСОВОЙ РАБОТЕ

Задания к курсовой работе предполагают достаточно свободное владение студентами одним из языков программирования высокого уровня и соответствующей средой разработки и отладки программ, знания арифметических и логических основ цифровой техники, а также изученных ранее профилирующих дисциплин: «Программирование», «Схемотехника», «Дискретная математика», «Прикладная теория цифровых автоматов», «Теория автоматов и формальных языков».

Содержание курсовой работы не повторяет содержание заданий к лабораторным занятиям и этим способствует охвату в рамках дисциплины более широкого круга задач моделирования объектов будущей профессиональной деятельности выпускников.

1 ТЕМА КУРСОВОЙ РАБОТЫ И ИСХОДНЫЕ ДАННЫЕ

1.1 Тема и цель курсовой работы

Тема курсовой работы – «Разработка среды для имитационного моделирования операционного устройства».

Цель курсовой работы – формирование практических навыков разработки имитационных моделей компонент и систем вычислительной техники с использованием инструментальных средств языка программирования высокого уровня.

В курсовой работе необходимо разработать интерактивную среду для моделирования поведения конкретного операционного устройства (ОУ), определяемого исходной микропрограммой выполнения арифметической операции, заданной в виде ГСА.

В вариантах заданий варьируются исходные ГСА, задающие микропрограммы выполнения арифметических операций, поэтому

при общем шаблоне задания на курсовую работу получают различные по содержанию программные реализации моделей.

1.2 Исходные данные

Исходными данными к курсовой работе являются:

- индивидуальный вариант микропрограммы выполнения арифметической операции, выдаваемый студентам в виде ГСА (пример на рисунке 1.1);
- требования к реализуемым уровням и режимам моделирования объекта;
- требования к обязательным возможностям интерфейсных средств среды моделирования;
- требования к оформлению пояснительной записки и особенностям программной реализации.

Операционное устройство считается состоящим из управляющего автомата (УА) и операционного автомата (ОА).

Структура ОА не проектируется. Основу модели ОА составляет набор процедур, моделирующих выполнения микроопераций над кодами данных и вычисления логических функций, задающих выходные сигналы ОА о состояниях кодов данных.

Форматы разрядных сеток для представления данных определены в каждой ГСА. Для представления этих данных в модели студент должен выбрать наиболее подходящие типы данных используемого языка программирования.

Исходные ГСА задают либо микропрограмму (МП) умножения исходных операндов A и B ($C := A \times B$), либо МП деления ($C := A/B$). Операнды A , B операций умножения и деления представляются 16-ти битными кодами формата «фиксированное положение позиционной точки» со знаковым разрядом.

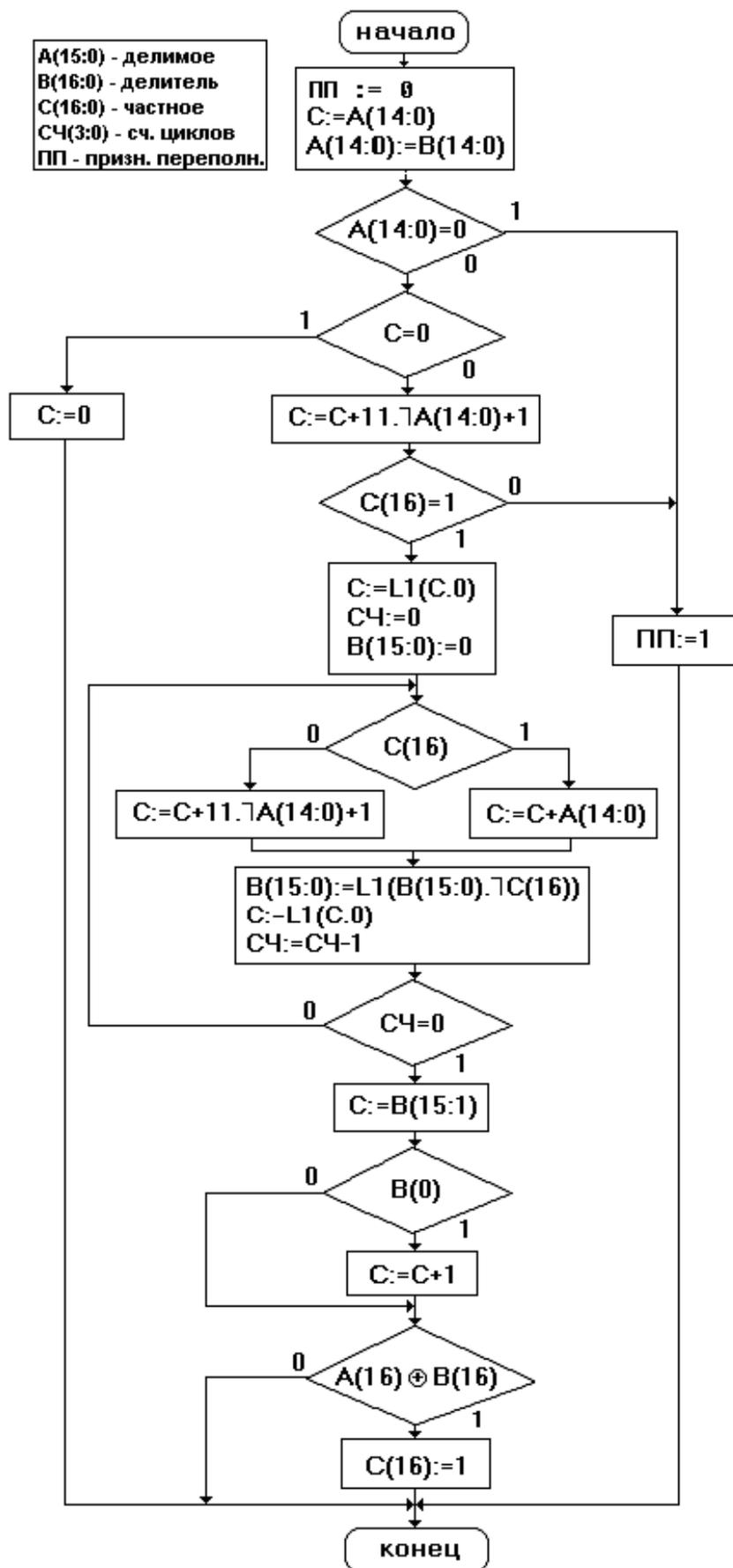


Рисунок 1.1 - Пример микропрограммы выполнения арифметической операции

Во всех вариантах ГСА предполагается: $|A| < 1$; $|B| < 1$; $|C| < 1$. Из этого следует, что позиционная точка на разрядной сетке (с фиксированным положением позиционной точки) находится между разрядом знака и разрядами цифровой части кодов.

Операнды и результат операции являются дробными числами, но должны моделироваться целочисленными кодами: вещественный формат чисел абсолютно недопустим, потому что этот формат имеет принципиально иную структуру, чем предполагается во всех ГСА. В работе необходимо дать объяснение возможности моделирования дробных чисел целочисленными типами данных используемого языка программирования.

В интерактивной среде моделирования реализуются два уровня моделирования ОУ:

- моделирование на уровне микропрограммы;
- моделирование на уровне взаимодействия УА и ОА;

Моделирование ОУ на уровне микропрограммы. Ветвления в алгоритме программируются средствами условного оператора *if*. Модель отдельной микрокоманды представляет собой последовательный вызов соответствующих процедур, моделирующих выполнения микроопераций. Эта последовательность закладывается в разрабатываемый программный код модели.

В результате такого моделирования ОУ можно проверить правильность микропрограммы выполнения арифметической операции.

Моделирование ОУ на уровне взаимодействия УА и ОА. Модель УА строится на основании результатов проектирования УА как автомата Мили на жесткой логике.

Модель УА формирует вектор Y управляющих сигналов, детализирующий команды для ОА до уровня микроопераций. Модель ОА в результате анализа состояния вектора Y вызывает процедуры выполнения указанных вектором микроопераций и затем вычисляет вектор X значений выходных сигналов, поступающих на вход УА.

В модели этого режима запрещается закладывать в программу какую-либо последовательность вызовов процедур моделирования микроопераций. Каждая такая процедура может быть вызвана только как следствие обнаружения конкретного (например, единичного) состояния определенной компоненты вектора Y .

Модель УА должна выделять следующие его компоненты:

- память состояний (ПС) УА на D триггерах;
- дешифратор кодов (ДК) состояний УА (это необязательный структурный компонент УА);
- комбинационные схемы (КС), формирующие вектор выходных сигналов Y , и вектор D сигналов управления состояниями разрядов ПС УА;
- память на D триггерах для запоминания некоторых компонент вектора логических условий.

Модели КС строятся на функциональном уровне, т.е. как последовательность булевых выражений, вычисляющих компоненты векторов Y и D .

В интерактивной среде на каждом из уровней моделирования реализуются два режима моделирования ОУ:

- пошаговый режим выполнения микропрограммы;
- автоматический режим выполнения микропрограммы.

В первом из режимов предполагается наличие на форме кнопки с надписью типа «Такт». Каждое нажатие такой кнопки должно вызывать продвижение по микропрограмме на 1 такт. Причем этот такт должен соответствовать понятию такта работы ОУ при проектировании УА на основе модели цифрового управляющего автомата типа Мили.

В автоматическом режиме предполагается наличие на форме кнопки с надписью типа «Выполнить МП». При нажатии на эту кнопку микропрограмма выполняется с начала до конца без вмешательства со стороны пользователя, т.е. автоматически.

Интерфейсные средства среды моделирования должны позволять:

- наблюдать на экране ГСА, размеченную состояниями автомата модели Мили;
- выбирать любой из описанных выше уровней и режимов моделирования:
- задавать начальные значения операндов операции;
- просматривать процесс выполнения микропрограммы по шагам (по тактам) с отображением на форме всех используемых в микропрограмме кодов;
- выполнять микропрограмму автоматически;
- отображать все изменения кодов переменных, участвующих в микропрограмме;
- при моделировании на уровне взаимодействия УА и ОА отображать на форме состояния векторов, задающих все входные, выходные и внутренние векторы УА в предположении, что УА спроектирован как автомат модели Мили на жесткой логике с элементами памяти на D триггерах, а также отображать на ГСА графическую метку, задающую текущее состояние УА.

1.2 Пояснения к содержанию исходных ГСА

1.2.1 Микропрограммный управляющий автомат

Микропрограммный управляющий автомат – это конечный автомат, обеспечивающий выполнение заданной микропрограммы операционным устройством.

Микропрограмма (МП) описывает алгоритм выполнения какой-либо сложной операции (арифметической, логической и т.п.) в терминах микроопераций (МО) и логических условий.

К МО относятся простейшие действия над словами информации, выполняемые элементами операционного автомата (ОА): загрузка и хранение слова, инверсия слова, сложения двух слов, ин-

кремент (+1), декремент (-1), сдвиг на 1 (или более) разряд влево или вправо и т.п.

МП часто записывается в виде ГСА (подобно ГСА в программировании). Идентификаторы в МО обозначают операционные элементы (ОЭ) операционного автомата (ОА). Например, RA – регистр A, CT – счетчик (Count) и т.д. Имя ОЭ на основе триггеров (то есть ОЭ, обеспечивающего хранение информации) является одновременно идентификатором кода, хранящимся на этом ОЭ. МО записываются в виде операторов присваивания с использованием идентификаторов и символов МО.

1.2.2 Примеры микроопераций

$RC := 0$ – записать (загрузить) в регистр RC число 0.

$RA := A$ – загрузить в регистр RA число A с регистра A.

$RA := \neg RA$ – поразрядное инвертирование содержимого регистра.

$RC := RA + RB$ – в регистр RC поместить результат сложения кодов, хранящихся в регистрах RA и RB.

$RA := L1 (RA.0)$ – сдвинуть влево (L - left) на 1 разряд содержимое регистра RA, а в освободившийся крайний правый разряд записать 0. Точка перед нулем означает операцию склеивания двух кодов. Таким образом, в данной МО сдвига участвует «склеенный» код.

$CT := CT + 1$ – выполнить МО счета. Такая МО отличается от соответствующего оператора языка программирования! В программах она всегда означает увеличение кода на 1, то есть, значение кода CT после операции всегда будет больше, чем его значение до операции. В МП в результате такой МО значение кода CT может стать равным 0! Это произойдет, когда значение кода было максимально возможным. При этом, в отличие от программ, факт переполнения разрядной сетки не фиксируется!

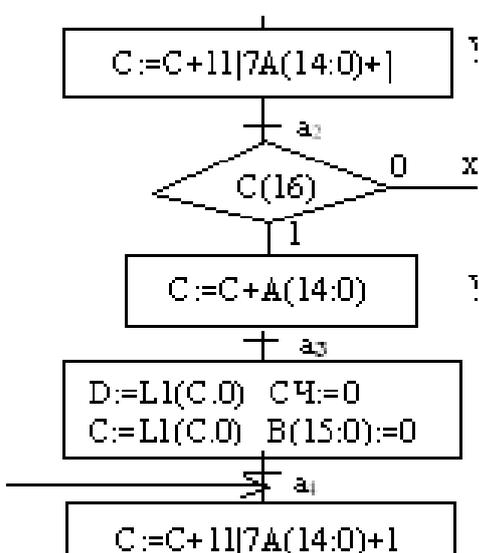
Все ОЭ характеризуются разрядностью. Разряды нумеруются справа-налево, начиная с нуля. Например, запись СТ[3..0] означает, что в МО участвуют как единое целое 4 конкретных разряда счетчика СТ. Его разряды (начиная со старших) обозначаются как СТ[3], СТ[2], СТ[1], СТ[0].

Обратите внимание! В операторе RA: = L1(RA.0) использованы круглые скобки: в них заключен аргумент операции сдвига. В квадратных скобках (точно, как в программах элементы массивов) указываются номера разрядов (элементов) кода. В вариантах ГСА эта особенность не всегда соблюдается, точнее, редко соблюдается: вместо квадратных скобок [...] использованы круглые скобки (...).

Во всех случаях, когда идентификатор регистра записан без указания на конкретный разряд или интервал разрядов, в МО участвует содержимое всего кода этого регистра.

МО, допускающие совместное выполнение, называются *совместимыми*. Они могут записываться в одной и той же операторной вершине ГСА. Совместимость МО достигается за счет того, что все они выполняются аппаратными средствами разных компонент ОА. Приписывание совместимых МО одной операторной вершине не следует толковать как одновременное выполнение всех этих МО. Они выполняются в течение одного такта работы ОА, но возможно в разные моменты этого такта.

В некоторых вариантах МП имеются ошибки (некорректности)



в записях МО. На приведенном фрагменте ГСА (рисунок 1.2) есть примеры таких ошибок:

- ни в одном из вариантов ГСА не должно быть цифры 7. Этой цифрой подменен символ « \rightarrow »;

- вертикальная черточка после кода «11» в верхней операторной вершине

Рисунок 1.2 – Фрагмент ГСА

подменяет символ «.», который, как сказано выше, означает операцию склеивания кодов.

Первая МО приведенного фрагмента ГСА должна выглядеть так:

$$C := C + 11.\neg A[14:0] + 1.$$

Содержание данной МО следующее:

- к поразрядной инверсии цифровых разрядов кода А с номерами от 14 до 0 слева «приклеивается» код «11»;

- полученный код складывается с кодом С и к результату прибавляется *единица младшего разряда*. То есть, эта 1 при вычислении суммы позиционируется под разрядом $(C + 11.\neg A[14:0])[0]$.

Поэтому рассматриваемая 1 не является целым числом 1!

Внимание! Во всех случаях операция «... + 1» выполняется прибавлением 1 к младшему разряду того кода, с которым она складывается. Например, в МО

$$C := C[31:15] + 1$$

цифра 1 позиционируется под разрядом C[15].

В микрооперациях типа

$$C := C[31:15] + A[14:0]$$

все слагаемые (их может быть больше двух) *выравниваются по младшему разряду кодов*, участвующих в сложении. В этом примере в операции суммирования участвует не код С, а только его часть: код C[31:15].

Возможны варианты МО вида

$$A[14:0] := \dots$$

В результате такой МО могут измениться только указанные разряды кода А: разряд А[15] должен остаться прежним и об этом необходимо позаботиться при разработке модели такой МО.

Необходимо понимать простой арифметический смысл каждой МО данной МП. Рассмотрим прокомментированную выше МО

$$C := C + 11.\neg A[14:0] + 1.$$

Во всех ГСА коды, например, А и А[15:0] тождественны поскольку они представляются на разрядной сетке длиной 16 бит. Когда число А представлено своим прямым кодом справедливо равенство $A[14:0] = |A|$. Код «11», приклеенный слева к коду А[14:0], занимает позиции двух знаковых разрядов модифицированного кода числа и поэтому обозначает знак «-». В результате код «11. $\neg A[14:0] + 1$ » - это дополнительный код числа «-|А|». Поэтому вся МО задает операцию $C := C - |A|$. Понятно, что рассматриваемая МО предполагает, что код С – это дополнительный код соответствующего числа.

1.2.3 Примеры логических условий

В МП в виде ГСА кроме МО используются логические условия (ЛУ) (условные вершины ГСА). ЛУ, как правило, вырабатываются операционным автоматом и могут принимать значения «0» (ложно) либо «1» (истинно).

