ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО 24153— 2012

Статистические методы

ПРОЦЕДУРЫ РАНДОМИЗАЦИИ И ОТБОРА СЛУЧАЙНОЙ ВЫБОРКИ

ISO 24153:2009
Random sampling and randomization procedures
(IDT)

Издание официальное

Москва
Стандартинформ
2014

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН Автономной некоммерческой организацией «Научно-исследовательский центр контроля и диагностики технических систем» (АНО «НИЦ КД») на основе собственного аутентичного перевода на русский язык международного стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 125 «Статистические методы в управлении качеством продукции»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 29 ноября 2012 г. № 1272-ст

4 Настоящий стандарт идентичен международному стандарту ИСО 24153:2009 «Процедуры отбора случайной выборки и рандомизации» (ISO 24153:2009 «Random sampling and randomization procedures»).

Наименование настоящего стандарта изменено относительно наименования указанного международного стандарта для приведения в соответствие с ГОСТ Р 1.5—2012 (пункт 3.5).

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов соответствующие им национальные стандарты Российской Федерации и межгосударственные стандарты, сведения о которых приведены в дополнительном приложении ДА

5 ВВЕДЕН ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в ГОСТ Р 1.0—2012 (раздел 8). Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (gost.ru)

© Стандартинформ, 2014

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения 1
2 Нормативные ссылки 1
3 Термины, определения и обозначения 2
4 Основные положения 4
5 Описание процедур случайного отбора выборки. Методы, использующие специальные устройства 5
6 Описание процедуры отбора псевдонезависимой случайной выборки. Метод таблицы 6
7 Описание процедуры отбора псевдонезависимой случайной выборки. Компьютерный метод 6
8 Применение к общим ситуациям отбора выборки 9
Приложение А (обязательное) Таблицы случайных чисел 15
Приложение В (справочное) Текст программы генерации случайных чисел 19
Приложение С (справочное) Текст программ для случайного отбора выборки и рандомизации 22
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации 28
Библиография 29

Введение

Случайный отбор выборки и процедуры рандомизации являются основой многих статистических методов, используемых при проведении экспериментов, контроле качества продукции, совершенствовании и планировании экспериментов в медицине, биологии, сельском хозяйстве и ряда научных областей. Многие стандарты на статистические методы связаны с выполнением экспериментов. В частности, все перечисленные в таблице стандарты по статистическому приемочному контролю основаны на предположении, что для отбора необходимого количества единиц продукции использован случайный отбор выборки.

Таблица — Перечень международных и соответствующих им национальных стандартов РФ по статистическому приемочному контролю

Обозначение и наименование международного стандарта	Обозначение и наименование идентичного национального стандарта
ИСО 2859-1:1999 Процедуры выборочного контроля по альтернативному признаку. Часть 1. Планы выборочного контроля последовательных партий на основе приемлемого уровня качества AQL	ГОСТ Р ИСО 2859-1—2007 Статистические методы. Процедуры выборочного контроля по альтернативному признаку. Часть 1. Планы выборочного контроля последовательных партий на основе приемлемого уровня качества
ИСО 2859-2:1985 Процедуры выборочного контроля по альтернативному признаку. Часть 2. Планы выборочного контроля отдельных партий на основе предельного качества LQ	ГОСТ Р 50779.72—99 (ИСО 2859-2—85) Статистические методы. Процедуры выборочного контроля по альтернативному признаку. Часть 2. Планы выборочного контроля отдельных партий на основе предельного качества LQ
ИСО 2859-3:2005 Процедуры выборочного контроля по альтернативному признаку. Часть 3. Выборочный контроль с пропуском партий	ГОСТ Р ИСО 2859-3—2009 Статистические методы. Процедуры выборочного контроля по альтернативному признаку. Часть 3. Контроль с пропуском партий
ИСО 2859-4:2002 Процедуры выборочного контроля по альтернативному признаку. Часть 4. Оценка соответствия заявленным уровням качества	ГОСТ Р ИСО 2859-4—2006 Статистические методы. Процедуры выборочного контроля по альтернативному признаку. Часть 4. Оценка соответствия заявленному уровню качества
ИСО 2859-5:2005 Процедуры выборочного контроля по альтернативному признаку. Часть 5. Система последовательных планов выборочного контроля на основе предела приемлемого качества (AQL) для контроля последовательных партий	ГОСТ Р ИСО 2859-5—2009 Статистические методы. Процедуры выборочного контроля по альтернативному признаку. Часть 5. Система последовательных планов на основе AQL для контроля последовательных партий
ИСО 2859-10:2006 Процедуры выборочного контроля по альтернативному признаку. Часть 10. Введение в стандарты серии ИСО 2859 на выборочный контроль по альтернативному признаку	ГОСТ Р ИСО 2859-10—2008 Статистические методы. Процедуры выборочного контроля по альтернативному признаку. Часть 10. Введение в стандарты серии ГОСТ Р ИСО 2859
ИСО 3951-1:2005 Процедуры выборочного контроля по количественному признаку. Часть 1. Требования к одноступенчатым планам на основе предела приемлемого качества (AQL) для контроля последовательных партий по единственной характеристике и единственному AQL	ГОСТ Р ИСО 3951-1—2007 Статистические методы. Процедуры выборочного контроля по количественному признаку. Часть 1. Требования к одноступенчатым планам на основе предела приемлемого качества для контроля последовательных партий по единственной характеристике и единственному AQL
ИСО 3951-2:2006 Процедуры выборочного контроля по количественному признаку. Часть 2. Общие требования к одноступенчатым планам выборочного контроля на основе предела приемлемого качества (AQL) при контроле последовательных партий по независимым характеристикам качества	ГОСТ Р ИСО 3951-2—2009 Статистические методы. Процедуры выборочного контроля по количественному признаку. Часть 2. Общие требования к одноступенчатым планам на основе AQL при контроле последовательных партий по независимым характеристикам качества
ИСО 3951-3:2007 Процедуры выборочного контроля по количественному признаку. Часть 3. Двухступенчатые схемы выборочного контроля на основе предела приемлемого качества (AQL) для контроля последовательных партий	ГОСТ Р ИСО 3951-3—2009 Статистические методы. Процедуры выборочного контроля по количественному признаку. Часть 3. Двухступенчатые схемы на основе AQL для контроля последовательных партий