СТ= 0 – проверка утверждения «содержимое счетчика СТ = 0».

RA[0] – проверка содержимого младшего разряда регистра RA. В данном случае это выражение эквивалентно утверждению «RA[0]=1».

2 ОБЩИЕ ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ И ЕЕ ОФОРМЛЕНИЮ

2.1 ЯЗЫК ПРОГРАММИРОВАНИЯ И ГРАФИЧЕСКИЙ РЕЖИМ

Программная часть курсовой работы выполняется с использованием среды *по выбору студента*, предназначенной для создания Windows приложений. Используемый графический режим видеосистемы должен быть не хуже, чем TrueColor с пространственным разрешением 1024*768 пикселей.

2.2 ОТЧЕТНЫЕ МАТЕРИАЛЫ ПО КУРСОВОЙ РАБОТЕ

В качестве результата выполнения курсовой работы *представляются*:

- какой-либо носитель информации;
- пояснительная записка.

Носитель информации должен содержать:

- исходные тексты всех разработанных программных единиц;
- комплект файлов, достаточный для запуска программы из соответствующей операционной системы (ОС) и среды разработки;
- «бумажную» и электронную версии пояснительной записки, выполненную в текстовом процессоре Word версии 2000 или выше с соблюдением требований ГОСТов на оформление технических текстов;
- файл readme с информацией об авторе, языке программирования и об ОС.

Структура пояснительной записки приведена ниже в подразделе 2.4.

При описании положения на форме интерфейсных средств не допускаются фразы со словами «выше/ниже», «правее/левее». Необходимо использовать нумерованные выноски за границы описываемой формы. Выноски нумеровать по часовой стрелке, начиная с самой левой и верхней. Затем под рисунком формы содержание интерфейсных средств поясняется через номера выносок. При этом желательно такое описание представить в форме таблицы с колон-

ками «№», «Назначение». При этом в колонке «Назначение» не должно быть фраз типа «предназначена для ...». Название колонки заменяет подобные фразы.

2.3 ОСНОВНЫЕ ТРЕБОВАНИЯ ГОСТ К ОФОРМЛЕНИЮ ТЕКСТА

Перечисляемые далее требования соответствуют правилам оформления выпускной квалификационной работы.

1. В качестве основного шрифта используется шрифт Times New Roman. Размер шрифта зависит от формата листа и требований издательств к форматированию текста. Размер может быть 14 пт., 12 пт. и даже 10 пт. В отчетах в качестве основного шрифта Вы должны **использовать шрифт 14 пт.** При этом **названия таблиц и рисунков (они обязательны!)** пишутся шрифтом на 2 пт меньше основного, то есть шрифтом 12 пт. Таким же шрифтом (12 пт) заполняется содержимое таблиц. **Межстрочный интервал** должен быть равным **1,5 строки**. Текст абзацев выравнивать «**по ширине**».

2. **Латинские буквы**, обозначающие переменные, пишутся *курсивным шрифтом*. **Латинские буквы**, обозначающие векторы, пишутся *полужирным курсивным шрифтом*. **Латинские буквы**, обозначающие матрицы, множества, пишутся **ПРЯМЫМ ПОЛУЖИРНЫМ ШРИФТОМ ПРОПИСНЫМИ БУКВАМИ**. Греческие буквы, символы операций и отношений, а также цифры пишутся прямым шрифтом (как основной текст документа).

3. Названия таблиц пишутся над таблицами. Строго над таблицами, то есть текст названия не должен выходить за габариты таблицы! При этом правая граница названия должна быть выровнена по правой границе таблицы! В конце названия **точка не ставится**.

4. По поводу рисунков используется понятие «подрисуночная надпись». То есть, название рисунка пишется под рисунком. Строго под рисунком! При этом слово «Рисунок» пишется полностью. В конце названия **точка не ставится**. Название рисунка можно центрировать в строке, можно писать выровненным по левой границе рисунка. Длинные названия пишутся в несколько строк. Примеры смотрите в этом документе.

5. Рисунки и таблицы размещаются по центру строк. Каждая таблица и рисунок «имеют право» на существование в тексте только тогда, когда им предшествует прямая ссылка (упоминание) в тексте.

6. Таблицы и рисунки отделяются от текста сверху и снизу пустой строкой.

7. Пункты оглавления (названия разделов, подразделов, параграфов ...) пишутся без точки в конце. Они могут размещаться по центру либо с абзацным отступом.

8. Пункты оглавления отделяются как от предшествующего, так и от озаглавливаемого текста интервалом большим, чем межстрочный. При этом интервал от предшествующего текста должен быть чуть больше, чем интервал перед озаглавливаемым текстом.

2.4 СТРУКТУРА ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

Пояснительная записка должна иметь следующую структуру.

1. Титульный лист (рисунок 2.1).
2. Содержание записки.

3. Задание, содержащее как исходные данные, так и требования, изложенные в данных методических указаниях.

4. Авторскую оценку соответствия качества проекта объявленным требованиям.

5. Математическую постановку задачи:

– описание структур разрядных сеток, используемых для представления всех исходных, результирующих и вспомогательных кодов с указанием типа кодов (прямой, обратный, дополнительный, модифицированный);

– описание особенностей алгоритма, заданного в виде ГСА. (Пример такого описания приведен ниже);

– привлекаемые для решения задачи методы (например, методы теории цифровых автоматов; методы моделирования цифровых схем с использованием двоичного (или иного) алфавита моделирования; теорию и методы булевой алгебры, автоматный подход к проектированию программ и т.д.), ссылки на основные алгоритмы, описанные в последующих пунктах пояснительной записки.

<p>Минобрнауки РФ федеральное государственное бюджетное образовательное учреждение высшего образования «Самарский государственный технический университет»</p> <p>Кафедра <i>"Вычислительная техника"</i></p> <p>КУРСОВАЯ РАБОТА по дисциплине <i>"Моделирование"</i> на тему "Разработка среды для имитационного моделирования операционного устройства".</p> <p>Выполнил(а) _____ Проверил _____ Оценка _____</p> <p><i>Самара 2020</i></p>

Рисунок 2.1- Форма титульного листа пояснительной записки

6. Описание структурной схемы УА с объяснением назначений ее компонент и логическим содержанием всех кодов. Поскольку УА должен быть автоматом модели Мили, то в общем случае в структурной схеме УА должна быть память (регистр) для хранения значений логических условий. Необходимо объяснить назначение этой памяти и аргументировать необходимость хранения каждого из записываемых в нее логических условий. При отсутствии такой необходимости значение логического условия не следует хранить в такой памяти.

7. Описание типов моделей компонент УА и ОА.

8. Описание процесса проектирования. Описание процесса проектирования моделей компонент УА и ОА должно содержать таблицы переходов УА, описание кодирования микроопераций и логических условий, описание разметки состояний УА на ГСА, все необходимые логические выражения.

Временные диаграммы, иллюстрирующие процессы распространения сигналов в схемах УА и ОА.

Описание содержания модели одного такта работы ОУ с указанием последовательности обращений к моделям компонент ОУ.

9. Описание программной реализации. Описание программной реализации должно содержать:

- требования к операционной системе (Windows 2000, ...);
- требования к составу и характеристикам оборудования (например: IBM PC совместимый компьютер класса Pentium 2 или выше с видеосистемой, поддерживающей графический режим True Color с пространственным разрешением не хуже, чем 1024×768 пикселей с видеопамятью объемом не менее 32 Мбайт и т.п.);
- указание использованного языка программирования;
- описание комплекта файлов, необходимых для запуска программы (*.exe файлы, драйверы и др.) и структуры размещения файлов по папкам;

– описание структуры программы в виде состава модулей и процедур с кратким описанием их назначения, состава параметров, взаимодействия;

– описание структуры данных;

– описание основных алгоритмов с привязкой к описанным ранее методам, модулям, процедурам и структурам данных;

– листинг основных структур данных и подпрограмм: тексты программ в записке лучше набирать шрифтом типа Courier.

10. Описание пользовательского интерфейса и инструкция по установке и запуску программы.

11. Библиографический список.

12. Содержание записки.

2.5 ОСНОВНЫЕ ТРЕБОВАНИЯ К ПРОГРАММНОЙ РЕАЛИЗАЦИИ

1. Необходимо полностью учесть содержание задания.

2. Выбрать среду реализации в соответствии с п. 2.1.

3. Программа не должна завершаться аварийно при любых действиях пользователя и при любых исходных данных.

4. Использовать модульный принцип программирования.

5. Текст программы должен содержать комментарии к используемым модулям, процедурам, структурам данных, основным переменным и шагам алгоритма. Строки текста писать с отступом от начала строки, подчеркивая подчиненность процедурам, функциям, операторам цикла и операторным скобкам.

6. Модели микропрограммы во всех случаях должны давать правильный результат арифметической операции. Если в исходной ГСА имеются какие-либо логические ошибки, то обнаружить их и исправить – задача автора курсовой работы.

3 МЕТОДИКА ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

Для сокращения последующего текста введем некоторые условные сокращения терминов, касающихся микропрограмм:

МП – микропрограмма;

МК – микрокоманда;

МО – микрооперация.

При выполнении курсовой работы рекомендуется придерживаться следующей последовательности действий.

1. Изучить (вспомнить) теоретический материал, изложенный в [2, 3]. При этом можно ограничиться кругом вопросов, связанных с выбранным вариантом арифметической операции.

2. В результате внимательного изучения заданного варианта МП выявить основные признаки реализованного в МП алгоритма и описать их. Например, на рисунке 3.1 показана МП выполнения операции деления. Перечислим основные признаки этой МП.

2.1. МП вычисляет частное $C := A / B$.

2.2. В МП предусмотрен исход «Переполнение разрядной сетки». Переполнение фиксируется, если $|A| \geq |B|$. Это означает, что частное должно быть по модулю строго меньше 1. Поскольку вычисленное частное в следующей команде программы может оказаться делимым или делителем, то естественно принять, что операнды A и B также должны быть по модулю строго меньше 1. Последнее замечание определяет структуру разрядной сетки для представления операндов и результата операции.

2.3. В МП предусмотрен ускоренный вариант получения частного в случае, когда делимое равно 0.

2.4. В МП операнды предполагаются представленными в прямых кодах.

2.5. В МП количество повторений цикла всегда на 1 больше количества вычисляемых разрядов частного, поэтому при завершении циклической части выполняются МО по округлению вычисленного частного.

2.7. В МП реализован алгоритм деления с восстановлением остатка. Но восстановление остатка выполняется не за счет МО вида $A := A + B$ («Остаток» := «Остаток» + «Делитель»), а за счет запоминания текущего значения остатка во вспомогательной переменной D и использования на следующем цикле в МО сдвига кода остатка либо только что вычисленного нового остатка, хранящегося в переменной C , либо предыдущего остатка, запомненного в переменной D .

2.8. В МП использован алгоритм деления со сдвигом кода остатка влево.

Выявленные особенности реализованного алгоритма дают более адекватное представление о микропрограмме и позволяют организовать процесс разработки моделей в правильном направлении.

3. Выполнить разметку состояний на ГСА, соответствующую состояниям управляющего автомата модели Мили. Этот этап выполнения работы требует привлечения знаний, полученных при изучении курса «Прикладная теория автоматов». Размеченные состояния автомата необходимы как для проектирования модели УА, так и для организации режима пошагового выполнения МП.

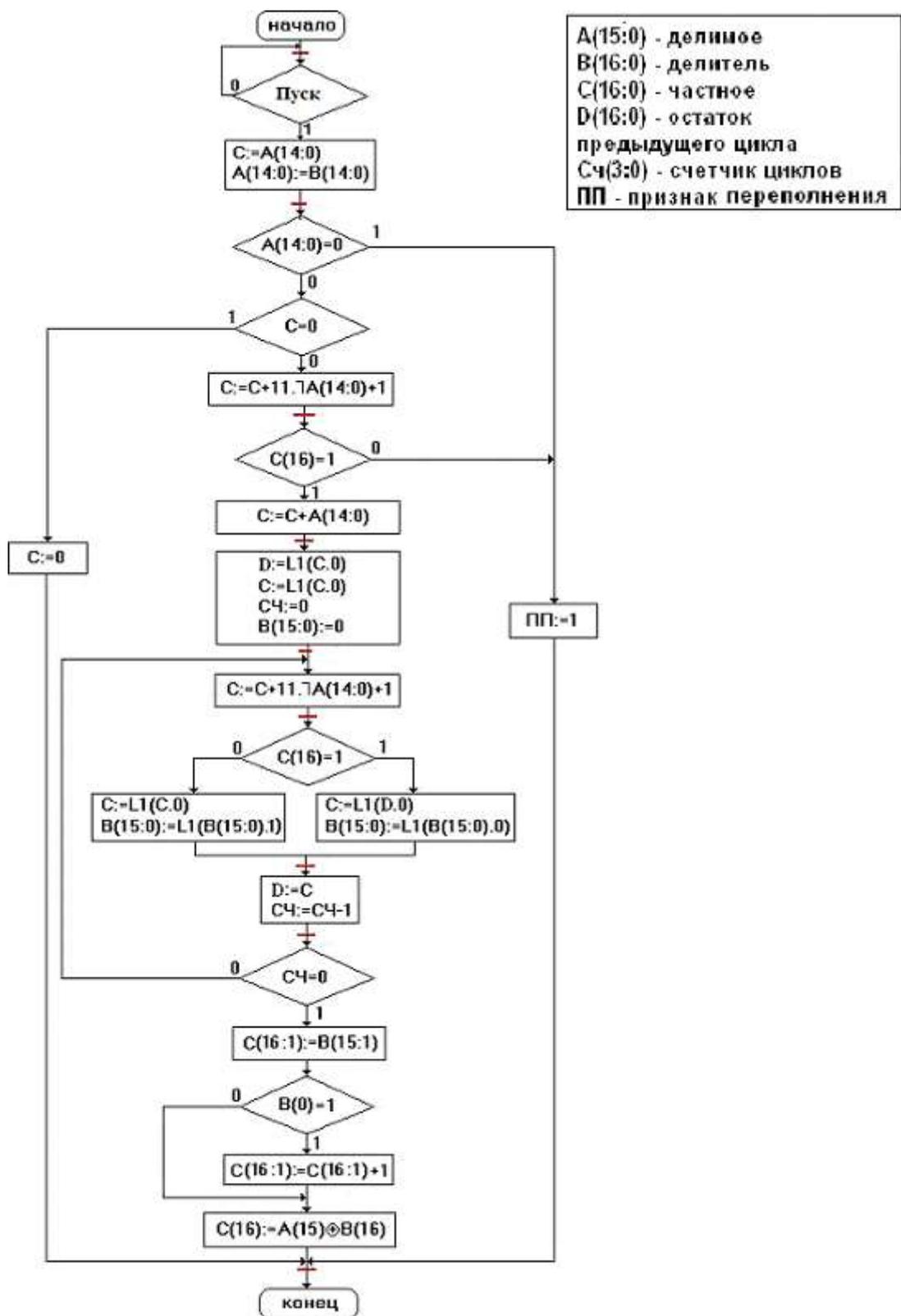


Рисунок 3.1- Пример МП
выполнения арифметической операции

4. Разработать **концептуальную модель** проекта. На уровне концептуальной модели **проектные решения** должны носить общий характер. Например:

- позволять или не позволять пользователю вводить исходную строку с помощью клавиатуры;
- в какой форме указывать пользователю очередной шаг алгоритма;
- если программа указывает конкретное состояние на ГСА, то отображаемые на форме значения всех кодов равны их значению до выполнения МО, инициируемых данным состоянием, или после выполнения этих МО;
- отображать или не отображать десятичные эквиваленты кодов;
- в структуре УА использовать или не использовать дешифратор кодов состояний;
- предусматривать модульную структуру или ограничиться одним модулем;
- интерфейсные средства, соответствующие двум уровням моделирования МП, представить на разных закладках или все расположить на одной закладке (рисунок 3.2) и т.п.

4. На основе проектных решений концептуального характера разработать интерфейсные средства программы. Для отображения кодов рекомендуется использовать компоненты типа «Таблица».

5. Разработать модели всех МО. При этом в качестве моделей обрабатываемых кодов использовать только целочисленные типы данных языка программирования. Учесть возможные неожиданные последствия использования типов данных со знаком.

При разработке моделей МО следует учитывать, что МО выполняются соответствующими операционными элементами, поэтому подход к содержанию МО с точки зрения программиста часто может привести к неверному ее толкованию, к ошибкам в моделировании. Так, например, в МП (см. рис. 3.1) имеется МО « $Sч := Sч - 1$ », которая первый раз выполняется при значении $Sч=0$. С позиций программиста после вычитания 1 из 0 получается число -1, но операционный элемент типа «счетчик» из состояния 0 при вычитании 1 перейдет в со-

стояние, соответствующее самому большому числу, которое может отобразить этот счетчик. В приведенной МП счетчик имеет 4 разряда, поэтому из состояния 0 при вычитании 1 он перейдет в состояние 15. В качестве модели этого счетчика следует взять переменную типа byte, поэтому указанный переход счетчика должен привести к установке этой переменной в состояние, задаваемое следующим двоичным кодом: 00001111.

6. Имея модели МО и размеченную ГСА, можно приступить к программированию модели МП.

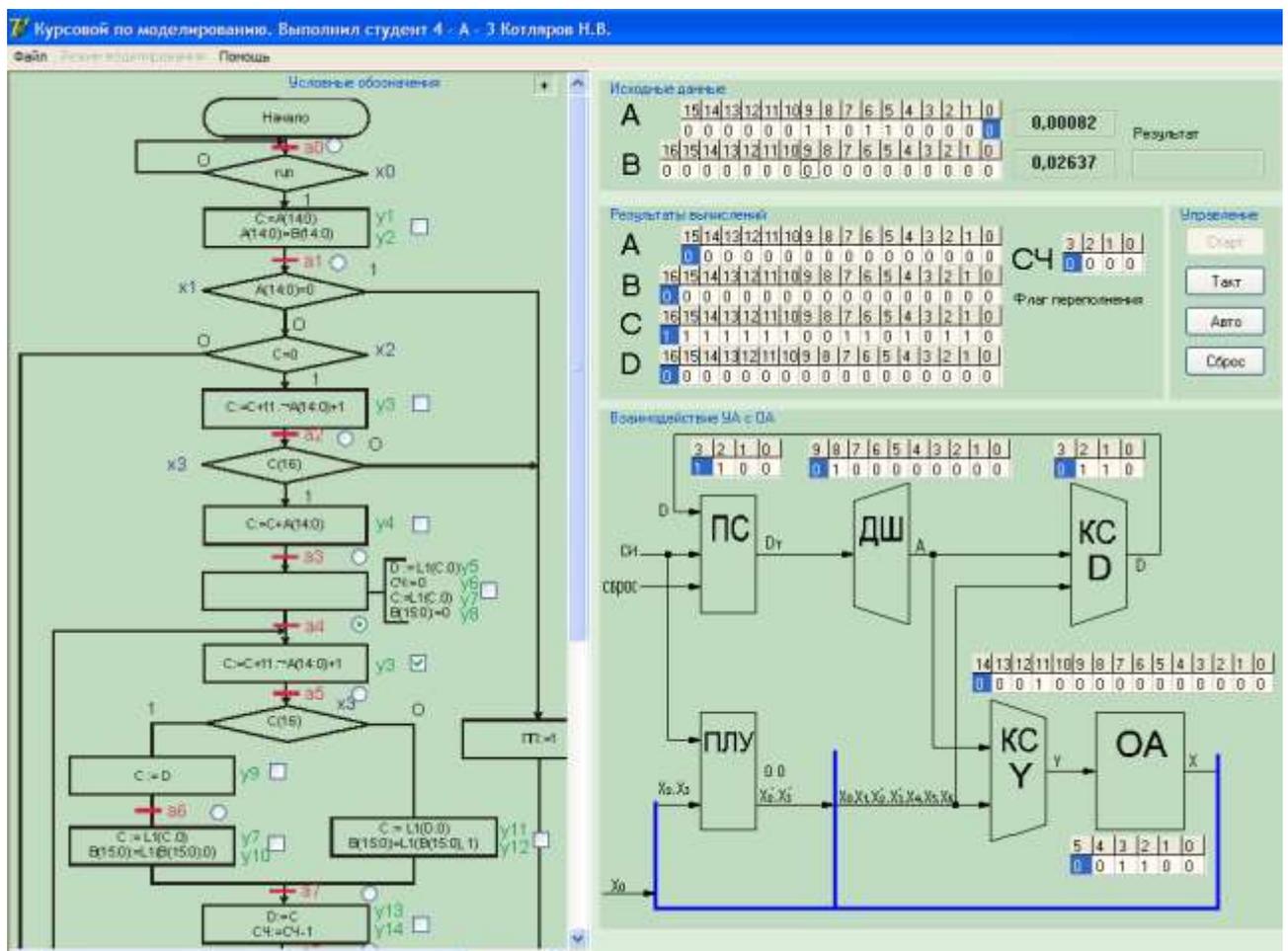


Рисунок 3.2- Пример интерфейса среды моделирования МП

7. Выполнить знакомые по курсу «Прикладная теория цифровых автоматов» шаги проектирования УА, ограничившись получением логических выражений, описывающих функционирование регистров и комбинационных схем, входящих в состав УА. Раскры-

вать структуру этих схем не требуется. Таким образом, компоненты УА моделируются на уровне функциональных моделей.

8. Понимая последовательность прохождения сигналов через схемы УА и моменты востребованности кодов, формируемых различными компонентами УА, разработать последовательность обращений к моделям компонент УА и ОА. Выполняя этот этап проектирования, студент должен продемонстрировать свое понимание изученных курсов «Схемотехника», «Прикладная теория цифровых автоматов», «Теория проектирования ЭВМ». Готовый «рецепт» здесь не дается. Неправильно спроектированная последовательность вызова моделей компонент может сделать бессмысленным наличие в структуре УА памяти логических условий. Бессмысленной потому, что при ее отсутствии в структуре УА поведение модели УА будет точно таким же, как при наличии этой памяти в составе УА. Поскольку такая память может оказаться необходимой по объективным причинам, то безразличие модели к ее присутствию либо отсутствию в структуре УА говорит о недостаточной адекватности модели моделируемому объекту.

9. При моделировании ОУ на уровне МП модель отдельной МК, содержащей несколько МО, может представлять собой заданную программистом последовательность вызовов моделей конкретных МО. При моделировании на уровне отображения взаимодействия УА и ОА модель УА формирует вектор Y выходных сигналов. Модель ОА должна выполнять анализ содержимого этого вектора, просматривая значения его компонент. При обнаружении компоненты, равной 1, модель ОА должна инициировать обращение к модели соответствующей МО. При разработке этих моделей следует принять меры к тому, чтобы конечный результат исполнения модели МК не зависел от относительных значений номеров, присвоенных отдельным МО при выполнении их кодирования. Иными словами, любые изменения последовательности вызовов моделей МО, составляющих данную МК, не должны как-либо влиять на конечный результат исполнения МК. В реальном ОА все МО данной

МК выполняются одновременно различными операционными элементами в составе ОА, поэтому результат исполнения любой из таких МО не может быть использован на этом же такте другими МО данной МК. В модели имеет место совершенно иная ситуация. Процессы, протекающие одновременно, любая программа на однопроцессорной вычислительной системе может моделировать только последовательно во времени. В результате в моделях может проявиться зависимость результатов моделирования МО в составе МК от последовательности обращений к моделям отдельных МО в составе МК. Это возможно, например, в МК, состоящей из следующих МО (пример из МП на рис. 3.2):

$$C[14..0]:= A[14..0];$$

$$A[14..0]:= B[14..0].$$

При непродуманной модели МК изменение последовательности вызова моделей приведенных МО может привести к тому, что получатся следующие значения кодов: $C[14..0] = A[14..0] = B[14..0]$. Ошибочность подобного совпадения трех кодов станет очевидной, если учтем, что в микрооперациях в правой части записаны коды до выполнения МК, а в левой части – коды после выполнения МК. Поэтому содержание МО более точно передают несколько иные их описания. Обозначим индексами t коды до выполнения МК, а индексами $t+1$ – коды после выполнения МК. Тогда приведенные МО следует понимать так:

$$C_{t+1}[14..0]:= A_t[14..0];$$

$$A_{t+1}[14..0]:= B_t[14..0].$$

Теперь ясно, что в составе этой МК указаны два совершенно разных кода, наполняющих регистр A , поэтому код с регистра B в принципе не может через регистр A попасть на регистр C . Эту особенность поведения реального ОА необходимо правильно смодели-

ровать. Решение этой проблемы за счет какой-либо особой нумерации МО (первой из МО в составе МК присвоим меньший номер, чем второй из них, в результате эти две МО будут моделироваться в последовательности, исключающей передачу кода из регистра *B* в регистр *C* через регистр *A*) будет оценено как неудовлетворительное.

Необходимо выполнить следующее требование к логике моделирования:

любые изменения номеров, присвоенных отдельным МО на этапе их кодирования, не должны как-либо влиять на результаты работы модели ОА.

10. При разработке моделей МО необходимо заботиться о том, чтобы части кодов, не указанные в левой части МО, не изменяли своего значения в результате выполнения этой МО. Например, МО

$$Am[31:15]:= A[14:0]$$

предполагает, что часть $Am(14:0)$ кода на регистре Am остается неизменной. Модель этой МО в виде процедуры может иметь следующий вид (язык Delphi):

```
procedure Y1; {Am(31:15):= A(14:0)}  
begin  
  Am:= ((A and $00007FFF) shl 15) or (Am and $00007FFF)  
end;
```

В операторе $\$00007FFF$ задает константу арифметики по основанию 16, равную 7FFF, представленную на целочисленной разрядной сетке длиной 32 бита. shl – операция сдвига кода влево: sh – shift; l – left.

11. Проверить работу разработанной среды при всех значимых вариантах исходных данных.

12. После отладки значимого фрагмента программного кода можно приступить к составлению пояснительной записки в части, отражающей отлаженный фрагмент программы.

4. ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ВОПРОСОВ К ЗАЩИТЕ КУРСОВОЙ РАБОТЫ

1. Определить основные требования со стороны разработанного продукта к аппаратной и программной комплектации компьютера.

2. Классифицировать разработанную модель.

3. Характеризовать принципиальные отличия моделей имитационных от моделей аналитических.

4. Показать по исходному тексту программы и объяснить использованную в программе структуру данных с указанием соответствия элементов структуры данных элементам и параметрам моделируемого объекта.

5. Объяснить структуру разработанного программного продукта на уровне модулей, процедур и функций.

6. Объяснить взаимосвязь указанной подпрограммы с другими программными единицами.

7. Показать по исходному тексту программы и объяснить использованную в программе структуру данных.

8. Характеризовать методы моделирования, использованные при разработке моделей компонент системы.

9. Объяснить какой алфавит моделирования был использован и почему. Привести примеры других алфавитов, используемых при моделировании компонент вычислительной техники.

10. Объяснить отличие структурного моделирования от функционального.

11. Характеризовать алгоритм, реализуемый исходной микропрограммой, представленной в виде ГСА.

12. Продемонстрировать работу разработанной среды на конкретных числовых данных с объяснением структур использованных разрядных сеток, интерпретацией исходных промежуточных и результирующих кодов с объяснением процесса исполнения моделируемой микропрограммы.

13. Объяснить последовательность событий в течение одного такта работы операционного устройства (ОУ).

14. Объяснить модели микроопераций, указанных преподавателем.

15. Объяснить структурную схему управляющего автомата и его поведение в течение одного такта работы ОУ.

5. ГРАФИК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

Студент должен заниматься курсовым проектированием в течение всего семестра, не откладывая работу «на потом». Поэтому необходимо придерживаться примерно следующего графика выполнения курсовой работы (таблица 5.1).

Таблица 5.1 – График выполнения курсовой работы

Примерный процент выполнения КР	Срок выполнения
Программная реализация: – 25-30% выполнения; – 50% выполнения; – 100% выполнения	8 – 10 недели; 11-12 недели; 15-16 недели

Окончание таблицы 4.1

Примерный процент выполнения КР	Срок выполнения
Пояснительная записка: – 25% выполнения; – 50% выполнения; – 100% выполнения.	12-13 недели; 13-14 недели; 15-16 недели.
Защиты курсовой работы	16 – 18 недели

Соблюдение графика проектирования контролируется преподавателем-руководителем курсовой работы. Контроль соблюдения графика осуществляется на лабораторной работе, ближайшей к указанному в графике сроку.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Кнут, Д.* Искусство программирования для ЭВМ. Т. 1 Получисленные алгоритмы / *Д. Кнут*; пер. с англ. под ред. *Ю.М. Баяковского*. – М.: Мир, 1978. – 839 с.
2. *Орлов, С.П.* Арифметика ЭВМ и логические основы переключательных функций: учеб. пособ. / *С.П. Орлов, Б.В. Мартельянов*. – Самара: Самар. гос. тех. ун-т, 2004 – 197 с.
3. *Орлов, С.П.* Арифметика ЭВМ и логические основы переключательных функций: учеб. пособ. / *С.П. Орлов, Б.В. Мартельянов*. 3-е изд., испр. и доп. – М.: Машиностроение – 1, 2005. – 256 с.
4. *Воронцов, И.В.* Теория автоматов и формальных языков: лабораторный практикум / *И.В. Воронцов, С.П. Орлов*. - Самара: Самар. гос. тех. ун-т, 2013. – 156 с.: с ил.

ЗАДАНИЯ НА ЛАБОРАТОРНЫЕ РАБОТЫ

ЛАБОРАТОРНАЯ РАБОТА №1 РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ КОМПИЛЯЦИИ АЛГЕБРАИЧЕСКОГО ВЫРАЖЕНИЯ ПО АЛГОРИТМУ ДИЙКСТРА

Трудоемкость задачи – 8 ч.

Цель работы – развитие математического кругозора и математической культуры, получение навыков формализации и разработки моделирующих алгоритмов, закрепление отдельных теоретических положений раздела курса.

Задание. Разработать среду для моделирования преобразования инфиксной (скобочной) формы записи алгебраических выражений в постфиксную (бесскобочную) форму (в обратную польскую запись) по алгоритму Дийкстры.

Принять, что в исходной инфиксной форме операции одного уровня приоритета исполняются слева направо.

Требования к организации интерфейса

Предусмотреть два режима выполнения преобразования – автоматический и пошаговый.

В процессе преобразований отображать:

- исходное выражение в инфиксной форме в неизменном виде;
- исходное выражение в процессе удаления из него очередного символа, переданного в стек или в выходную строку;
- выходную строку, задающую выражение в постфиксной форме, в состояниях, отображающих процесс последовательного удлинения этой строки по мере передачи в нее очередных символов из стека или из исходной инфиксной строки;
- содержимое стека.

Ограничения на сложность задачи

1. Длина исходной строки символов не превышает 80.

2. Используются 4 арифметические операции (+, -, *, /) и 4-5 функций, причем 2-3 из этих функций должны иметь ограниченную область определения.

3. В исходном выражении числовые константы отсутствуют, а имена всех переменных состоят ровно из одного символа.

4. Символ «-» используется только как символ бинарной операции вычитания.

5. Корректные выражения задаются по правилам языка Паскаль: каждая операция задается явно без использования правил умолчания.

6. Стек можно организовать и как список, и как массив. Необходимо исключить из программы попытки произвольного обращения к содержимому стека. Обращение допускается только через указатель стека, причем индексное выражение с участием указателя стека может состоять только из имени указателя стека. Более сложные индексные выражения с участием указателя стека недопустимы.

Дополнительные требования

1. Исходное выражение может содержать синтаксические ошибки как в скобочной структуре, так и в порядке следования символов. Некоторые примеры возможных ошибок даны в табл. 1.

2. Синтаксическая ошибка любого типа должна быть обнаружена и указана в соответствующем комментарии. Процесс преобразования прекращается при обнаружении первой синтаксической ошибки.

Все лабораторные работы в рамках данной дисциплины предполагают, что студенты владеют средствами разработки Windows приложений, поэтому приводимая далее методика не является руководством по программированию.

Выражение	Ошибка
...A +* d...	Подряд два символа операций
...A d...	Подряд две переменные
...A (...	Между переменной и скобкой отсутствует символ операции

Методика выполнения работы

1. Изучить теоретический материал, изложенный в гл. 1 данного пособия.

2. Разработать концептуальную модель проекта. На уровне концептуальной модели проектные решения должны носить общий характер. Например:

– принимать или не принимать во внимание отдельные рекомендации, изложенные в теоретическом разделе;

– позволять или не позволять пользователю вводить исходную строку с помощью клавиатуры;

– в какой форме указывать пользователю на содержимое и результат очередного шага преобразования;

– если программа указывает на конкретный шаг преобразования, то содержимое входной строки, выходной строки и стека – это то, что было до выполнения указанного шага, или это результаты выполнения шага;

– какие математические функции будут доступны пользователю и т.п.

3. Разработать интерфейсные средства программы, руководствуясь в качестве примера изображением экранной формы (рис. 1 и 2). В рамках данной работы возможно использование интерфейса, значительно более простого, чем показано в примере. Обязательное содержимое интерфейсных средств оговорено в сформулированном выше задании на лабораторную работу.

4. Разработать структуру проекта на уровне модулей.

5. Разработать обработчики событий и все вспомогательные подпрограммы.