Окончание таблицы

Обозначение и наименование международного стандарта	Обозначение и наименование идентичного национального стандарта
ИСО 3951-5:2006 Процедуры выборочного контроля по количественному признаку. Часть 5. Последовательные планы выборочного контроля на основе предела приемлемого качества (AQL) и известного стандартного отклонения	ГОСТ Р ИСО 3951-5—2009 Статистические методы. Процедуры выборочного контроля по количественному признаку. Часть 5. Последовательные планы на основе AQL для известного стандартного отклонения
ИСО 8422:2006 Последовательные планы выборочного контроля по альтернативному признаку	ГОСТ Р ИСО 8422—2011 Статистические методы. Последовательные планы выборочного контроля по альтернативному признаку
ИСО 8423:2008 Последовательные планы выборочного контроля по количественному признаку для процента несоответствующих единиц продукции (стандартное отклонение известно)	ГОСТ Р ИСО 8423—2011 Статистические методы. Последовательные планы выборочного контроля по количественному признаку для процента несоответствующих единиц продукции (стандартное отклонение известно)
ИСО 14560:2004 Процедуры выборочного контроля по альтернативному признаку. Уровни качества в несоответствующих единицах продукции на миллион	ГОСТ Р ИСО 14560—2007 Статистические методы. Процедуры выборочного контроля по альтернативному признаку. Уровни качества в несоответствующих единицах продукции на миллион
ИСО 18414:2006 Процедуры статистического приемочного контроля по альтернативному признаку. Система нуль-приемки на основе принципа резерва доверия к качеству продукции	ГОСТ Р ИСО 18414—2008 Статистические методы. Процедуры статистического приемочного контроля по альтернативному признаку. Система нуль-приемки на основе показателя резерва доверия к качеству продукции
ИСО 21247:2005 Комбинированные системы нуль-приемки и процедуры управления процессом при выборочном контроле продукции	ГОСТ Р ИСО 21247—2007 Статистические методы. Комбинированные системы нуль-приемки и процедуры управления процессом при выборочном контроле продукции

Кроме того, в стандартах ИСО 2859-3 и ИСО 21247 приведены условия применения случайного отбора выборки при определении возможности использования методов контроля с пропуском партии, а также для отбора единиц продукции при применении планов непрерывного выборочного контроля. Таким образом, очень важно до применения всех перечисленных стандартов обеспечить случайный отбор выборки.

Несмотря на то, что принципы настоящего стандарта универсальны и могут быть применены во всех ситуациях, где требуется случайный отбор выборки, а выборочные единицы могут быть четко определены (в виде дискретных единиц продукции), существует много ситуаций, в которых исследуемый материал не может быть предоставлен в виде дискретных единиц продукции. В таких ситуациях следует применять стандарты серии ИСО 11648 «Статистические аспекты выборочного контроля нештучной продукции»[1)] (все части).

Приведенные в настоящем стандарте процедуры рандомизации прошли проверку на соответствие заявленным свойствам, т. е. на соответствие указанному закону распределения с большим циклом повторения псевдослучайных чисел.

При использовании других процедур рандомизации следует помнить, что до их применения необходимо убедиться в качестве применяемых процедур.

Приведенные в настоящем стандарте таблицы псевдослучайных чисел могут быть использованы во всех случаях, когда это необходимо, в том числе при использовании стандартов, требующих применения таблиц случайных чисел.

[1)] В РФ действуют национальные стандарты:

ГОСТ Р ИСО 11648-1—2009 «Статистические методы. Выборочный контроль нештучной продукции. Часть 1. Общие принципы», идентичный международному стандарту ИСО 11648-1:2003 и

ГОСТ Р ИСО 11648-2—2009 «Статистические методы. Выборочный контроль нештучной продукции. Часть 2. Отбор выборки сыпучих материалов», идентичный международному стандарту ИСО 11648-2:2001.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Статистические методы

ПРОЦЕДУРЫ РАНДОМИЗАЦИИ И ОТБОРА СЛУЧАЙНОЙ ВЫБОРКИ

Statistical methods. Randomization and random sampling procedures

Дата введения — 2013—12—01

1 Область применения

Настоящий стандарт устанавливает процедуры рандомизации и случайного отбора выборки. Представленные в настоящем стандарте процедуры охватывают методы, предусматривающие применение технических средств, таблиц случайных чисел, компьютерных программ. Для компьютерных программ приведены общее описание соответствующего алгоритма и текст программы.

Настоящий стандарт применим в ситуациях, когда в требованиях технического регулирования, договора или другого стандарта установлено применение случайного отбора выборки или рандомизации. Методы случайного отбора выборки применимы при:

a) проведении статистического приемочного контроля дискретных единиц продукции, представленных на контроль в партиях;

b) отборе выборки для целей наблюдений и исследований;

c) проведении аудита системы менеджмента качества;

d) выборе экспериментальных единиц, порядка их обработки и проверки при выполнении плана эксперимента.

В настоящий стандарт также включена информация по случайному отбору выборки или рандомизации результатов, которая может быть использована специалистами службы качества и контролирующих органов при проведении аудита или проверок.

В настоящем стандарте представлены не все возможные способы применения процедур случайного отбора выборки и рандомизации и не все возможные стратегии отбора выборки и определения объема выборки. В соответствующих ситуациях следует использовать стандарты, перечисленные во введении.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты:

ИСО 3534-1 Статистика. Словарь и условные обозначения. Часть 1. Общие статистические термины и термины, используемые в вероятностных задачах (ISO 3534-1, Statistics — Vocabulary and symbols — Part 1: General statistical terms and terms used in probability)

ИСО 3534-2 Статистика. Словарь и условные обозначения. Часть 2. Прикладная статистика (ISO 3534-2, Statistics — Vocabulary and symbols — Part 2: Applied statistics)

ИСО 3534-3 Статистика. Словарь и условные обозначения. Часть 3. Планирование эксперимента (ISO 3534-3, Statistics — Vocabulary and symbols — Part 3: Design of experiments)

ИСО 80000-2 Величины и единицы. Часть 2. Математические символы и знаки для применения в естественных науках и технологиях (ISO 80000-2, Quantities and units — Part 2: Mathematical signs and symbols to be used in the natural sciences and technology)

Издание официальное

3 Термины, определения и обозначения

В настоящем стандарте применены термины по ИСО 3534-1, ИСО 3534-2, ИСО 3534-3, а также следующие термины с соответствующими определениями.

3.1 Термины и определения

3.1.1 **кластер** (cluster): Часть совокупности (3.1.6), представляющей собой объединение непересекающихся групп выборочных единиц (3.1.13), выделенная в соответствии с установленным правилом.

[ИСО 3534-2:2006, 1.2.28]

3.1.2 **кластерный отбор** (cluster sampling): Отбор выборки (3.1.12), при котором отбирают случайные выборки (3.1.8) из кластеров (3.1.1), и все отобранные выборочные единицы (3.1.13) объединяют в общую выборку (3.1.11).

[ИСО 3534-2:2006, 1.3.9]

3.1.3 **изменение порядка, полная перестановка** (derangement complete permutation): Такая перестановка элементов набора[1)], при которой ни один элемент не остается на своем исходном месте (например, набор цифр {3, 1, 2} представляет собой полную перестановку элементов набора {1, 2, 3}).

3.1.4 **контролируемая партия** (lot): Определенная часть совокупности (3.1.6), составленная для выборочного контроля и отражающая свойства совокупности.

Примечание — Целями выборочного контроля могут быть, например, определение оценки среднего некоторой характеристики или принятие решения о приемке или отклонении партии.

[ИСО 3534-2:2006, 1.2.4]

3.1.5 **многостадийный отбор** (multistage sampling): Отбор (3.1.12), при котором выборку (3.1.11) отбирают в несколько приемов (стадий), выборочные единицы (3.1.13) на каждой стадии отбирают из более крупных выборочных единиц, отобранных на предыдущей стадии.

Примечание — Многостадийный отбор выборки отличается от многокритериального отбора выборки. Многокритериальный отбор выборки представляет собой отбор выборки в соответствии с несколькими критериями одновременно.

[ИСО 3534-2:2006, 1.3.10]

3.1.6 **совокупность** (population): <reference> Множество всех исследуемых единиц.

[ИСО 3534-2:2006, 1.2.1]

3.1.7 **псевдонезависимый случайный отбор выборки** (pseudo-independent random sampling): Отбор выборки (3.1.12), при котором выборку (3.1.11) из *n* выборочных единиц (3.1.13) отбирают из совокупности (3.1.6) в соответствии с таблицей случайных чисел или компьютерным алгоритмом, разработанным таким образом, что каждая из возможных комбинаций по *n* выборочных единиц имеет установленную вероятность отбора (см. также 4.4).

3.1.8 **случайная выборка** (random sample): Выборка (3.1.11), отобранная с помощью случайного отбора выборки (3.1.9).

3.1.9 **случайный отбор выборки** (random sampling): Отбор выборки (3.1.12), при котором выборку (3.1.11) из *n* выборочных единиц (3.1.13) отбирают из совокупности (3.1.6) таким образом, что все возможные комбинации по *n* выборочных единиц имеют равную вероятность быть отобранными.

[ИСО 3534-2:2006, 1.3.5]

3.1.10 **рандомизация** (randomization): Процесс, с помощью которого единицы набора размещают в случайном порядке.

Примечание — Если из совокупности (3.1.6), состоящей из натуральных чисел от 1 до *n*, извлекать числа случайным образом (т. е. таким способом, при котором все числа имеют одинаковые шансы быть отобранными) одно за другим без возвращения, пока совокупность не будет исчерпана, то порядок отбора чисел называют случайным или рандомизированным.

Если эти *n* чисел связаны с *n* различными объектами или обработками, которые устанавливают в соответствии с порядком этих *n* чисел, то порядок объектов или обработок называют случайным или рандомизированным.

[1)] Набор — конечное множество элементов.

3.1.11 **выборка** (sample): Подмножество совокупности (3.1.6), состоящее из одной или нескольких выборочных единиц (3.1.13).

[ИСО 3534-2:2006, 1.2.17]

3.1.12 **отбор выборки** (sampling): Действия по извлечению или составлению выборки (3.1.11).

[ИСО 3534-2:2006, 1.3.1]

3.1.13 **выборочная единица, единица** (sampling unit, unit): Один из элементов, составляющих выборку (3.1.11).

Примечание 1 — Выборочная единица может содержать один или несколько объектов (например, коробок спичек), но для нее получают один результат испытаний или наблюдений.

Примечание 2 — Выборочная единица может состоять из дискретных объектов или определенного количества нештучной продукции (материала).

[ИСО 3534-2:2006, 1.2.14]

3.1.14 **отбор с возвращением** (sampling with replacement): Отбор выборки (3.1.12), при котором каждую выборочную единицу (3.1.13) после отбора возвращают в совокупность (3.1.6) до отбора следующей выборочной единицы.

[ИСО 3534-2:2006, 1.3.15]

3.1.15 **отбор без возвращения** (sampling without replacement): Отбор выборки (3.1.12), при котором каждую выборочную единицу (3.1.13) отбирают из совокупности (3.1.6) только один раз (не возвращая ее в совокупность до отбора следующей выборочной единицы).

3.1.16 **начальное число** (seed): Число или набор чисел, используемых для инициализации алгоритма, используемого для псевдонезависимого случайного отбора выборки (3.1.7) или установления исходной точки в таблице случайных чисел.

3.1.17 **простая случайная выборка** (simple random sample): Выборка (3.1.11), отобранная методом отбора простой случайной выборки (3.1.18).

[ИСО 3534-2:2006, 1.2.24]

3.1.18 **отбор простой случайной выборки** (simple random sampling): Отбор выборки (3.1.12), при котором выборку (3.1.11) из n выборочных единиц (3.1.13) отбирают из совокупности (3.1.6) таким способом, что все возможные комбинации из n выборочных единиц имеют одинаковую вероятность быть отобранными.

[ИСО 3534-2:2006, 1.3.4]

3.1.19 **расслоенная выборка** (stratified sampling): Выборка (3.1.11), отобранная таким образом, что выборочные единицы (3.1.13) отбирают из различных слоев (3.1.21) совокупности, причем каждый слой представлен в выборке хотя бы одной выборочной единицей.

[ИСО 3534-2:2006, 1.3.6]

3.1.20 **отбор расслоенной простой случайной выборки** (stratified simple random sampling): Отбор выборки, при котором выборку формируют на основе отбора простой случайной выборки (3.1.18) из каждого слоя (3.1.21) совокупности.

[ИСО 3534-2:2006, 1.3.7]

3.1.21 **слой** (stratum): Исчерпывающая часть совокупности, являющаяся более однородной относительно исследуемой характеристики, чем вся совокупность (3.1.6)[1)].

[ИСО 3534-2:2006, 1.2.29]

3.2 Обозначения

В настоящем стандарте использованы математические символы и обозначения в соответствии с ИСО 80000-2, а также следующие обозначения:

d_i — i-я цифра, количество сторон монеты или граней игральной кости;

N — объем партии;

n — объем выборки;

n_j — объем j-ой выборки;

U — равномерно распределенная случайная величина на открытом интервале (0, 1);

x_i — i-е значение случайной величины x;

$j!$ — факториал j;

[1)] Слои совокупности являются непересекающимися, т. е. не содержат общих единиц.

$\lceil z \rceil$ — функция округления z до ближайшего целого в большую сторону (наименьшее целое число, больше или равное действительному числу z);

$\lfloor z \rfloor$ — функция округления z до ближайшего целого в меньшую сторону (наибольшее целое число, равное действительному числу z или меньше его).

4 Основные положения

4.1 Применение случайного отбора выборки является необходимым условием правильного применения большей части планов выборочного контроля при контроле продукции на промышленных предприятиях. Применение рандомизации, использующей принципы случайного отбора выборки, необходимо также при планировании экспериментов, поскольку это повышает достоверность результатов эксперимента и обеспечивает использование статистических методов для их интерпретации. Целью случайного отбора выборки является обеспечение возможности применения теории вероятностей к практическим задачам. Эта цель не достижима при использовании других способов отбора выборки. Например, отбор выборки, основанный на принципах интуиции или личных предпочтений, позволяет случайно достичь хороших результатов, но при этом полученные оценки всегда имеют смещение и, следовательно, такой отбор выборки может привести к серьезным ошибкам при принятии решений. Случайный отбор выборки направлен на устранение смещения путем обеспечения равной вероятности отбора в выборку для каждой единицы (отбор с возвращением) или для каждой выборки данного объема (отбор без возвращения).