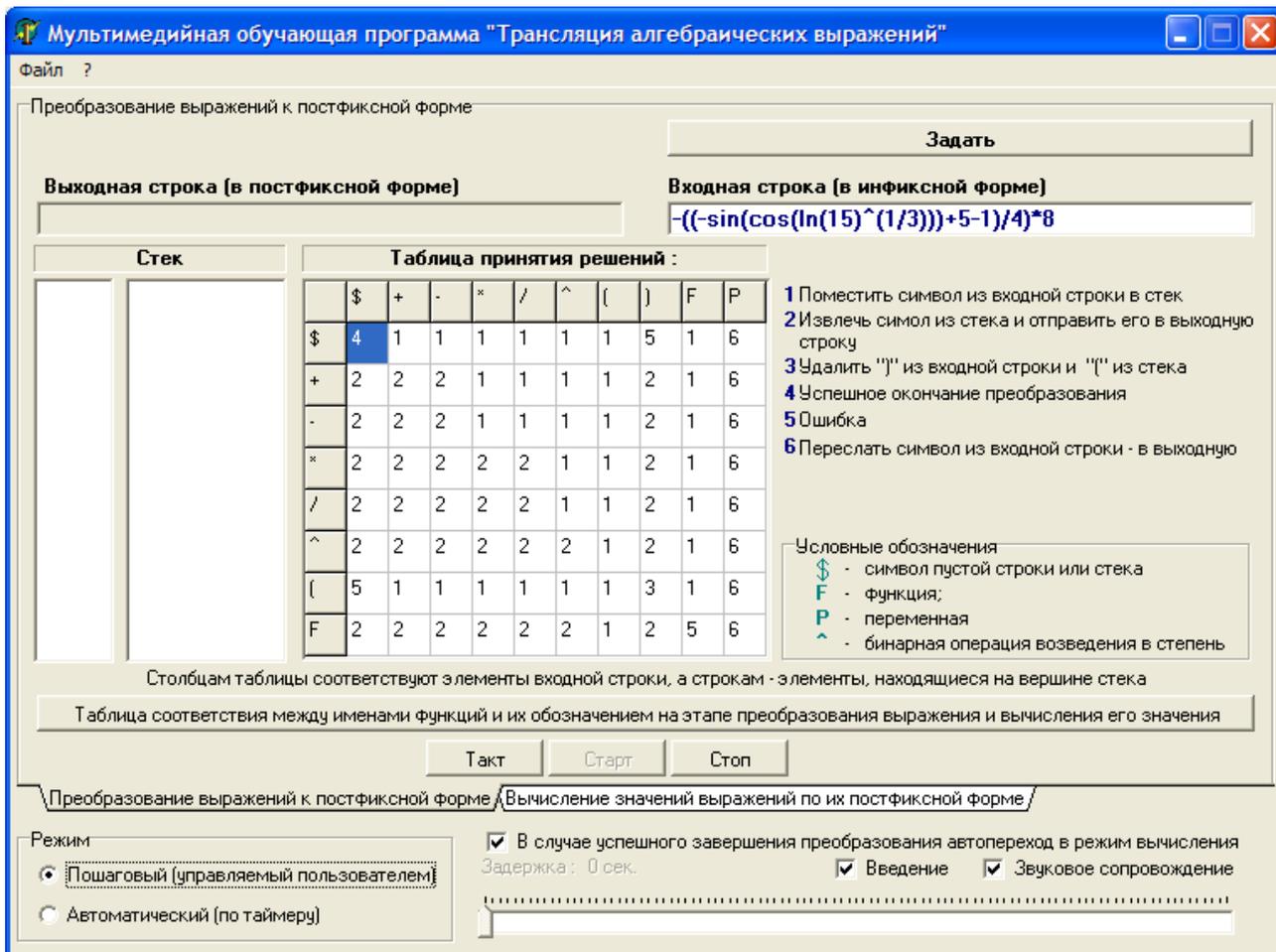


Рис. 1. Главная экранная форма мультимедийного приложения

Примечания

1. Существует понятие «Дно стека». При графическом изображении стека это «Дно» должно задавать нижнюю, а не верхнюю границу области стековой памяти.

2. Информация в стеке накапливается от его дна вверх.

3. Запись информации в стек и ее чтение из стека не вызывает физического перемещения в памяти ранее записанных данных. Это означает, что на форме символы в области стековой памяти не должны двигаться по экрану.

4. Отображение каким-либо визуальным средством указателя стека в состоянии, указывающем на вершину стека, обязательно.

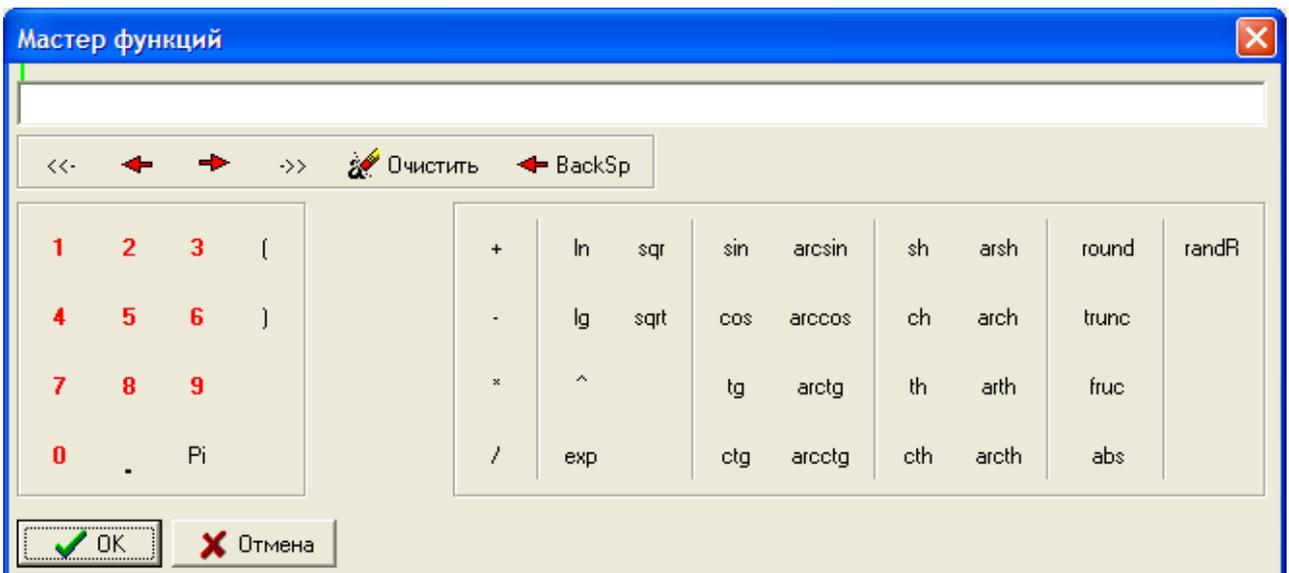


Рис. 2. Экранная форма мультимедийного приложения для ввода исходного выражения

5. Следует понимать, что при считывании информации из стека содержимое ячеек стека не модифицируется, поэтому всякие «обнуления» ячеек стека выполнять запрещается, поскольку такие «обнуления» отсутствуют в реальном стеке и логически излишни. Логика работы стековой памяти такова, что для ее «обнуления» достаточно указатель стека установить на его дно.

6. Для ввода исходного выражения полезно предусмотреть кнопочное меню с символами переменных, арифметических операций и выбранных алгебраических функций. Ввод исходной инфиксной строки с помощью клавиатуры запретить, что не позволит пользователю задавать неразрешенные функции и непредусмотренные имена переменных.

7. Спланировать вычислительные эксперименты для отладки программы и демонстрации преподавателю в процессе отчета по работе.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем особенности инфиксной, постфиксной и префиксной форм записи алгебраических выражений?

2. Каковы недостатки инфиксной формы записи алгебраических выражений?

3. Каковы достоинства постфиксной и префиксной форм записи алгебраических выражений?

4. Что обозначает термин «стек» ?

5. Что такое указатель стека?

6. Как программа управляет указателем стека в режимах записи и чтения?

7. Каково назначение стека в алгоритме Дейкстры?

8. Какой метод программирования положен в основу алгоритма Дейкстры?

9. Объяснить содержание указанного преподавателем шага алгоритма Дейкстры. Показать его реализацию в программе.

10. Заданное выражение представить в трех формах записи.

11. С какой целью выполняется кодирование имен функций и числовых констант?

12. При указании на очередной шаг процесса преобразования на форме состояния объектов отображают то, что есть до выполнения шага, или то, что получилось после выполнения этого шага. Высказанный ответ подтвердить экспериментально.

13. Объяснить, какие компоненты среды разработки (языка программирования) использованы для отображения интерфейсных средств.

Преподаватель может задать любой иной вопрос, относящийся к теоретическим положениям рассматриваемой темы, к затрагиваемым общим вопросам теории моделирования, рассмотренным к данному моменту в лекционном материале, к любым сторонам вопросов программной реализации задачи.

ЛАБОРАТОРНАЯ РАБОТА №2

РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ВЫЧИСЛЕНИЯ ЗНАЧЕНИЯ АЛГЕБРАИЧЕСКОГО ВЫРАЖЕНИЯ, ПРЕДСТАВЛЕННОГО В ПОСТФИКСНОЙ ФОРМЕ ЗАПИСИ

Трудоемкость задачи – 8 ч.

Цель работы – развитие математического кругозора и математической культуры, получение навыков формализации и разработки моделирующих алгоритмов, закрепление отдельных теоретических положений раздела курса.

Задание. Разработать среду для моделирования вычисления значений логических выражений, представленных в постфиксной форме записи (в форме обратной польской записи). В качестве исходного выражения в постфиксной форме должно быть взято выражение из среды, разработанной при выполнении предыдущей работы.

Требования к организации интерфейса

Предусмотреть два режима выполнения преобразования – автоматический и пошаговый.

На форме предусмотреть поля для задания числовых значений использованных переменных.

В процессе преобразований отображать:

- исходное выражение в постфиксной форме в неизменном виде;
- изменяющуюся строку, отображающую удаление символов из выражения по мере использования их для вычислений;
- содержимое стека.

Следует понимать, что при считывании информации из стека содержимое ячеек стека не модифицируется, поэтому всякие «обнуления» ячеек стека выполнять запрещается, поскольку такие «обнуления» отсутствуют в реальном стеке и логически излишни.

Ограничения на сложность задачи

1. Длина строки символов, задающей выражение, не превышает 80 символов.

2. Используются только 4 арифметические операции: +, -, *, / и 4-5 функций, причем 2-3 из этих функций должны иметь ограниченную область определения.

3. Считаем, что все возможные синтаксические ошибки в выражении выявлены на этапе получения постфиксной формы.

Дополнительные требования

1. При вычислении значения выражения может возникнуть попытка деления на ноль. Такая ситуация должна вызвать появление соответствующего комментария и прекращение дальнейших вычислений с сохранением на экране текущего содержимого отображаемых данных: исходной строки, стека.

2. Перед вычислением значения функции необходимо проверять аргумент на принадлежность области определения функции.

Все лабораторные работы в рамках данной дисциплины предполагают, что студенты владеют средствами разработки Windows приложений, поэтому приводимая далее методика не является руководством по программированию.

Методика выполнения работы

1. Изучить теоретический материал, изложенный в гл. 2 данного пособия.

2. Разработать концептуальную модель проекта. На уровне концептуальной модели проектные решения должны иметь общий характер.

3. Разработать интерфейсные средства программы, руководствуясь в качестве примера изображением экранной формы (рис. 1). В рамках данной работы возможно использование интерфейса, значительно более простого, чем показано в примере. Обязательное содержимое интерфейсных средств оговорено в сформулированном выше задании на лабораторную работу.

4. Разработать структуру проекта на уровне модулей.

5. Разработать обработчики событий и все вспомогательные подпрограммы.

6. Учесть перечисленные ниже замечания.

При выполнении лабораторной работы необходимо учесть следующее.

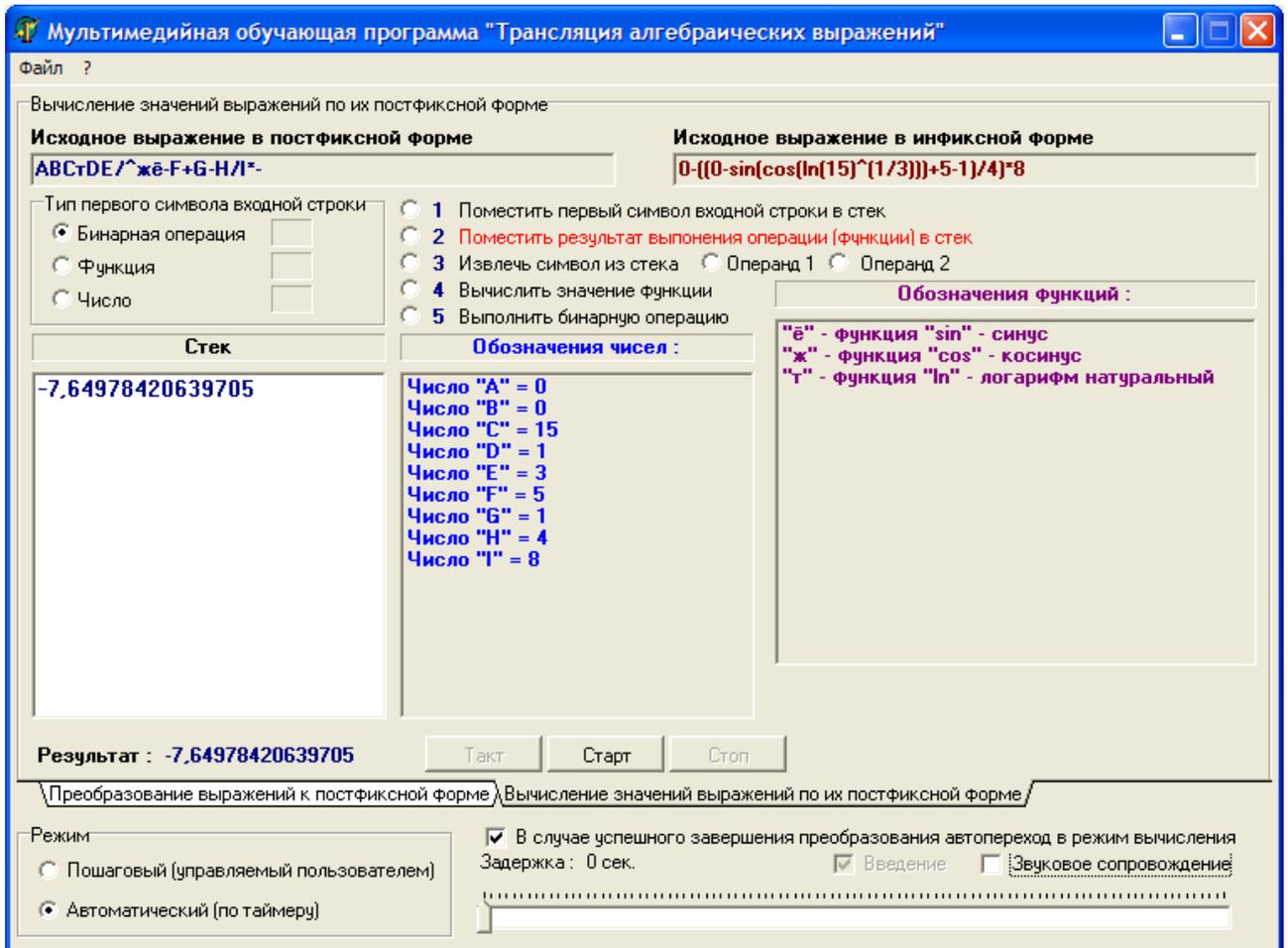


Рис. 1. Экранная форма мультимедийного приложения

1. Существует понятие «Дно стека». При графическом изображении стека это «Дно» должно задавать нижнюю, а не верхнюю границу области стековой памяти.

2. Информация в стеке накапливается от его дна вверх.

3. В рассмотренном в гл. 2 примере на многих рисунках отображаются числа в ячейках стека, расположенных выше той, на которую показывает указатель стека. Эти числа недоступны для обра-

ботки, что следует из сущности стековой организации памяти. Изображение этих чисел имело целью подчеркнуть:

- считывание данных из стека не разрушает информацию в нем: при считывании указатель стека перемещается в направлении дна стека и в ячейках выше указываемой остаются ранее записанные данные;
- ранее записанные в ячейку стека данные могут быть изменены только в результате записи в эту ячейку очередного операнда выражения или вычисленного промежуточного результата.

4. Учесть все замечания из предыдущей работы, относящиеся к стеку.

На все особые ситуации, не оговоренные в задании, программа может реагировать так, как пожелает автор.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем особенности инфиксной, постфиксной и префиксной форм записи алгебраических выражений?

2. Каковы недостатки инфиксной формы записи алгебраических выражений?

3. Каковы достоинства постфиксной и префиксной форм записи алгебраических выражений?

4. Что обозначает термин «стек»?

6. Каково назначение стека в алгоритме вычисления значения выражения?

7. В чем отличия стеков, использованных в данной и в предыдущей работах?

8. Вычислить вручную (можно пользоваться калькуляторами) значение арифметического выражения, заданного в постфиксной форме.

9. Объяснить содержание указанного преподавателем шага алгоритма вычисления. Показать его реализацию в программе.

10. Какова роль скобок в постфиксной форме записи выражения?

Преподаватель может задать любой иной вопрос, относящийся к теоретическим положениям рассматриваемой темы, к затрагиваемым общим вопросам теории моделирования, рассмотренным к данному моменту в лекционном материале, к любым сторонам вопросов программной реализации задачи.

ЛАБОРАТОРНАЯ РАБОТА №3
РАЗРАБОТКА И ИССЛЕДОВАНИЕ ГЕНЕРАТОРОВ
ПСЕВДОСЛУЧАЙНЫХ ЧИСЛОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Трудоемкость задачи – 8 ч.