4.2 Равновероятным случайным отбором с возвращением является отбор, при котором вероятность отбора конкретной единицы партии, состоящей из N единиц, всегда равна $1/N$. Существует N^n возможных вариантов отбора упорядоченной выборки объема n из N единиц и $(N + n - 1)!/[n!(N - 1)!]$ вариантов отбора неупорядоченной случайной выборки объема n из N единиц (см. примечание ниже).

При отборе простой случайной выборки без возвращения вероятность того, что конкретная единица из партии объема N будет отобрана при отборе первой выборочной единицы, равна $1/N$, $1/(N - 1)$ — при отборе второй, $1/(N - 2)$ — третьей и так далее. Если n выборочных единиц отобраны случайным образом из партии объема N без возвращения, то каждой комбинации из n единиц соответствует одинаковая вероятность отбора, равная вероятности одновременного отбора n единиц из N. Количество возможных различных неупорядоченных случайных выборок по n единиц из партии объема N (отличающихся только составом) равно $N!/[n!(N - n)!]$, что представляет собой число сочетаний по n единиц из N. Количество возможных упорядоченных случайных выборок по n единиц, отобранных из партии в N единиц, без возвращения (отличающихся и составом и порядком расположения в выборке) равно $N!/(N - n)!$, т. е. равно произведению числа сочетаний по n единиц из N на число перестановок из n единиц. Следует отметить, что случайная выборка без возвращения более всего распространена в качестве стратегии отбора выборки при статистическом приемочном контроле.

Примечание — При отборе выборки с возвращением, например, трех из пяти единиц, выборки {1, 1, 2}, {1, 2, 1}, и {2, 1, 1} различны, если учитывают порядок элементов, однако эти выборки не имеют различий, если порядок элементов не учитывают.

4.3 Случайный отбор выборки может быть получен только при строгом выполнении разработанных специальных процедур. В настоящем стандарте представлено несколько методик случайного отбора выборки. Методы, использующие специальные устройства, предполагают, что монеты и игральные кости имеют равные вероятности выпадения граней или сторон при броске, а сам бросок не вносит смещения. Учитывая трудности непосредственного выполнения методов отбора выборки с использованием калькулятора или компьютера (см. [9], [10], [12] и [13]), в настоящем стандарте приведен простой компьютерный метод генерации случайной выборки. Следует помнить, что при использовании всех приведенных ниже методов необходимо, чтобы каждой единице партии заранее было поставлено в соответствие число от 1 до N, так, чтобы выборочные единицы в партии в результате были однозначно идентифицированы.

4.4 При описании процедуры или метода случайного отбора выборки (см. [8]) прилагательное «псевдонезависимый» в тексте настоящего стандарта часто опущено. Кроме того, прилагательное «случайный» часто использовано в том смысле, что соответствующее существительное (число или перестановка) является результатом процесса, который случайным образом генерирует такое число или перестановку. В представленных примерах для наглядности использованы небольшие объемы выборки.

5 Описание процедур случайного отбора выборки. Методы, использующие специальные устройства

5.1 Метод урны

5.1.1 В урну помещают N четко пронумерованных идентичных объектов (например билеты, пластинки, шары), однозначно представляющих каждую из N единиц партии и хорошо их перемешивают.

5.1.2 Для отбора без возвращения, вслепую выбирают объекты из урны, один за другим, не возвращая их в урну, и произвольно перемешивая объекты между выемками, пока не будет получено необходимое количество n выборочных единиц.

Примечание — Этот метод обычно используют в лотереях.

5.1.3 Для отбора с возвращением, вслепую выбирают объекты из урны один за другим, возвращая отобранный объект в урну после каждой выемки, и полностью перемешивая объекты между выемками, пока не будет получено необходимое количество n выборочных единиц. При использовании этого метода одна и та же единица может быть отобрана в выборку несколько раз.

5.2 Метод монет или игральных костей

5.2.1 Определяют количество m монет или игральных костей (бросков монеты или игральной кости) по следующей формуле (N — объем партии, k — количество сторон или граней используемого объекта)

$$m = \lceil \ln N / \ln k \rceil.$$

5.2.2 При использовании нескольких монет или игральных костей устанавливают четкое соответствие каждой монеты или кости определенной цифре в последовательности цифр ($d_1, \ldots, d_m$). При использовании единственной монеты или кости устанавливают соответствие результата первого броска цифре d_1, второго броска — цифре d_2 и так далее.

5.2.3 Бросают монету или игральную кость и записывают m значений d_i ($i = \overline{1,m}$). Трансформируют полученные результаты в десятичное целое число по следующей формуле

$$y = 1 + \sum_{i=1}^{m} (d_i - 1) k^{m-i}.$$

5.2.4 Повторяют действия в соответствии с 5.2.3, отбрасывая все значения, превышающие N, и (при отборе выборки без возвращения) все значения, отобранные ранее, до получения n выборочных единиц.

Пример 1 — Контролеру необходима случайная выборка объемом 4 единицы из партии в 20 единиц. Для формирования выборки он использует единственную монету. В соответствии с 5.2.1 для получения каждого случайного числа необходимо выполнить m = 5 бросков монеты. Определено, что одна сторона монеты соответствует цифре 1, а другая – цифре 2. Первая последовательность бросков дает набор чисел {1, 2, 1, 2, 2}, который в соответствии с 5.2.3 дает число 1 + (0) · 2^4 + (1) · 2^3 + (0) · 2^2 + (1) · 2^1 + + (1) · 2^0 = 12. Следующие три последовательности бросков дают наборы {1, 2, 2, 2, 1}, {1, 1, 2, 2, 1} и {2, 2, 1, 2, 2}, которые дают числа 15, 7 и 28 соответственно. Так как значение 28 превышает объем партии, его отбрасывают, и выполняют броски, пока не будет получено еще одно действительное число, необходимое для формирования случайной выборки.

Пример 2 — Необходимо отобрать случайную выборку с объемом 4 единицы из партии в 50 единиц. Для формирования выборки контролер использует несколько шестигранных игральных костей различного цвета. В соответствии с 5.2.1 необходимо для получения каждого случайного числа m = 3 игральные кости. Выпавшее значение синей, зеленой и красной игральных костей контролер записывает в том же порядке (синяя, зеленая, красная). Однако уравнение 5.2.3 позволяет получить действительные числа из интервала от 1 до 50 только в том случае, когда первая игральная кость дает значения 1 или 2. Следовательно, работа может быть упрощена, если выпавшие значения синей кости будут преобразованы в 1 или 2 без изменения вероятности результата. Контролер решает заранее, что нечетные выпавшие значения синей кости будет рассматривать как 1, а четные — как 2. Первый бросок дает {3, 3, 4}, который в соответствии с 5.2.3 дает число 1 + (2)·6^2 + (2)·6^1 + (3)·6^0 = 88 (слишком большое), которое после преобразования к {1, 3, 4} дает 16. Еще три броска дают {6, 1, 3} (после преобразования {2, 1, 3}), {5, 6, 6} (после преобразования {1, 6, 6}) и {2, 5, 5} (после преобразования {2, 5, 5}), которые в соответствии с 5.2.3 дают значения 39, 36 и 65 соответственно. Так как значение 65 превышает объем партии, его необходимо отбросить и выполнить дополнительные броски, пока не будет получено еще одно действительное число.

Пример 3 — В условиях задачи, рассмотренной в примере 2, очевидно, что использование трех игральных костей позволяет получить числа от 1 до $6^3 = 216$ (при объеме партии в 50 единиц). Контролер принимает решение фиксировать все результаты от 1 до 200, преобразуя их в числа из интервала от 1 до 50, и отбрасывать результаты более 200, чтобы избежать вероятностного искажения результатов. Четыре броска предыдущего примера в соответствии с этой схемой дают {3, 3, 4},{6, 1, 3}, {5, 6, 6} и {2, 5, 5}, соответствующие числам 88, 183, 180 и 65. Из этих чисел произведено вычитание числа 50 столько раз, пока число не окажется в интервале от 1 до 50 (0 интерпретируют как N), что дает значения 38, 33, 30 и 15 соответственно. Таким образом, была получена выборка из 4 единиц, т. е. дальнейшие броски не нужны. Очевидно, что математически этот процесс эквивалентен применению уравнения $v_2 = 1 + (v_1 - 1) \bmod N$, где v_1 — исходное значение, а v_2 — значение из интервала от 1 до N.

6 Описание процедуры отбора псевдонезависимой случайной выборки. Метод таблицы

6.1 Таблицы случайных чисел

В приложении А приведены две таблицы случайных чисел. В каждой таблице приведено 3600 случайных чисел от 0 до 9, указанных в 60 строках по 60 колонок в каждой. Использование таблиц кратко описано ниже и более подробно в приложении А.

Примечание — Числа, приведенные в таблице, аналогичны значениям, выпадающим при бросании 10-гранного кубика. Количество цифр *m*, необходимых для отбора выборки, соответствует количеству бросков игральной кости.

6.2 Описание метода

6.2.1 Определяют количество цифр *m*, соответствующее объему партии *N*. Если объем партии является степенью 10, отбрасывают начальную цифру объема партии и интерпретируют оставшиеся нули при их появлении, как цифру, равную объему партии (например, если $N = 1000$, значение 000 при его появлении интерпретируют как 1000).

6.2.2 Случайным образом выбирают начальную точку (т. е. например, номер строки и колонки в таблице), используя метод, описанный в А.2.2.

6.2.3 Полученную цифру вместе с $(m - 1)$ цифрами справа от нее записывают как единое число. Если правее начальной цифры (до 60-ой колонки) недостаточно цифр для получения необходимого числа, используют колонки 1, 2 и так далее вместо колонок 61, 62 и так далее соответственно.

6.2.4 Увеличивают номер строки на единицу, повторяют 6.2.3 и записывают полученное значение. Если полученный номер строки превышает 60, в качестве следующей используют строку 1 и увеличивают номер каждой колонки на *m*.

6.2.5 Повторяют 6.2.4, отбрасывая все значения более *N* и, в случае отбора выборки без возвращения, все значения, которые были уже отобраны, пока необходимое количество выборочных единиц *n* не будет получено.

Пример — Контролеру необходимо отобрать случайную выборку объема 5 единиц из партии в 200 единиц. Случайной начальной точкой, определенной бросанием монеты, является число, расположенное в строке 57 и колонке 59 таблицы 1. Так как N много меньше максимального значения, которое можно получить на основе 3-х цифр (т. е. 1000), аудитор принимает решение использовать результаты из интервала от 1 до 1000, переводя их в интервал от 1 до 200. Получены следующие пять чисел: 848, 670, 902, 034 и 518. После перевода их в интервал от 1 до 200 получены значения 48, 70, 102, 34 и 118.

7 Описание процедуры отбора псевдонезависимой случайной выборки. Компьютерный метод

7.1 Краткий обзор

7.1.1 В настоящем стандарте использованы алгоритмы, приведенные в [1], [7] и [13]. Эти алгоритмы обладают математическими и статистическими свойствами, необходимыми для обеспечения случайного отбора выборки, создания программ на различных языках программирования и компьютерах, позволяющих проводить верификацию и ревизию отобранных значений, используемых для проверок, контроля, аудита. Текст основных элементов программы на языке программирования Си приведен в приложении В.

7.1.2 В алгоритмах использованы два основных элемента:

a) алгоритм инициализации, генерирующий квазислучайное начальное целое число на основе времени, прошедшего с указанной даты;

b) генератор случайных чисел.

7.1.3 При использовании для верификации или аудита алгоритм инициализации, упомянутый в 7.1.2 a) и описанный в 7.2, может быть заменен введением начального значения вручную. Это значение должно быть целым числом из интервала от 1 до 2 147 483 398 включительно. Это значение необходимо сохранять и указывать в отчете (при составлении). Однако при использовании для контроля качества и при планировании эксперимента должна быть предусмотрена возможность не использовать компьютерный выбор начального значения, который может быть выбран по умолчанию.

Примечание — Этапы алгоритмов приведены в более формализованном виде, что может быть полезно при программировании методов. Текст программы со ссылками на подразделы настоящего стандарта приведен в приложении В.

7.2 Алгоритм инициализации

7.2.1 Алгоритм инициализации включает:

a) алгоритм вычисления времени, прошедшего с указанной даты;

b) алгоритм генерации случайных чисел на основе равномерного распределения, использующий случайное число, полученное на этапе a).

7.2.2 Приведенный алгоритм позволяет определить количество секунд с 2000-01-01 00:00:00 до текущих даты и времени.

Описание алгоритма.

a) Определяют системные дату и время компьютера в виде символьной переменной, сохраняют значение этой переменной и указывают в отчете, а затем анализируют переменную по компонентам (год, месяц, день, час, минута, секунда).

b) Вычисляют количество полных дней d_e, начиная с указанной даты, используя для текущей даты числовые значения года y (с четырьмя цифрами), месяца m_1 и дня d, обработанные следующим образом:

если $m_1 < 3$, то $m_1 = m_1 + 12$, $y = y - 1$,

$$d_e = d + \lfloor (153m_1 - 457)/5 \rfloor + 365y + \lfloor y/4 \rfloor - \lfloor y/100 \rfloor + \lfloor y/400 \rfloor - 730\,426.$$

Примечание — Формула для d_e может быть упрощена для календарных лет до 2099 года включительно заменой членов после $\lfloor y/4 \rfloor$ на « –730 441».

c) Вычисляют общее количество секунд s_e, начиная с указанной даты, используя расчеты в соответствии с b) и время дня (в 24-часовом формате «hh:mm:ss»), указанное в символьной переменной в соответствии с a) на основе формулы

$$s_e = 86400d_e + 3600h + 60m_2 + s,$$

где h, m_2 и s — часы, минуты и секунды соответственно.