Цель работы – закрепление теоретических положений темы и получение навыков разработки и исследования генераторов псевдослучайных числовых последовательностей.

Задание. Разработать среду для исследования генератора псевдослучайных числовых последовательностей, встроенного в используемый Вами язык программирования, и разработанного собственного генератора, реализующего метод Лемера. Провести исследования генераторов средствами разработанной среды. Реализовать статистический метод приближенного вычисления числа π .

Методика выполнения

1. Изучить теоретический материал, изложенный в лекциях.
2. Разработать концептуальную модель проекта. На уровне концептуальной модели проектные решения должны носить достаточно общий характер.
3. Разработать интерфейсные средства программы. Пример «ленивого» интерфейса представлен на рисунке 1.
4. Сгенерировать заданное количество n ($100 \leq n \leq 10000$ и задается после запуска программы в виде содержимого поля ввода) псевдослучайных целых чисел диапазона $[0; 99]$.
5. Вычислить статистическое среднее значение (математическое ожидание) и дисперсию сгенерированной последовательности. Вычисленные значения числовых характеристик отобразить на форме. В рамках данной работы нет необходимости запоминать сгенерированную последовательность чисел. Данные, необходимые для вычисления числовых характеристик и построения перечисленных ниже гистограмм, можно накопить в процессе генерирования числовой последовательности, не запоминая эту последовательность.

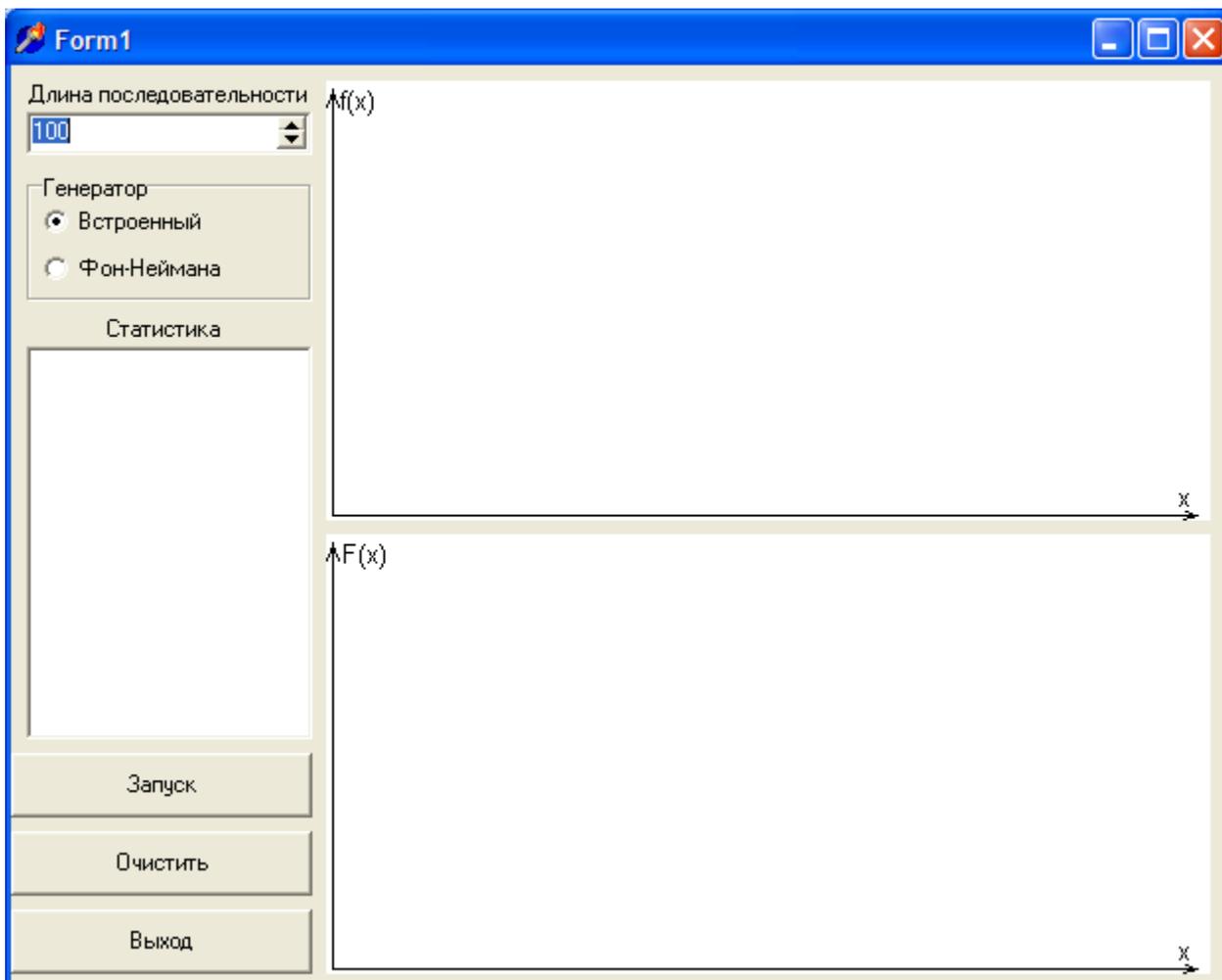


Рис. 1. Пример экранной формы

6. Для визуального исследования сгенерированной последовательности на равномерность построить:

- гистограмму для функции $f(X)$, задающей относительные (приведенные) частоты появления конкретных значений случайной величины. Эта функция представляет собой дискретный аналог функции плотности вероятностей (дифференциальной функции распределения);

- график для интегральной функции распределения $F(X)$.

7. Гистограмма функции $f(X)$ должна состоять из 100 «столбиков», которые на изображении могут касаться друг друга, либо быть отделены между собой некоторым интервалом. График функции $F(X)$ должен содержать 100 «ступенек», т.е. по одной ступеньке на каждое возможное значение случайной величины. Некоторые соседние ступеньки мо-

гут находиться на одной горизонтальной прямой. Вертикальные элементы ступенек должны быть прорисованы для соединения ступенек в одно целое, но при этом следует понимать, что эти вертикальные элементы к графику функции, строго говоря, не относятся.

8. Должна иметься возможность представления на экране в двух системах координат (СК) (одна СК для $f(X)$, другая – для $F(X)$) одновременно двух семейств гистограмм, соответствующих числовым последовательностям разной длины. Варианты (до 6 вариантов) длин числовых последовательностей задаются пользователем, например, в командной строке, в текстовом файле или в диалоговом режиме. Разные гистограммы должны рисоваться разным цветом с отображением соответствующим цветом информации о длине последовательности. Следующий вариант гистограммы и графика формируется после нажатия какой-либо клавиши или кнопки мыши, и изображения очередных гистограмм и графиков накладываются на предыдущие изображения.

9. Гистограммы и графики должны использовать все (или почти все) пространство формы, выделенное для них, но не усекаются границами формы. Оси координат должны иметь числовые отметки.

10. Отобразить на форме значение числа π , вычисленное статистическим методом. При этом использовать объем выборки, задаваемый в отдельном поле специально для вычисления значения числа π .

11. Гистограммы и графики можно строить стандартными средствами используемого языка программирования, либо собственными методами.

12. Спланировать вычислительные эксперименты для отладки программы и демонстрации преподавателю в процессе отчета по работе.

Приближенное вычисление числа π

В результате вычислительных экспериментов должны убедиться, что исследуемые генераторы псевдослучайных числовых последовательностей действительно порождают значения с законом рас-

пределения близким к равномерному. Точное вычисление нескольких старших разрядов числа π статистическим методом является косвенным подтверждением равномерности порождаемого генераторами закона распределения потому что идея метода вычисления числа π основана на равномерности этого закона распределения.

Содержание алгоритма вычисления числа π следующее.

Генерируется количество пар случайных значений интервала $[0, 1)$, равное заданному объему выборки. Каждая пара рассматривается как пара (x, y) координат точек плоскости. Понятно, что все такие точки будут находиться в границах квадрата со стороной длиной 1 «прижатого» к координатным осям и расположенного в первой четверти координатной плоскости. Когда использованный генератор числовых значений действительно порождает равномерный закон распределения чисел на интервале $[0, 1)$, моделируемые точки оказываются распределенными внутри указанного квадрата равномерно. То есть, на совокупности точек не будут наблюдаться ни их сгущения, ни их разреженности. Поэтому количество точек, заполнивших некоторую площадь внутри этого квадрата, будет зависеть только от величины этой площади и не будет зависеть от ее формы.

Далее рассуждаем следующим образом.

Представим себе круг радиусом $R=1$ с центром в начале системы координат. Будем рассматривать сектор этого круга, расположенный в первой четверти координатной плоскости. Этот сектор полностью находится внутри нашего единичного квадрата и имеет площадь

$S = \pi R^2/4$. С учетом сделанных ранее замечаний вероятность P попадания точки внутрь рассматриваемого сектора круга равна отношению количества точек, попавших в сектор, к общему количеству сгенерированных точек, и поэтому равна отношению площади S к площади рассматриваемого квадрата. Поскольку площадь квадрата равна 1, то получаем $P = S$ (следует понимать, что здесь S величина безразмерная, поскольку получена как отношение площадей):

$$P \approx \pi R^2/4 = \pi/4.$$

Из последней формулы получаем:

$$\pi \approx 4P.$$

Для вычисления числа π осталось вычислить значение P . Теорема Пифагора для этого пригодится.

Предостережение от принципиальных ошибок

Часто в программе предусматривается цикл, в который встроена формула $x_{i+1} = (Ax_i + B) \bmod C$, являющаяся основой метода Лемера. Это принципиально плохо! Необходимо разработать функцию, одно обращение к которой возвращает ровно одно число интервала $[0, 1)$. Математический смысл указанных скобок должен быть Вам понятен. На основе этой функции, назовем ее, например, *Lemer*, необходимо разработать функцию *LemerN(N)*, возвращающую целые числа интервала $[0, N-1]$, но не массив чисел. То есть, необходимо разработать две функции: *Lemer* и *LemerN*.

Функцию *Lemer* следует использовать для вычисления числа π .

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Объяснить понятия: непрерывная и дискретная случайные величины; кривая распределения; функция распределения; функция плотности вероятностей; математическое ожидание; дисперсия; среднеквадратичное отклонение.
2. Объяснить геометрический смысл понятий математического ожидания и дисперсии.
3. Написать формулы математического ожидания и дисперсии для равномерно распределенной случайной величины.
4. Как вычисляется вероятность попадания случайной величины на заданный отрезок по функции распределения и по функции плотности вероятностей?

5. Нарисовать идеальные графики для функции распределения и функции плотности вероятностей непрерывной и дискретной случайных величин с равномерным законом распределения.

6. Как зависят статистические значения математического ожидания, дисперсии, функции распределения и функции плотности вероятности от объема выборки?

7. Доказать разумность вычисленных программой статистических характеристик и построенных гистограмм и графиков.

8. Показать в программе и объяснить по тексту программы указанный шаг решения задачи.

9. Дать объяснения по указанному фрагменту программы.

10. Объяснить общий подход к построению генераторов псевдослучайных последовательностей.

11. Что такое «цикл последовательности» и чем объясняется его существование? Каковы требования предъявляются к длине цикла последовательности?

12. Объяснить метод Лемера. Показать по тексту программы его реализацию.

Преподаватель может задать любой иной вопрос, относящийся к теоретическим положениям рассматриваемой темы, к затрагиваемым общим вопросам теории моделирования, рассмотренным к данному моменту в лекционном материале, к любым сторонам вопросов программной реализации задачи.

ЛАБОРАТОРНАЯ РАБОТА №4

РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ПРОСТЕЙШЕЙ СМО

Трудоемкость задачи – 8 ч.

Цель работы – закрепление теоретических положений темы и получение навыков разработки и исследования имитационных моделей систем массового обслуживания.

Задание. Разработать *имитационную модель* простейшей разомкнутой СМО с одним прибором обслуживания и одним потоком заявок.

Каждая заявка с вероятностью p_1 получает низкий приоритет, в противном случае – высокий приоритет. Параметр p_1 должен задаваться интерфейсными средствами формы. Дисциплина обслуживания очереди – *без прерывания обслуживания и без отказов в обслуживании*. Последнее означает, что заявка покидает очередь только в результате ее выбора на обслуживание.

Собираемая статистика определена по вариантам в табл. 1.

Таблица 1

Варианты заданий по бригадам

Параметры СМО	Номера учебных бригад									
	1	2	3	4	5	6	7	8	9	10
1. Законы распределения:										
1.1. Интервалов прибытия:										
– равномерный	×		×		×		×		×	
– экспоненциальный		×		×		×		×		×
1.2. Времен обслуживания заявок:										
– равномерный	×	×			×	×			×	×
– экспоненциальный			×	×			×	×		
2. Дисциплины обслуживания очереди:										
2.1. FIFO – для заявок низкого приоритета и LIFO – для заявок высокого приоритета	×			×	×			×	×	
2.2. LIFO – для заявок низкого приоритета и FIFO – для заявок высокого приоритета		×	×			×	×			×
3. Статистика относительно очереди:										

Параметры СМО	Номера учебных бригад									
	1	2	3	4	5	6	7	8	9	10
– текущая длина очереди	×	×	×	×	×	×	×	×	×	×
– максимальная длина очереди			×	×	×				×	×
– средняя длина очереди	×	×	×	×	×	×	×	×	×	×
– среднее время ожидания в очереди	×	×	×	×	×	×	×	×	×	×
– максимальное время ожидания в очереди	×	×				×	×	×		
4. Статистика относительно прибора обслуживания:										
– минимальное время обслуживания заявки	×	×			×	×	×			
– среднее время обслуживания заявки	×	×	×	×	×	×	×	×	×	×
– максимальное время обслуживания заявки			×	×				×	×	×
– время простоя прибора		×	×	×				×	×	
– нагрузка прибора	×				×	×	×			×

Заданные в варианте законы распределения случайных величин должны быть представлены на форме интерфейсными средствами, позволяющими управлять всеми параметрами законов распределения. Так, например, для равномерного закона распределения необходимо предусмотреть поля для задания минимального и максимального значений случайной величины.

Каждая из заявок получает собственные значения параметров с учетом заданных законов распределения.

Методика выполнения

1. Изучить теоретический материал, изложенный в гл. 3 и 5 пособия.
2. Разработать концептуальную модель проекта.
3. Разработать интерфейсные средства программы (рис. 1).
4. Выполнить математическую проработку вопросов, связанных со сбором статистики.
5. Выполнить математическую проработку вопросов, связанных с генерацией числовых последовательностей с заданными законами распределения с учетом текущих параметров распределения.
6. Разработать структуру данных для описания отдельной заявки и контроля процесса ее продвижения по СМО.

7. Разработать структуру данных для сбора статистики.

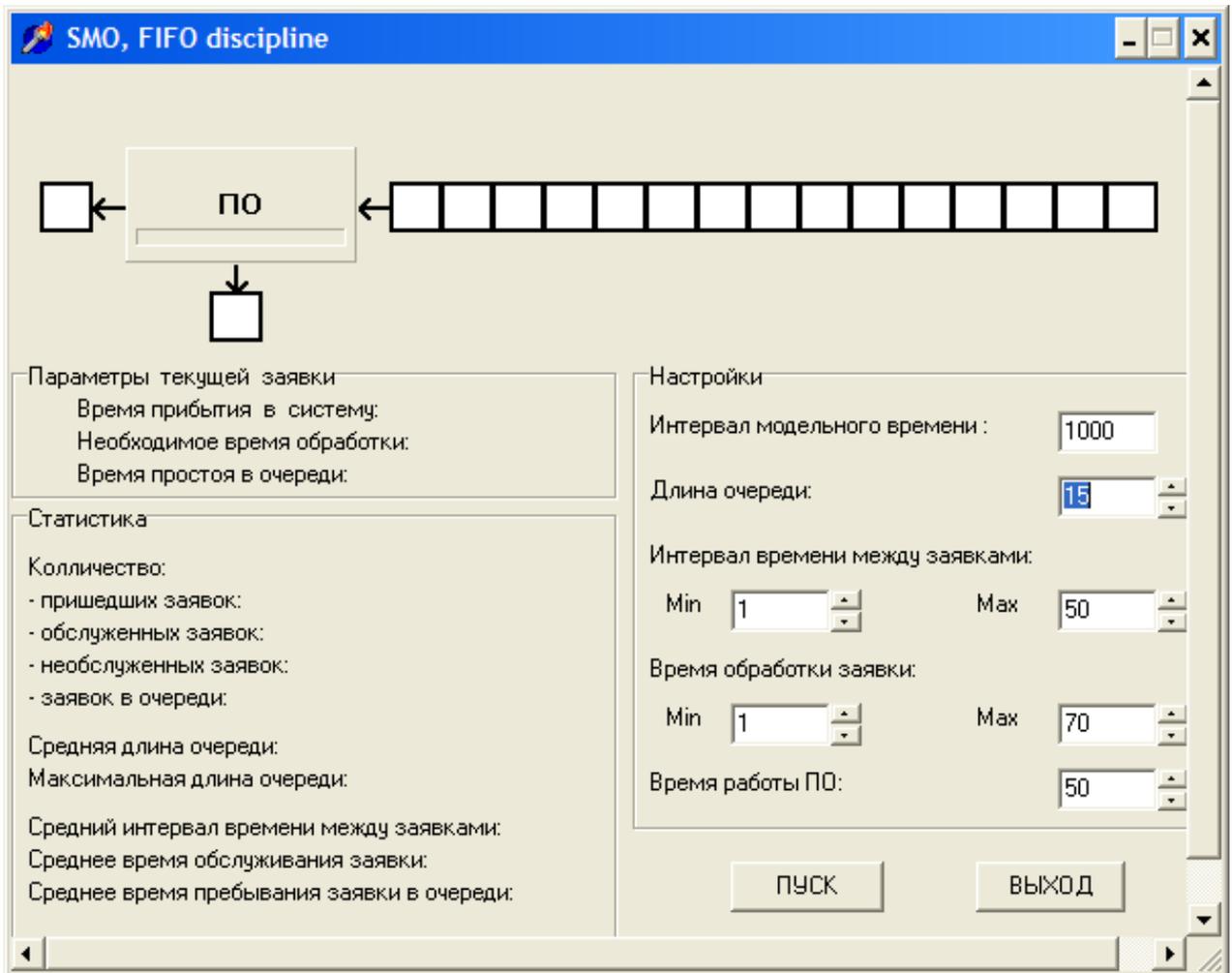


Рис. 1. Пример экранной формы

8. Продумать логические и числовые зависимости между статистическими данными, что необходимо для тщательного и добросовестного тестирования разрабатываемой программы.

9. Разработать, отладить и оттестировать программу.

10. Спланировать вычислительные эксперименты, которые будут демонстрироваться преподавателю в процессе отчета о работе.

11. Подготовиться к ответам на контрольные вопросы.

Замечания к содержанию рис. 1

1. Обычно очередь и прибор обслуживания изображают так, что заявки продвигаются не справа налево, как на рисунке, а наоборот – слева направо.

2. Экранная форма взята в том виде, как ее разработал студент. Грамматическая ошибка в слове «Колличество» в поле «Статистика» не должна быть предметом для подражания.

3. Название поля входных данных «Длина очереди» передает логический смысл этого поля не совсем точно. Имеется в виду максимально допустимая длина очереди. Названия элементов интерфейса должны быть максимально адекватными логическому содержанию этих элементов.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что понимается под СМО?
2. Дать формулировки понятиям: событие, поток событий.
3. Объяснить понятия: непрерывная и дискретная случайные величины; кривая распределения; функция распределения; функция плотности вероятностей; математическое ожидание; дисперсия; среднеквадратичное отклонение.
4. Объяснить технический и геометрический смысл понятий математического ожидания и дисперсии.
5. Написать формулы математического ожидания и дисперсии для равномерно распределенной случайной величины, для нормального распределения, для экспоненциального распределения.
6. Как можно получить генератор псевдослучайных величин с экспоненциальным законом распределения?
7. Как можно получить генератор псевдослучайных величин с нормальным законом распределения?
8. Как вычисляется средняя длина очереди?
9. Что такое дисциплина обслуживания очереди? Дать характеристики видов дисциплин обслуживания.
10. Привести примеры типов СМО.
11. Привести примеры моделей потоков событий.
12. Что описывает Пуассоновский закон распределения?
13. Что означает понятие «простейший поток»?

14. Доказать правильность сформированных программой результатов, показав корректность логических и арифметических взаимосвязей полученных статистических результатов моделирования.

Преподаватель может задать и другие вопросы.

ЛАБОРАТОРНАЯ РАБОТА №5

РАЗРАБОТКА СРЕДЫ ДЛЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ МИКРОПРОГРАММЫ ВЫПОЛНЕНИЯ АРИФМЕТИЧЕСКОЙ ОПЕРАЦИИ

Трудоемкость задачи – 8 ч.

Цель работы – закрепление теоретических положений раздела, получение навыков разработки и имитационного моделирования микропрограмм на примере микропрограммы выполнения арифметической операций.

Задание

1. Разработать алгоритм и микропрограмму выполнения арифметической операции над целыми числами со знаком, представленными на разрядной сетке длиной 1 байт. В случае микропрограммы деления вычислять целую часть частного, т.е. вычислять частное вида $A \text{ div } B$. Разработать среду для имитационного моделирования исполнения микропрограммы.

2. Тип арифметической операции и базовые признаки алгоритма ее выполнения студенты берут из варианта микропрограммы одного из членов учебной бригады, выданной в качестве исходных данных для курсовой работы по курсу «Моделирование».

3. В среде должны быть предусмотрены интерфейсные средства для задания кодов операндов, для отображения кодов всех вспомогательных переменных микропрограммы и кода результата, для отображения ГСА микропрограммы и метки на ней, указывающей текущее состояние процесса исполнения микропрограммы.

4. Состояния кодов должны отображаться на форме по мере их изменения.

5. Должны быть предусмотрены два режима выполнения микропрограммы:

– автоматический режим: микропрограмма выполняется полностью в результате одного щелчка кнопкой мыши на кнопке типа

«Старт», изображенной на форме. При этом промежуточные состояния кодов можно на форме не отображать;

– пошаговый режим: при щелчке кнопкой мыши на кнопке типа «Такт» формы процесс исполнения микропрограммы продвигается на 1 шаг, соответствующий одному такту работы операционного устройства. При этом все изменения кодов отображаются на форме, а на изображении микропрограммы в виде ГСА должна отображаться метка текущего состояния процесса исполнения микропрограммы: эта метка должна соответствовать состоянию управляющего автомата модели Мили.

Методика выполнения

1. Изучить теоретический материал, изложенный в гл. 6 пособия. Более подробное изложение соответствующих вопросов можно найти в [7, 8]. При изучении теоретического материала можно ограничиться кругом вопросов, связанных с выбранным вариантом арифметической операции.

2. В результате внимательного изучения выбранного варианта микропрограммы выявить основные признаки реализованного в микропрограмме алгоритма. Например, на рис. 1 в виде ГСА показана микропрограмма выполнения операции деления. Перечислим основные признаки этой микропрограммы, которые необходимо перенести в разрабатываемую микропрограмму, и отметим признаки, которые не следует переносить в новую микропрограмму.

2.1. Микропрограмма вычисляет частное $C := A / B$.

В рамках этой лабораторной работы необходимо реализовать вычисление целой части частного, то есть, $C := A \text{ div } B$.

2.2. В микропрограмме предусмотрен исход «Переполнение разрядной сетки». В рамках исходных данных этой лабораторной работы переполнение разрядной сетки исключено, поскольку при целых значениях операндов A и B частное не может быть больше, чем делимое A .

2.3. В микропрограмме предусмотрен ускоренный вариант получения частного в случае, когда делимое равно 0.

2.4. В микропрограмме операнды предполагаются представленными в прямых кодах.

2.5. В микропрограмме количество повторений цикла на 1 больше количества вычисляемых разрядов частного. В рамках исходных данных этой лабораторной работы вычисляется частное C в виде $C := A \text{ div } B$, поэтому в микропрограмме количество повторений ее циклической части должно определяться количеством значащих разрядов частного.

2.6. В микропрограмме по завершении циклической части выполняются микрооперации по округлению вычисленного частного. В разрабатываемой микропрограмме дробная часть должна отбрасываться, поэтому шаги округления частного не потребуются.

2.7. В микропрограмме реализован алгоритм деления с восстановлением остатка, но восстановление остатка выполняется не за счет микрооперации вида $A := A + B$ («Остаток» := «Остаток» + «Делитель»), а за счет запоминания текущего значения остатка во вспомогательной переменной D и использования на следующем цикле в микрооперации сдвига кода остатка либо только что вычисленного нового остатка, хранящегося в переменной C , либо предыдущего остатка, запомненного в переменной D .

2.8. В микропрограмме использован алгоритм деления со сдвигом кода остатка влево.

3. На основании результатов предыдущего пункта разработать новую микропрограмму выполнения соответствующей операции с учетом содержания задания на лабораторную работу.

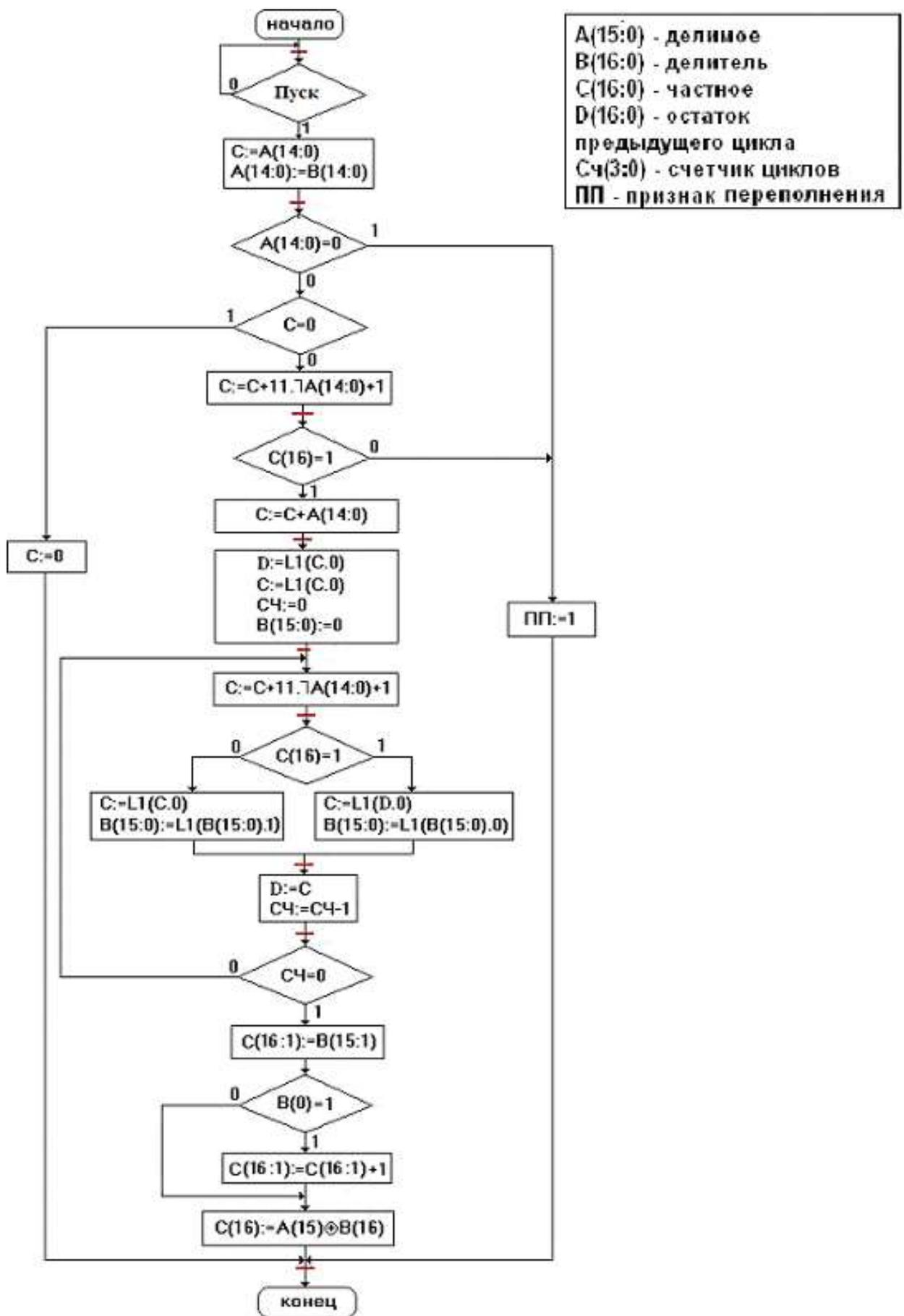


Рис. 1. Пример микропрограммы выполнения арифметической операции

4. Выполнить разметку состояний на ГСА, соответствующую состояниям управляющего автомата модели Мили. Этот этап выполнения работы требует привлечения знаний, полученных при изучении курса «Прикладная теория автоматов». Размеченные состояния автомата необходимы для организации режима пошагового выполнения микропрограммы и не предполагают построения каких-либо таблиц переходов, которые необходимы при проектировании управляющего автомата.

5. Разработать интерфейсные средства программы. Для отображения кодов рекомендуется использовать компоненты типа String Grid.

6. Разработать модели всех использованных микроопераций. При этом в качестве моделей обрабатываемых кодов использовать только беззнаковые целочисленные типы данных языка программирования.

7. Выполнить программирование и отладку всех необходимых процедур и функций. При разработке моделей микроопераций следует учитывать, что микрооперации выполняются соответствующими операционными элементами, поэтому подход к содержанию микроопераций с точки зрения программиста часто может привести к неверному ее толкованию. Так, например, в микропрограмме (см. рис. 1) имеется микрооперация « $S_c := S_c - 1$ », которая первый раз выполняется при значении $S_c = 0$. С позиций программиста после вычитания 1 из 0 получается число -1. Но операционный элемент типа «счетчик» из состояния 0 при вычитании 1 перейдет в состояние, соответствующее самому большому числу, которое может отобразить этот счетчик. В приведенной микропрограмме счетчик имеет 4 разряда, поэтому из состояния 0 при вычитании 1 он перейдет в состояние 15. В качестве модели этого счетчика следует взять переменную типа `byte`, поэтому указанный переход счетчика должен привести к установке этой переменной в состояние, задаваемое следующим двоичным кодом: 00001111.

8. Проверить работу разработанной среды при всех значимых вариантах исходных данных.

9. Модель среды не должна давать сбоев при любых манипуляциях пользователя.

10. Спланировать вычислительные эксперименты, которые будут демонстрироваться преподавателю в процессе отчета о работе.

11. Подготовиться к ответам на контрольные вопросы.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Назвать основные признаки метода выполнения операции, реализованные в разработанном алгоритме.

2. Подтвердить вычислительным экспериментом, что в микропрограмме реализованы именно заданные особенности метода выполнения операции.

3. В каких кодах представляются исходные операнды операции?

4. Какие приемы ускорения получения результата реализованы в микропрограмме?

5. Показать в исходном тексте программы модель указанной преподавателем микрооперации.

6. Написать модель заданной преподавателем микрооперации.

7. Объяснить, как реализован пошаговый режим выполнения микропрограммы: почему на одно и то же действие пользователя «нажатие кнопки Такт», модель реагирует по-разному, выполняя разные фрагменты микропрограммы?

8. В микропрограмме на рис. 1 до циклической части переменной $Sч$ присваивается значение 0. В циклической части выполняется микрооперация $Sч := Sч - 1$. Условие окончания цикла имеет вид $Sч = 0$. Возможно ли в такой микропрограмме достижение логического конца цикла? Ответ аргументировать.

9. Какому типу автомата (модели Мура или модели Мили) в реализованной модели соответствует такт выполнения микропрограммы?

Преподаватель может задать любой иной вопрос, относящийся к теоретическим положениям рассматриваемой темы, к затрагиваемым общим вопросам теории моделирования, рассмотренным к данному моменту в лекционном материале, к любым сторонам вопросов программной реализации задачи.

ВОПРОСЫ К ЭКЗАМЕНУ

Примерная структура билета

Министерство образования и науки РФ
Федеральное государственное бюджетное
образовательное учреждение высшего образования

САМАРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра Вычислительная техника

БИЛЕТ № 0

по дисциплине «Моделирование»

Направление подготовки 09.03.04 «Программная инженерия»

Институт АИТ Семестр 6

1. Свойства моделей.
2. Потоки событий и их модели.

Составил _____ проф. Мартемьянов Б.В.
подпись Ф.И.О.

Утверждаю:

Заведующий кафедрой _____ проф. Иващенко А.В.
подпись Ф.И.О.

« ____ » _____ 20 ____ г.
дата

« ____ » _____ 20 ____ г.
дата

Раздел 1. Основные понятия теории моделирования

1. «Объект моделирования», «модель», «моделирование». Цель моделирования.
2. Отличительные особенности моделирования как метода исследования. Основные направления применения моделирования.
3. Аналогия. Аналогия прямая и косвенная. Адекватность модели.
4. Свойства моделей.
5. Модели материальные и идеальные, концептуальные и математические.
6. Модели линейные, нелинейные, статические, динамические, вероятностные.
7. Модели непрерывные и дискретные, детерминированные и стохастические
8. Модели функциональные и структурные.

Раздел 2. Моделирование вычислительных процессов

9. Три формы записи алгебраических выражений: инфиксная, постфиксная и префиксная. Недостатки и достоинства форм выражений.

10. Неформальное преобразование инфиксного выражения в постфиксное и в префиксное.

11. Алгоритм Дейкстры: концептуальная модель процесса преобразования инфиксного выражения в постфиксное.

12. Алгоритм Дейкстры: таблица решений.

13. Стековая память и особенности ее моделирования.

14. Концептуальная модель вычисления значения выражения по его постфиксной форме.

15. Формализация вычисления значения выражения по его постфиксной форме.

16. Пример пошагового выполнения алгоритма вычисления значения выражения, представленного в постфиксной форме записи.

Раздел 3. Статистическое моделирование

17. Модели аналитические и имитационные: понятия, преимущества и недостатки.