Примечание 1 — В некоторых языках программирования есть встроенные функции для вычисления s_e непосредственно. Перед использованием такие встроенные функции должны быть проверены для обеспечения учета влияния високосных лет, летнего/зимнего времени (в зависимости от обстоятельств).

Примечание 2 — В 32-битовой реализации этого алгоритма значение s_e увеличивается до переполнения соответствующей ячейки. В программе необходимо предусмотреть проверки того, что это число находится в интервале от 1 до 2 147 483 398 включительно.

d) Значение, полученное в соответствии с c), является начальным числом для генератора начального числа и используется для получения заключительного начального числа. Это значение хранят в виде отдельной переменной и указывают в отчете.

e) Количество j обращений к генератору случайных чисел является случайным целым от 1 до 100, полученным на основе двух наименее существенных цифр значения, полученного в соответствии с перечислением c), увеличенного на 1, и может быть рассчитано по формуле

$$j = s_e - 100\lfloor s_e/100 \rfloor + 1.$$

7.2.3 Алгоритм генерации начального числа (функция инициализации) представляет собой линейное конгруэнтное рекуррентное отношение.

Описание алгоритма (для компьютеров, способных обрабатывать 32-битовые целые числа).

a) $x_{i+1} = 40692\ x_i \bmod 2\ 147\ 483\ 399$;

b) $k = \lfloor x_i / 52774 \rfloor$;

c) $x_{i+1} = 40692\ (x_i - 52774\ k) - 3791\ k$;

d) если $x_{i+1} < 0$, то $x_{i+1} = x_{i+1} + 2\ 147\ 483\ 399$.

7.2.4 Генерируют начальное число алгоритма случайного отбора выборки, выбирая результат 7.2.2 c) в качестве x_i и затем обращаясь к формуле 7.2.3 j раз (см. 7.2.2 e)), заменяя каждый раз x_i на x_{i+1} до тех пор, пока не будет выполнено необходимое количество обращений.

7.2.5 Заключительное значение x_{i+1} на этапе 7.2.4 является случайным целым числом из интервала от 1 до 2 147 483 398 и может быть использовано как начальное число в алгоритме отбора выборки, описанном в 7.3 (в частности, как значение y_i в 7.3.6 b)). Это значение сохраняют как отдельную переменную и указывают в отчете.

7.3 Алгоритм генерации случайных чисел

7.3.1 Алгоритм генерации случайных чисел включает

a) сохранение массива чисел, полученных с помощью алгоритма генерации равномерно распределенных случайных чисел;

b) перестановки чисел, полученных на основе алгоритма генерации равномерно распределенных случайных чисел.

7.3.2 Создают массив A из 32 элементов, который используют для сохранения выходных данных алгоритма случайного отбора выборки.

7.3.3 Для формирования массива A используют следующий генератор случайных чисел.

Описание алгоритма (для компьютеров, способных обрабатывать 32-битовые целые числа):

a) $x_{i+1} = 40\ 014\ x_i \bmod 2\ 147\ 483\ 563$;

b) $k = \lfloor x_i / 53\ 668 \rfloor$;

c) $x_{i+1} = 40\ 014\ (x_i - 53\ 668k) - 12\ 211k$;

d) если $x_{i+1} < 0$, то $x_{i+1} = x_{i+1} + 2\ 147\ 483\ 563$.

7.3.4 Инициализируют массив A, присваивая x_i результат 7.1.3 или 7.2.5, и затем обращаются к генератору в соответствии с 7.3.3 a) 40 раз, заменяя x_i на x_{i+1} при каждом обращении, отбрасывая первые 8 значений, а затем присваивая каждое из оставшихся в массиве 32 значений x_{i+1} в обратном порядке (т. е. с 32-го элемента до 1-го элемента).

7.3.5 Выбирают первый элемент из массива A (т. е. $A[1]$) в качестве значения k, инициализирующего комбинированный алгоритм генерации случайных чисел.

7.3.6 Комбинированный генератор случайных чисел для генерации случайной выборки представляет собой комбинацию линейных конгруэнтных рекуррентных соотношений и определения индекса элемента массива A.

Описание алгоритма:

a) $x_{i+1} = 40\ 014\ x_i \bmod 2\ 147\ 483\ 563$;

b) $y_{i+1} = 40\ 692\ y_i \bmod 2\ 147\ 483\ 399$;

c) $J = \lfloor 32\ k / 2\ 147\ 483\ 563 \rfloor + 1$;

d) $k = A[J] - y_{i+1}$;

e) $A[J] = x_{i+1}$;

f) если $k < 1$, то $k = k + 2\ 147\ 483\ 562$.

7.3.7 Алгоритм в соответствии с 7.3.6 инициализируют путем присвоения x_i значения x_{i+1}, полученного в соответствии с 7.3.4, и присвоения y_i значения в соответствии с 7.2.5. Значения x_{i+1} и y_{i+1} служат далее значениями x_i и y_i для всех последующих обращений к алгоритму. Случайный индекс J для элемента массива A вычисляют, используя значение k (см. 7.3.5), которому присваивают значение, равное разности $(A[J] - y_{i+1})$, а $A[J]$ заменяют на x_{i+1}. Наконец, при необходимости получения положительного значения, знак k меняют на противоположный.

7.3.8 Входом алгоритма случайного отбора выборки является значение k, представляющее собой случайное число из интервала от 1 до 2 147 483 562, трансформированное в действительное равномерно распределенное случайное число U из интервала от 0 до 1 (исключая конечные точки этого интервала) следующим образом: $U = k/2\ 147\ 483\ 563$.

7.3.9 Выход 7.3.8 может быть представлен в виде случайного целого числа L, равномерно распределенного в интервале от 1 до N включительно, следующим образом:

$$L = \lfloor N\ U \rfloor + 1.$$

7.3.10 Выполняют генерацию случайной выборки в соответствии с 7.3.6—7.3.9, повторяя эти действия до тех пор, пока не будет получено необходимое количество случайных чисел.

7.4 Отчет

Если отчет необходимо сохранять для аудита или других проверок установленных требований, необходимо записывать объем партий и объем выборки.

Кроме того, следует записывать значение начального числа, введенного вручную в соответствии с 7.1.3 или полученного с помощью случайного генератора начального числа.

В отчете также указывают:

a) системные дату и время компьютера, используемые для вычисления начального числа;

b) значение начального числа в 7.2.2 d);

c) заключительное значение в соответствии с 7.2.5.

8 Применение к общим ситуациям отбора выборки

8.1 Общие положения

8.1.1 Ниже приведены алгоритмы для нескольких стратегий отбора случайной выборки в различных практических ситуациях.

8.1.2 В настоящем разделе U — случайная действительная переменная, равномерно распределенная в интервале от 0 до 1 за исключением значений в конечных точках интервала (см. 7.3). Если для определения U использован другой источник и известно, что полученные значения лежат в интервале, не включающем точку 0, принимают $U = (1 - U)$. Если использованный источник значений U включает 0 и 1 (обе конечные точки интервала), значение 1 должно быть исключено.

8.2 Случайное целое из интервала

Приведенный алгоритм позволяет генерировать случайное целое K из интервала от M до N включительно.

Описание алгоритма.

a) Генерируют случайное действительное число U.

b) Вычисляют $K = M + \lfloor U\ (N - M + 1) \rfloor$.

8.3 Случайная перестановка

Приведенный алгоритм позволяет получить случайную перестановку из N единиц по n для массива A с N различными элементами.

Описание алгоритма:

a) Формируют массив $A[1{:}N]$, состоящий из N индексированных элементов, расположенных в исходном порядке.

b) Присваивают J значение 1.

c) Генерируют случайное целое число K из интервала от J до N включительно.

d) Значения $A[J]$ и $A[K]$ меняют местами.

e) Увеличивают J на 1.

f) Если J меньше или равно n, возвращаются к c).

g) Получена случайная перестановка из первых n значений массива A.

8.4 Случайное изменение порядка

Приведенный алгоритм позволяет для массива A из N различных элементов произвести случайное изменение порядка N элементов.

Описание алгоритма:

a) Формируют из N индексированных элементов в исходном порядке массив $A[1:N]$ и создают копию массива $A[1:N]$ в массиве $B[1:N]$.

b) Выполняют случайную перестановку N элементов из N (т. е. всех) массива $B[1:N]$ в соответствии с 8.3.

c) Сравнивают элементы с 1-го до N-го массивов A и B для выявления равных.

d) Если элемент массива B равен соответствующему элементу массива A, прекращают сравнение и переходят к выполнению b).

e) Получен массив B со случайным изменением порядка элементов массива A.

Примечание — Этот алгоритм может быть более эффективным при выполнении b) и c), если сравнение элементов $A[J]$ и $B[J]$ выполнять сразу после определения $B[J]$, не дожидаясь полной перестановки набора B.

8.5 Случайный отбор с возвращением

Приведенный алгоритм позволяет произвести случайный отбор с возвращением единственной выборки из n единиц из партии объема N.

Описание алгоритма.

a) Генерируют случайное целое число K из интервала от 1 до N включительно.

b) Повторяют a) до тех пор, пока не будет получено n значений K.

Примечание — Этот метод может быть применен неоднократно для получения любого количества выборок любого объема. Если полученные значения единственной выборки не ранжированы, такая выборка может быть использована для последовательного выборочного контроля.

8.6 Случайная выборка без возвращения

Приведенный алгоритм позволяет произвести случайный отбор единственной выборки, состоящей из n различных единиц, из партии N единиц.

Описание алгоритма.

a) Метод 1.

1) Генерируют случайное целое число K из интервала от 1 до N включительно.

2) Проверяют, что это значение K не было ранее генерировано. Если это значение K не было ранее генерировано, его сохраняют, в противном случае — отбрасывают.

3) Повторяют 1) и 2) до тех пор, пока не будет получено n различных значений K.

b) Метод 2.

1) Выполняют случайную перестановку n единиц из N в соответствии с 8.3.

2) Используют первые n значений массива A в качестве случайной выборки.

Примечание — Каждый из этих методов может быть использован для получения любого количества выборок различного объема (для таких целей, как двухступенчатый или многоступенчатый контроль). При использовании общего количества n_i индивидуальных объемов выборки n_i в качестве входного значения n алгоритма, генерирующего неупорядоченные данные, первые n_1 полученных значений являются первой выборкой, следующие n_2 значений — второй выборкой и т. д. Кроме того, если полученные значения единственной выборки не ранжированы, то выборка может быть использована для последовательного выборочного контроля.

8.7 Случайный отбор выборки для планов непрерывного выборочного контроля (CSP[1])

План непрерывного выборочного контроля CSP-1 разработан для применения при контроле качества продукции производственной линии и представляет собой смену периодов сплошного и выборочного контроля (после i последовательно принятых единиц продукции сплошной контроль заменяют на выборочный контроль с долей отбора f, который снова заменяют на сплошной контроль после обнаружения несоответствующей единицы продукции). Ниже приведено описание двух методов, позволяющих отбирать продукцию для выборочного контроля.

Описание алгоритма.

a) Метод 1.

1) Для каждой единицы продукции генерируют случайное действительное значение U ($U \in [0,1]$).

2) Если U меньше или равно f, соответствующую единицу выбирают для выборочного контроля.

---

[1] CSP — continuous sampling plans.

3) Повторяют 1) и 2) до тех пор, пока не будет обнаружена несоответствующая единица продукции.

b) Метод 2.

1) Для каждой группы из n единиц продукции, где $n = 1/f$, генерируют случайное целое число K из интервала от 1 до n включительно.

2) Выбирают единицу, соответствующую значению K, в качестве выборочной единицы для контроля.

3) Повторяют 1) и 2) до тех пор, пока не будет получена несоответствующая единица продукции.

Примечание — Для планов CSP-1 значение f устанавливают равным числу, обратному к целому числу.

8.8 Отбор расслоенной случайной выборки

Для партии, составленной из двух или более слоев объема N_i, отбирают единственную случайную выборку объема n_i из каждого i-го слоя, используя методы, приведенные в 8.3 или 8.6, если необходим отбор выборки без возвращения, или метод, приведенный в 8.5, если необходим отбор с возвращением.

8.9 Отбор единственной случайной выборки из партии с первоначально неизвестным объемом

Приведенный алгоритм позволяет получить единственную случайную выборку, состоящую из n различных единиц, отобранную из партии, первоначальный объем которой неизвестен (но не менее n).

Описание алгоритма (адаптированный метод из [11]):

a) Определяют первые n единиц в партии и формируют массив $A[1{:}n]$.

b) Если в партии существует еще единица продукции, устанавливают N равным номеру следующей единицы, в противном случае переходят к f).

c) Генерируют случайное целое число K из интервала от 1 до N включительно.

d) Если $K > n$, устанавливают $A[K] = N$.

e) Переходят к выполнению b).

f) Получена случайная выборка объема n в массиве A при объеме партии не менее N.

Примечание — Данный метод может быть также использован, если объем партии известен.

8.10 Отбор единственной упорядоченной случайной выборки без возвращения

Приведенный алгоритм позволяет получить единственную случайную выборку n различных единиц из партии объема N, генерированную сразу в порядке возрастания.

Описание алгоритмов:

a) Метод 1 (адаптированный метод из [2]).

1) Инициализируют следующие переменные:

i) создают массив $A[1{:}n]$;

ii) устанавливают $L = N$, $K = (N - n)$ и $J = 0$.

2) Увеличивают значение J на 1.

3) Если $J \geq n$, переходят к 8).

4) Генерируют случайное действительное число U и устанавливают $P = 1$.

5) Устанавливают $P = P \cdot K/N$.

6) Если $P \leq U$:

i) устанавливают $A[J] = (N - L + 1)$ и уменьшают L на 1;

ii) переходят к выполнению 2).

7) Если $P > U$:

i) уменьшают L и K на 1;

ii) переходят к выполнению 5).