18. Статистическая модель случайных процессов на примере СМО.

19. Статистическое имитационное моделирование и метод Монте-Карло.

20. Пример статистического моделирования: приближенное вычисление числа Π .

21. Методика статистического моделирования.

22. Планирование имитационных экспериментов с моделями, анализ и интерпретация результатов моделирования на ЭВМ

23. Случайные события, случайные величины (СВ). СВ непрерывные и дискретные. Основные числовые характеристики случайных величин.

24. Понятия: закон распределения СВ, дифференциальная и интегральная функции распределения (ДФР и ИФР) СВ и их взаимосвязь.

25. Свойства ДФР и ИФР.

26. Вычисление вероятности попадания СВ на заданный интервал по ДФР и по ИФР.

27. Числовые характеристики распределения непрерывных и дискретных СВ (с формулами).

28. Свойства математического ожидания.

29. Свойства дисперсии и среднеквадратичного отклонения.

30. Математические модели специальных распределений СВ: равномерное распределение.

31. Математические модели специальных распределений СВ: нормальное распределение.

32. Математические модели специальных распределений СВ: экспоненциальное распределение.

33. Математические модели специальных распределений СВ: распределение Пуассона.

34. Правило трех сигм нормального распределения.

35. Псевдослучайные числовые последовательности (ПЧП). Общий метод моделирования ПЧП и свойства ПЧП.

36. Моделирование равномерно распределенной СВ методом Лемера.

37. Построение генераторов с заданным законом распределения числовых последовательностей: универсальный подход.

38. Получение стандартных законов распределения СВ на основе равномерно распределенной СВ: генератор ПЧП с нормальным законом распределения.

39. Получение стандартных законов распределения СВ на основе равномерно распределенной СВ: генератор ПЧП с экспоненциальным законом распределения.

40. Задачи и методы теории СМО. Простейшая СМО (разомкнутая, один источник заявок, одна очередь, один прибор обслуживания), ее параметры, характеристики и сопутствующие понятия.

41. Классификация СМО (однородные, неоднородные, одноканальные, многоканальные, однофазные, многофазные, замкнутые, разомкнутые, сети СМО).

42. Потоки событий и их модели (однородные и неоднородные; регулярные; ординарные; рекуррентные; без последствия; с ограниченным последствием; стационарные и нестационарные).

43. Простейший поток событий и его свойства.

44. Понятие дисциплина обслуживания заявок. Классификация дисциплин обслуживания: FIFO, LIFO, без приоритетов, с приоритетами абсолютными и относительными, с прерыванием обслуживания.

45. Стационарный пуассоновский (простейший) поток событий (свойства; вероятностные характеристики: функция распределения, интенсивность, мат. ожидание и дисперсия интервала; тенденция событий к группированию).

46. Понятия аналитическая модель СМО, имитационная модель СМО. Сравнительный анализ двух классов моделей по сложности реализации, универсальности и разрешимости, трудоемкости.

47. Применение моделей СМО для задач анализа и синтеза систем.

48. События основные и вспомогательные.

49. Моделирование основных событий на примере простейшей СМО.

50. Моделирование таймера модельного времени.

Раздел 4. Логическое моделирование элементов цифровой техники

51. Модели элементов вычислительных систем на разных уровнях детализации:

- уровень принципиальных электрических (электронных) схем.

- уровень функциональных (логических) схем.
- уровень межрегистровых передач.

52. Модели функциональные и структурные. Интерпритативные и компилятивные алгоритмы моделирования.

53. Методы моделирования цифровых устройств: синхронное моделирование; событийное моделирование; асинхронное моделирование. Адекватности методов моделирования.

54. Понятия о статическом и динамическом рисках сбоя и состязаниях сигналов.

55. Алфавиты моделирования.

56. Модели элементов КС.

57. Функциональная модель КС.

58. Ранги и ранжирование элементов КС.

КОНТРОЛИРУЮЩИЕ ТЕСТЫ

1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ МОДЕЛИРОВАНИЯ

1. Задание {{ 1 }}

Выбрать наиболее правильную и полную характеристику понятия «моделирование».

- Моделирование – это диалектико-материалистический метод познания и научного исследования.
- Моделирование – это метод исследования, основанный на постановке прямого эксперимента с исследуемым объектом.
- Моделирование – это метод косвенного исследования проектируемых или реально существующих объектов.
- Моделирование – это метод познания, основанный на выдвижении и проверке гипотез.
- Моделирование – это метод косвенного исследования проектируемых или реально существующих объектов, реализуемый как процесс получения информации об изучаемых свойствах объекта-оригинала с помощью изучения свойств объекта-модели.

2. Задание {{ 2 }}

Вместо многоточий вставить наиболее подходящий термин.

- прямая ...; косвенная ...

- 1 - модель
- 2 - характеристика
- 3 - аналогия
- 4 - теория

3. Задание {{ 3 }}

Вместо многоточия вставить наиболее подходящий термин.

«Если результаты моделирования совпадают с поведением реальных объектов и могут быть использованы для прогнозирования протекающих в них процессов с погрешностью не выше, заданной, то считается, что модель ... объекту.»

4. Задание {{ 4 }}

Вместо многоточий вставить наиболее подходящий термин:

«Моделирование базируется на математической теории ..., согласно которой абсолютное ... может иметь место лишь при замене одного объекта другим точно таким же».

- 1 - аналогий
- 2 - сравнения
- 3 - подобия

- 4 - выбора
- 5 - адекватности

5. Задание {{ 5 }}

Вместо многоточий вставить наиболее подходящий термин:

«Между механической системой и электронной схемой, поведение которых описывается дифференциальным уравнением одного и того же вида (например, уравнением второго порядка) существует ...»

- 1 – несущественная аналогия
- 2 – отношение адекватности
- 3 – косвенная аналогия
- 4 – существенная аналогия
- 5 – прямая аналогия

6. Задание {{ 6 }}

Указать термин, не характеризующий свойства модели

- 1 - упрощенность
- 2 - приближительность
- 3 - информативность
- 4 - адекватность
- 5 - сходимость

7. Задание {{ 7 }}

Указать термин, не характеризующий свойства модели

- 1 - стационарность
- 2 - управляемость
- 3 - полнота
- 4 - адаптивность
- 5 - эволюционируемость

8. Задание {{ 8 }}

Вместо многоточия вставить наиболее подходящий термин

«С позиций теории познания модели делятся на два больших класса:
- модели ...;
- модели *идеальные*»

9. Задание {{ 9 }}

Вместо многоточия вставить наиболее подходящий термин

«С позиций теории познания модели делятся на два больших класса:
- модели *материальные*;
- модели ...»

13. Задание {{ 13 }}

Вместо многоточия вставить наиболее подходящий термин

«По степени детализации описания оригинала абстрактные модели можно разделить на два класса;

- модели ...;
- модели *математические*.»

14. Задание {{ 14 }}

Вместо многоточия вставить наиболее подходящий термин

«По степени детализации описания оригинала абстрактные модели можно разделить на два класса;

- модели *концептуальные*;
- модели»

15. Задание {{ 15 }}

Вместо многоточия вставить наиболее подходящий термин

«Модели Мура и Мили абстрактных конечных автоматов относятся к классу ...»

- 1 - непрерывно – детерминированных моделей
- 2 - непрерывно – стохастических моделей
- 3 - дискретно – стохастических моделей
- 4 - дискретно – детерминированных моделей

16. Задание {{ 16 }}

Вместо многоточия вставить наиболее подходящий термин

«Дифференциальные уравнения в качестве математических моделей относятся к классу ...»

- 1 - непрерывно – детерминированных моделей
- 2 - непрерывно – стохастических моделей
- 3 - дискретно – стохастических моделей
- 4 - дискретно – детерминированных моделей

17. Задание {{ 17 }}

Вместо многоточия вставить наиболее подходящий термин

«Системы массового обслуживания в качестве математических моделей относятся к классу ...»

- 1 - непрерывно – детерминированных моделей
- 2 - непрерывно – стохастических моделей
- 3 - дискретно – стохастических моделей
- 4 - дискретно – детерминированных моделей

18. Задание {{ 18 }}

Вместо многоточия вставить подходящий термин

«Модели, если отвлечься от областей, сфер их применения, бывают трех типов:

- ...;
- прагматические;
- инструментальные».

19. Задание {{ 19 }}

Вместо многоточия вставить подходящий термин

«Модели, если отвлечься от областей, сфер их применения, бывают трех типов:

- познавательные;
- ...;
- инструментальные».

20. Задание {{ 20 }}

Вместо многоточия вставить подходящий термин

«В зависимости от степени абстрагирования при описании физических свойств технической системы различают три основных иерархических уровня:

- верхний или ...;
- средний или макроуровень;
- нижний или микроуровень».

21. Задание {{ 21 }}

Вместо многоточия вставить подходящий термин

«По характеру отображаемых свойств технического объекта математические модели делятся на ... и структурные».

23. Задание {{ 23 }}

Вместо многоточия вставить подходящий термин

«По характеру отображаемых свойств технического объекта математические модели делятся на функциональные и ...».

24. Задание {{ 24 }}

Вместо многоточия вставить подходящий термин

«Если при моделировании учитываются инерционные свойства объекта и (или) изменение во времени объекта или внешней среды, то модель называют ...».

25. Задание {{ 25 }}

Вместо многоточия вставить подходящий термин

«Если при моделировании не учитываются инерционные свойства объекта и изменение во времени объекта или внешней среды, то модель называют ...».

26. Задание {{ 26 }}

Среди следующих пар терминов выбрать пару наиболее не подходящую ко всем остальным парам.

- 1 – статическая – динамическая
- 2 – детерминированная - вероятностная
- 3 – стационарный - ординарный
- 4 – непрерывная - дискретная
- 5 – идеальная – материальная

2. МОДЕЛИРОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ

27. Задание {{ 27 }}

Классифицировать одним словом форму записи выражения: $ab+a/cd^*$ -

28. Задание {{ 28 }}

Классифицировать одним словом форму записи выражения: $/+aba^*cd$

29. Задание {{ 29 }}

Классифицировать форму записи выражения: $ab+a/cd^*$ -

- 1 – бесскобочная
- 2 – функциональная;
- 3 – инфиксная;
- 4 – постфиксная;
- 5 – префиксная

30. Задание {{ 30 }}

Классифицировать форму записи выражения: $/+aba^*cd$

- 1 – функциональная
- 2 – инфиксная;
- 3 – постфиксная;
- 4 – префиксная

31. Задание {{ 31 }}

Выражение $\sin((a-b)/b)^*(b-a)$ эквивалентно выражению

- 1) $ab/b \sin ba^*$
- 2) $*\sin/-abb-ba$
- 3) $*/\sin -abb-ba$
- 4) $*/-\sin abb-ba$

32. Задание {{ 32 }}

Выражение $*\sin/-abb-ba$ эквивалентно выражению

- 1) $ab/b \sin ba^*$
- 2) $\sin((a-b)/b)*(b-a)$
- 3) $*/\sin -abb-ba$
- 4) $*/-\sin abb-ba$

33. Задание {{ 33 }}

Выражение $(\sin(a-b))/b*(b-a)$ эквивалентно выражению

- 1) $ab/b \sin ba^*$
- 2) $*\sin/-abb-ba$
- 3) $*/\sin-abb-ba$
- 4) $*/-\sin abb-ba$

34. Задание {{ 34 }}

Выражение $*/\sin-abb-ba$ эквивалентно выражению

- 1) $ab/b \sin ba^*$
- 2) $(\sin(a-b))/b*(b-a)$
- 3) $*\sin/-abb-ba$
- 4) $*/-\sin abb-ba$

35. Задание {{ 35 }}

Выражение $\sin((a-b)/b)*(b-a)$ эквивалентно выражению

- 1) $ab/b \sin ba^*$
- 2) $*\sin/-abb-ba$
- 3) $*/\sin -abb-ba$
- 4) $*/-\sin abb-ba$

36. Задание {{ 36 }}

Выражение $ab/b \sin ba^*$ эквивалентно выражению

- 1) $*/-\sin abb-ba$
- 2) $*\sin/-abb-ba$
- 3) $*/\sin -abb-ba$
- 4) $\sin((a-b)/b)*(b-a)$

37. Задание {{ 37 }}

Вместо многоточия вставить подходящее слово:

«Общим видимым признаком постфиксной и префиксной форм записи любых алгебраических выражений является отсутствие ...»

38. Задание {{ 38 }}

Вместо многоточия вставить подходящее слово:

«Бесскобочные формы записи алгебраических выражений не предполагают использование понятия ... операций».

39. Задание {{ 39 }}

Вместо многоточия вставить подходящее слово:

«Форму записи алгебраических выражений со знаком операции после операндов называют ...».

40. Задание {{ 40 }}

Вместо многоточия вставить подходящее слово:

«Форму записи алгебраических выражений со знаком операции перед операндами называют ...».

41. Задание {{ 41 }}

Выбрать правильный вариант окончания предложения:

«Большой вклад в исследование бесскобочных форм записи алгебраических выражений внес польский математик ...»

- 1 - Петрарка
- 2 – Бонч Брюевич
- 3 - Лукасевич
- 4 - Колосовский
- 5 - Пшеньковский

42. Задание {{ 42 }}

Вместо многоточия вставить подходящее слово:

«В основе алгоритма Дейкстра преобразования инфиксной формы записи выражения в постфиксную форму известный в программировании метод, основанный на использовании ... принятия решений»

43. Задание {{ 43 }}

Выбрать подходящий термин вместо многоточия:

«Доступ к содержимому стека организуется по принципу ...»

- 1 – Random access
- 2 – FIFO
- 3 – LIFO
- 4 – LILO
- 5 – Stack Pointer

44. Задание {{ 44 }}

Выбрать подходящий термин вместо многоточия:

«В самом «низу» стека находится ... стека»

45. Задание {{ 45 }}

В следующем арифметическом выражении каждая цифра – это отдельное число: **83-61-/83-***

В процессе вычисления значения выражения в строке остались символы **3-***. Какое число находится в этот момент в вершине стека?

46. Задание {{ 46 }}

В следующем арифметическом выражении каждая цифра – это отдельное число: **83-61-/83-***

В процессе вычисления значения выражения в строке остались символы **3-***. Какое число находится в этот момент над дном стека?

47. Задание {{ 47 }}

В следующем арифметическом выражении каждая цифра – это отдельное число: **83-61-/83-***

Чему равно значение выражения?

48. Задание {{ 48 }}

. В следующем арифметическом выражении каждая цифра – это отдельное число:

/+ /+5435-95

Чему равно значение выражения?

3. СТАТИСТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

49. Задание {{ 49 }}

Вместо многоточия вставить подходящее слово:

«Проведение на ЭВМ вычислительных экспериментов с математическими моделями, имитирующими поведение реальных объектов, процессов или систем называется ... моделированием».

50. Задание {{ 50 }}

Вместо многоточия вставить подходящее слово:

«Реальные процессы и системы можно исследовать с помощью двух типов математических моделей: ... и имитационных».

51. Задание {{ 51 }}

Вместо многоточия вставить подходящее слово:

«Реальные процессы и системы можно исследовать с помощью двух типов математических моделей: ... и аналитических».

52. Задание {{ 52 }}

Вместо многоточия вставить подходящее слово:

«В ... моделях поведение реальных процессов и систем (РПС) задается в виде явных функциональных зависимостей (уравнений линейных или нелинейных, дифференциальных или интегральных, систем этих уравнений)».

53. Задание {{ 53 }}

Вместо многоточия вставить подходящее слово:

«...моделирование представляет собой численный метод проведения на ЭВМ вычислительных экспериментов с математическими моделями, имитирующими поведение реальных объектов, процессов и систем во времени в течении заданного периода».

54. Задание {{ 54 }}

Вместо многоточия вставить подходящее слово:

«Численный метод, решающий задачу генерирования последовательности псевдослучайных чисел с заданными законами распределения, получил название "метод статистических испытаний" или "метод ... - Карло"».

55. Задание {{ 55 }}

Вместо многоточия вставить подходящее слово:

«Метод ... - это численный метод, моделирующий на ЭВМ псевдослучайные числовые последовательности с заданными вероятностными характеристиками»

56. Задание {{ 56 }}

Вместо многоточия вставить подходящее слово:

«... случайной величиной называется случайная величина, возможные значения которой непрерывно заполняют какой-то промежуток».

57. Задание {{ 57 }}

Вместо многоточия вставить подходящее слово:

«... (прерывной) случайной величиной называется случайная величина, принимающая отделенные друг от друга значения, которые можно пронумеровать».

58. Задание {{ 58 }}

Вместо многоточия вставить подходящее слово:

«Законом ... случайной величины называется всякое соотношение, устанавливающее связь между возможными значениями случайной величины и соответствующими им вероятностями, с которыми СВ принимает именно эти значения».

59. Задание {{ 59 }}

Вместо многоточия вставить подходящее слово:

«Множество вещественных чисел интервала $[0, 1]$ *несчетно* и имеет мощность, названную ...».

60. Задание {{ 60 }}

Вместо многоточия дать название функции $F(x)$, если $F(x) = P(X < x)$, где P обозначает вероятность события.

« $F(x)$ – это функция ...»

61. Задание {{ 61 }}

Вместо многоточия вставить термин, характеризующий свойство функции распределения любой случайной величины: как непрерывной, так и дискретной.

«Функция распределения $F(x)$ любой случайной величины X обладает следующими свойствами:

1. Область значений $F(x)$ совпадает с закрытым интервалом $[0, 1]$.
2. $F(x)$ – это ... функция».

62. Задание {{ 62 }}

Вместо многоточия вставить термин, характеризующий свойство функции распределения любой дискретной случайной величины.

«Для дискретной случайной величины X функция $F(x)$ есть разрывная ... функция, непрерывная слева от каждого их возможных значений X ».

63. Задание {{ 63 }}

Вместо многоточия вставить одно слово:

«Если $F(x)$ - функция распределения непрерывной случайной величины, тогда функция $f(x) = dF(x)/dx$ называется ... распределения этой случайной величины».

64. Задание {{ 64 }}

Если $f(x)$ – плотность распределения случайной величины, тогда после знака равенства следует поставить ...

$$\int_{-\infty}^{+\infty} f(x) dx =$$

65. Задание {{ 65 }}

Вместо многоточий вставить подходящие слова:

«Если x_i – возможное значение дискретной случайной величины, а p_i - вероятность значения x_i , тогда формула $\sum(x_i p_i)$, в которой суммирование ведется по всем возможным значениям индекса i , задает этой случайной величины».

66. Задание {{ 66 }}

Вместо многоточия вставить подходящее слово:

«Математическое ожидание квадрата центрированной случайной величины задает ... этой случайной величины».

67. Задание {{ 67 }}

Вместо многоточия вставить подходящее слово:

«Если x_i – возможное значение дискретной случайной величины; p_i - вероятность значения x_i ; m_x – математическое ожидание этой случайной величины,

тогда формула $\sum((x_i - m_x)^2 p_i)$, в которой суммирование ведется по всем возможным значениям индекса i , задает ... этой случайной величины».

68. Задание {{ 68 }}

Вместо многоточия вставить подходящее слово:

«Распределение случайной величины, задаваемое формулой $F(\tau) = 1 - e^{-\tau\lambda}$, называется ... распределением».

69. Задание {{ 69 }}

Вместо многоточия вставить подходящее слово:

«Плотностью распределения, задаваемой формулой $f(\tau) = \lambda e^{-\tau\lambda}$, обладает ... распределение».

70. Задание {{ 70 }}

Вместо многоточия вставить одно наиболее подходящее слово:

«В экспоненциальном распределении с плотностью $f(\tau) = \lambda e^{-\tau\lambda}$ параметр λ задает ... потока».

71. Задание {{ 71 }}

Вместо многоточия вставить слово:

«Распределение, задающее вероятность $P_n(\tau)$ того, что за промежуток времени τ произойдет ровно n событий, называется распределением ...»

72. Задание {{ 72 }}

Вместо многоточий вставить подходящие слова:

«Для ... математическое ожидание m_n количества событий за время τ равно $m_n(\tau)$: $m_n(\tau) = \lambda\tau$ ».

73. Задание {{ 73 }}

Вместо многоточия вставить слово:

«Пара параметров (m, σ) задает однозначно ... распределение».

74. Задание {{ 74 }}

Вместо многоточия вставить фамилию математика:

«Нормальное распределение связано с именем математика ...».

75. Задание {{ 75 }}

Задать недостающий термин:

«К основным числовыми характеристиками случайной величины относятся:

- математическое ожидание;
- ...;
- среднее квадратичное отклонение».

76. Задание {{ 76 }}

Вместо многоточия вставить термин:

«Корень квадратный из среднеквадратичного отклонения задает ...».

77. Задание {{ 77 }}

Функция распределения дискретной случайной величины (СВ) задана частично следующей таблицей:

Значения x СВ: 10 11 12 13 14 15 16

Вероятности 0,2 0,25 0,3 0,33 0,4 0,52 0,58

$P(X < x)$

Вычислить вероятность попадания СВ на закрытый интервал [12; 15].

78. Задание {{ 78 }}

Функция распределения дискретной случайной величины (СВ) задана частично следующей таблицей:

Значения x СВ: 10 11 12 13 14 15 16 17

Вероятности 0,2 0,25 0,3 0,33 0,4 0,52 0,58 0,65

$P(X < x)$

Вычислить вероятность попадания СВ на закрытый интервал [11; 16].

79. Задание {{ 79 }}

Функция распределения дискретной случайной величины (СВ) задана частично следующей таблицей:

Значения x СВ: 10 11 12 13 14 15 16 17

Вероятности 0,2 0,25 0,3 0,33 0,4 0,52 0,58 0,65

$$P(X < x)$$

Вычислить вероятность попадания СВ на закрытый интервал [13; 16]: ...

80. Задание {{ 80 }}

В следующем определении найти неточно использованный термин (одно слово) и написать его в качестве ответа точно в том виде, как он дан в определении.

«Функция Random языка программирования используется для генерации случайных чисел, равномерно распределенных на заданном интервале».

81. Задание {{ 81 }}

В следующем определении найти неточно использованный термин (одно слово) и в качестве ответа написать точную формулировку этого термина.

«Функция Random языка Паскаль используется для генерации случайных чисел, равномерно распределенных на заданном интервале».

82. Задание {{ 82 }}

В одном из приведенных вариантов оператор языка Паскаль позволяет получать очередное значение X псевдослучайной величины, равномерно распределенной на отрезке от 100 до 250, включая границы. Задать номер этого варианта.

- 1. $X := \text{Random}(150); \quad X := X + 100;$
- 2. $X := \text{Random}(250); \quad X := X + 100;$
- 3. $X := \text{Random}(151); \quad X := X + 100;$
- 4. $X := \text{Random}(151) + 100;$
- 5. $X := \text{Random}(350) - 100;$
- 6. $X := \text{Random}(150) + 100;$

83. Задание {{ 83 }}

Указать номер лишнего (неверного) элемента в последующем перечислении.

Известны, например, следующие методы генерации псевдослучайных последовательностей чисел:

- 1 – метод фон Неймана;
- 2 – линейный конгруэнтный метод;
- 3 – метод срединных квадратов;
- 4 – метод Рунге-Кутта;
- 5 – мультипликативный метод.

84. Задание {{ 84 }}

Частным случаем метода Лемера генерации псевдослучайных числовых последовательностей является

- 1 – метод фон Неймана;
- 2 – линейный конгруэнтный метод;
- 3 - мультипликативный метод;
- 4 – метод срединных квадратов;
- 5 - метод Рунге-Кутты

85. Задание {{ 85 }}

Вместо многоточия вставить термин:

«Метод генерации псевдослучайных числовых последовательностей, основанный на соотношении

$$x_{i+1} = (Ax_i + B) \bmod C,$$

называется методом ...»

86. Задание {{ 86 }}

Вместо многоточий вставить слова:

«Название метода Лемера, раскрывающее его математическую сущность: метод»

87. Задание {{ 87 }}

Назвать одним словом (в именительном падеже) тип распределения, к которому относятся числовые последовательности, генерируемые методом Лемера.

Тема: Основы теории моделей систем массового обслуживания

88. Задание {{ 88 }}

Среди следующих пар терминов выбрать пару наиболее не подходящую ко всем остальным.

- 1 - однородный – неоднородный;
- 2 - регулярный – стационарный;
- 3 - аналитическая – имитационная;
- 4 - ординарный – простейший;
- 5 - без последствия – с ограниченным последствием

89. Задание {{ 89 }}

Выбрать термин, в определении которого не принимается во внимание фактор времени.

- 1. регулярный
- 2. рекуррентный
- 3. ординарный
- 4. однородный
- 5. стационарный

90. Задание {{ 90 }}

Выбрать поток (потоки?), интенсивность которого (которых?) не изменяется со временем.

- 1. стационарного
- 2. рекуррентного
- 3. ординарного
- 4. однородного
- 5. простейшего

91. Задание {{ 91 }}

Вместо многоточия вставить наиболее подходящий термин

«... поток событий - это поток, в котором все функции распределения интервалов между событиями совпадают».

92. Задание {{ 92 }}

Вместо многоточия вставить наиболее подходящий термин

«...поток событий представляет собой детерминированную последовательность событий».

93. Задание {{ 93 }}

Указать лишнее свойство в следующем определении.

«Стационарным Пуассоновским (простейшим) называется поток, обладающий тремя свойствами:

- 1 – стационарностью;
- 2 – ординарностью;
- 3 - регулярностью;
- 4 – отсутствием последствия.

94. Задание {{ 94 }}

Добавить недостающий термин в следующем определении.

«Стационарным Пуассоновским (простейшим) называется поток, обладающий тремя свойствами: стационарностью, ..., отсутствием последствия».

95. Задание {{ 95 }}

Добавить недостающий термин в следующем определении.

«Стационарным Пуассоновским (простейшим) называется поток, обладающий тремя свойствами: ..., ординарностью, отсутствием последствия».

96. Задание {{ 96 }}

Вместо многоточия вставить наиболее подходящий термин:

«... поток характеризуется тем, что в любой данный момент времени вероятность прихода более чем одной заявки настолько меньше, чем вероятность прихода только одной заявки, что можно считать: в любой данный момент времени может прийти не более одной заявки».

97. Задание {{ 97 }}

Сколько разных дисциплин обслуживания перечислено далее:

FIFO, LIFO, LILO, FILO ?

Ответ давать в виде цифры.

98. Задание {{ 98 }}

К какому элементу простейшей СМО относится термин “дисциплина обслуживания?”.

- 1 – к входному потоку заявок;
- 2 – к выходному потоку заявок;
- 3 – к источнику заявок;
- 4 – к очереди;
- 5 – к прибору обслуживания.

99. Задание {{ 99 }}

Привести недостающий термин из классификации СМО, характеризующий простейшую СМО, рассмотренную в рамках лабораторных работ по моделированию.

«Простейшая СМО – это одиночная одноканальная однофазная ... СМО».

4. ЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ ЭЛЕМЕНТОВ ЦИФРОВОЙ ТЕХНИКИ

106. Задание {{ 133 }}

Среди следующих терминов выбрать термины, обозначающие виды моделей

- 1- функциональные;
- 2- интерпретативные;
- 3- компилятивные;
- 4- структурные.

107. Задание {{ 134 }}

Среди следующих терминов выбрать термины, обозначающие виды алгоритмов моделирования:

- 1- функциональные;
- 2- интерпретативные;
- 3- компилятивные;
- 4- структурные.

108. Задание {{ 135 }}

Выбрать термин, не характеризующий адекватность модели моделируемой схеме:

- 1 – синхронное;
- 2 – асинхронное;
- 3 – событийное;
- 4 – ординарное;
- 5 – с интервальными задержками

109. Задание {{ 136 }}

Можно ли комбинационную схему рассматривать как цифровой автомат?

Если можно, то указать тип этого автомата:

- 1 – нельзя;
- 2 – автомат Мура с одним состоянием;
- 3 – автомат Мили с одним состоянием;
- 4 – автомат Мили с ноль состояниями
- 5 – автомат Мура с ноль состояниями

110. Задание {{ 137 }}

Выбрать вид модели, отличающейся наибольшей степенью адекватности моделируемой схеме:

- 1 – синхронная;
- 2 – интерпретативная;
- 3 – событийная;
- 4 – асинхронная;
- 5 – компилятивная.

111. Задание {{ 138 }}

Выбрать вид модели, которая в готовом виде (т.е. без перепрограммирования) отличается возможностью применения к более широкому классу схем:

- 1 – синхронная;
- 2 – интерпретативная;
- 3 – событийная;
- 4 – асинхронная;
- 5 – компилятивная.

112. Задание {{ 139 }}

Простейшая из моделей, позволяющая как-то учесть временные задержки в распространении сигнала через элементы схемы:

- 1 – синхронная;
- 2 – асинхронная;
- 3 – интерпретативная;
- 4 – компилятивная;
- 5 – событийная.

113. Задание {{ 140 }}

Вместо многоточия вставить наиболее подходящий термин.

«Процесс вычисления удаленности каждого данного элемента схемы от ее входов называется ...».

114. Задание {{ 141 }}

Вместо многоточия вставить наиболее подходящий термин.

«Мера удаленности данного элемента схемы от ее входов называется ... элемента»

115. Задание {{ 142 }}

Модель отдельного логического элемента можно строить на основе:

- 1 – выражений булевой алгебры;
- 2 – циклического алгоритма вычисления значений булевой функции;
- 3 – таблиц истинности.

Какой из перечисленных подходов следует применять при использовании троичного алфавита моделирования?

- 1 – никакой из перечисленных;
- 2 – первый;
- 3 – второй;
- 4 – третий;
- 5 – любой из перечисленных

116. Задание {{ 143 }}

К какому из перечисленных далее методов моделирования цифровых схем относится следующая характеристика?

“Моделирование данным методом обеспечивает получение адекватных результатов с точностью выбранного дискрета времени...”

- 1 – моделирование с использованием линий задержек;
- 2 – асинхронное моделирование;
- 3 – синхронное моделирование;
- 4 – событийное моделирование;

□ 5 – компилятивное моделирование.

117. Задание {{ 144 }}

Вместо многоточия вставить наиболее подходящий термин.

«... риск сбоя возникает в случае, когда сигнал на выходе схемы не должен измениться при смене входного сигнала, но наблюдается его кратковременное изменение (импульс)».

118. Задание {{ 145 }}

Вместо многоточия вставить наиболее подходящий термин.

«... риск сбоя предполагает возможность многократного изменения сигнала на выходе схемы, когда выходной сигнал должен был изменить свое значение на противоположное».

Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие процесс формирования компетенций

Учебная дисциплина формирует компетенции в соответствии с рабочей программой. Процедура оценивания представлена в табл. 1 и реализуется поэтапно:

1-й этап процедуры оценивания: оценивание уровня достижения каждого из запланированных результатов обучения – дескрипторов (знаний, умений, владений) в соответствии со шкалами и критериями, установленными картами компетенций ОПОП (Приложение 1 ОПОП). Экспертной оценке преподавателя подлежит сформированность отдельных индикаторов, для оценивания которых предназначена данная оценочная процедура текущего контроля и промежуточной аттестации согласно матрице соответствия оценочных средств результатам обучения.

2-й этап процедуры оценивания: интегральная оценка достижения обучающимся запланированных результатов обучения по итогам отдельных видов текущего контроля и промежуточной аттестации.

Шкала и процедура оценивания сформированности компетенций

На этапе промежуточной аттестации используется система оценки успеваемости обучающихся, которая позволяет преподавателю оценить уровень освоения материала обучающимися. Критерии оценивания сформированности планируемых результатов обучения (дескрипторов) представлены в карте компетенции ОПОП.

Форма оценки знаний: оценка - 5 «отлично»; 4 «хорошо»; 3 «удовлетворительно»; 2 «неудовлетворительно». Лабораторные работы, практические занятия, практика оцениваются: «зачет», «незачет». Возможно использование балльно-рейтинговой оценки.

Шкала оценивания:

«Зачет» – выставляется, если сформированность заявленных дескрипторов компетенций на 50 % и более оценивается не ниже «удовлетворительно» при условии отсутствия критерия «неудовлетворительно». Выставляется, когда обучающийся показывает хорошие знания изученного учебного материала; самостоятельно, логично и последовательно излагает и интерпретирует материалы учебного курса; полностью раскрывает смысл предлагаемого вопроса; владеет основными терминами и понятиями изученного курса; показывает умение переложить теоретические знания на предполагаемый практический опыт.

«Отлично» – выставляется, если сформированность заявленных дескрипторов компетенций на 80 % и более (в соответствии с картами компетенций ОПОП) оценивается критериями «хорошо» и «отлично», при условии отсутствия оценки «неудовлетворительно»: студент показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи повышенной сложности, свободно использовать справочную литературу, делать обоснованные выводы из результатов анализа конкретных ситуаций;

«Хорошо» – выставляется, если сформированность заявленных дескрипторов компетенций на 50% и более (в соответствии с картами компетенций ОПОП) оценивается критериями «хорошо» и «отлично», при условии отсутствия оценки «неудовлетворительно», допускается оценка «удовлетворительно»: обучающийся показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи, предусмотренные рабочей программой, ориентироваться в рекомендованной справочной литературе, умеет правильно оценить полученные результаты анализа конкретных ситуаций;

«Удовлетворительно» – выставляется, если сформированность заявленных дескрипторов компетенций 50 % и более (в соответствии с картами компетенций ОПОП) оценивается критериями «удовлетворительно», «хорошо» и «отлично»: обучающийся показал знание основных положений фактического материала, умение получить с помощью преподавателя правильное решение конкретной практической задачи из числа

предусмотренных рабочей программой, знакомство с рекомендованной справочной литературой;

«Неудовлетворительно», «Незачет» – выставляется, если сформированность заявленных дескрипторов компетенций менее чем 50 % (в соответствии с картами компетенций ОПОП) оценивается критериями «удовлетворительно», «хорошо» и «отлично»: при ответе обучающегося выявились существенные пробелы в знаниях основных положений фактического материала, неумение с помощью преподавателя получить правильное решение конкретной практической задачи из числа предусмотренных рабочей программой учебной дисциплины.