8) Получена случайная выборка из элементов, номера которых в массиве A расположены в порядке возрастания.

b) Метод 2 (адаптированный метод из [3]).

1) Пусть $C(a,b)$ — функция, равная числу сочетаний из a по b без повторений (также известная как биномиальный коэффициент и равная $a!/[(a - b)! \cdot b!]$).

2) Генерируют случайное целое число L из интервала от 1 до $C(N, n)$ включительно.

3) Создают массив $A[1{:}n]$.

4) Устанавливают $K = 0$, $J = 1$ и $m = (n - 1)$.

5) Устанавливают $A[J] = 0$.
6) Если $J \neq 1$, устанавливают $A[J] = A\ [J - 1]$.
7) Устанавливают $A[J] = A[J] + 1$.
8) Устанавливают $R = C(N - A[J], n - J)$.
9) Увеличивают K на R.
10) Если $K < L$, переходят к выполнению 7).
11) Уменьшают K на R.
12) Увеличивают J на 1.
13) Если $J \leq N$, переходят к выполнению 5).
14) Устанавливают $A[n] = A[m] + L - K$.
15) Получена случайная выборка, расположенная в порядке возрастания номеров массива A.

Примечание — Вследствие ограничений компьютерного представления больших целых чисел, а также применения генераторов случайных чисел, необходимо проверить, что метод 2 в вычислительном отношении выполним, а генерируемые случайные числа не имеют чрезмерного смещения.

Пример — Необходимо получить упорядоченную случайную выборку объема 5 единиц из партии в 25 единиц. Существует 25!/(20!5!) = 53130 возможных комбинаций по 5 единиц из 25 единиц, что выполнимо при использовании современных компьютеров. Для этих целей выбран генератор случайных чисел, описанный в разделе 7, его максимальное значение 2 147 483 562 более чем в 40 419 раз больше, чем максимальное из требуемых чисел, таким образом, смещение метода является незначительным. Генерировано единственное случайное целое число (7319) из интервала от 1 до 53130. В результате получена выборка {1, 7, 13, 18, 19}.

8.11 Кластерный отбор

Для совокупности или партии, состоящей из кластеров единиц, составляют перечень кластеров и отбирают случайную выборку, используя методы, приведенные в 8.3 или 8.6, если необходим отбор без возвращения. Полученная выборка включает единицы всех кластеров.

8.12 Случайный отбор выборки из набора целых чисел с вероятностью отбора числа в выборку пропорциональной его значению

Отбор, пропорциональный значению единиц, применяют к совокупности, состоящей из единиц, представляющих собой различные целые числа. Приведенный метод обеспечивает получение случайной выборки, отобранной таким образом, что вероятность отбора числа в выборку пропорциональна значению этого числа.

Описание алгоритмов.

a) Метод 1.
1) В списке из N единиц, расположенных в соответствии с возрастанием значений, последовательно указывают кумулятивные суммы S_i, соответствующие каждой единице.
2) Генерируют случайное целое число K из интервала от 1 до S_N включительно, где S_N — общая кумулятивная сумма совокупности.
3) Из перечня отбирают в выборку единицу, которой соответствует наибольшее значение кумулятивной суммы, не превышающее K.
4) Повторяют 2) и 3) до тех пор, пока необходимое количество n единиц выборки (с возвращением или без возвращения) не будет получено.

b) Метод 2.
1) Из перечня N различных значений единиц совокупности определяют максимальное значение M.
2) Генерируют два случайных целых числа (K, L), K из интервала от 1 до N включительно, а L из интервала от 1 до M включительно.
3) Если значение K-й единицы не превышает L, выбирают K-ую единицу в качестве выборочной единицы.
4) Повторяют 2) и 3) до тех пор, пока не будет получено необходимое количество n выборочных единиц (с возвращением или без возвращения).

Пример — Маркетинговая компания намерена провести отбор домашних хозяйств с выбором, пропорциональным количеству членов семьи в хозяйстве. Число членов семьи десяти домашних хозяйств в порядке возрастания: 2, 2, 3, 3, 3, 4, 4, 5, 6, 7. Соответствующее кумулятивное число членов семьи: 2, 4, 7,

10, 13, 17, 21, 26, 32, 39. Необходимо получить случайную выборку из 4 домашних хозяйств без возвращения. Генерировано 4 случайных целых числа от 1 до 39, {7, 33, 2, 11}. Соответствующие единицы выборки — это хозяйства, указанные в списке под номерами {3, 9, 1, 4}.

8.13 Многостадийный отбор

Для совокупности или партии с иерархической структурой выбирают случайную выборку из бóльших групп, затем делают подвыборку меньших групп из каждой ранее отобранной группы, продолжая эту процедуру до тех пор, пока не будет достигнут уровень единицы. На каждой стадии используют методы случайного отбора выборки, приведенные в 8.3 или 8.6, если необходима выборка без возвращения, или метод, приведенный в 8.5, если необходим отбор с возвращением. Количество единиц в итоговой выборке равно произведению объемов выборки на каждой стадии.

Пример — Партия состоит из 20 поддонов с 20 ящиками на каждом поддоне. Каждый ящик содержит 10 единиц продукции. Покупатель намерен проверить продукцию, используя стратегию многостадийного отбора. Отобрана случайная выборка из 4 поддонов. С каждого отобранного поддона отобрана случайная выборка из 4-х ящиков. Наконец, из каждого отобранного ящика отобрана случайная выборка из 3 единиц. Эта процедура обеспечивает получение выборки из 48 единиц продукции из партии в 4000 единиц.

8.14 Рандомизация при планировании эксперимента

При использовании планирования экспериментов рандомизацию применяют для выполнения таких действий, как назначение экспериментальной обработки исследуемых единиц или объектов и установление порядка их оценки, включая порядок оценки при повторении эксперимента. При этом может быть использован любой из приведенных методов рандомизации.

Описание алгоритмов.

a) Метод 1.

1) Для каждого элемента из перечня N обработок или единиц (в зависимости от обстоятельств) назначают различные целые числа от 1 до N.
2) Выполняют случайную перестановку N целых чисел из N (т. е. всех).
3) Выполняют экспериментальные действия в последовательности, соответствующей порядку, определенному в 2).

b) Метод 2.

1) Генерируют N случайных действительных чисел U_i и присваивают их значения в порядке следования соответствующим элементам перечня обработок или единиц (в зависимости от обстоятельств).
2) Сортируют элементы перечня обработок или единиц в порядке возрастания соответствующих им значений U_i.
3) Выполняют экспериментальную деятельность в соответствии с полученной последовательностью (см. 2).

Пример 1 — В процессе медицинских исследований необходимо проверить воздействие нового препарата по сравнению с обычно используемым препаратом в конкретной медицинской ситуации. В клинических испытаниях участвуют двенадцать добровольцев, каждому из которых присваивают номер от 1 до 12. Исследователь планирует применить препарат А (новый препарат) к 6 субъектам, а препарат В (обычно применяемый препарат) к другим 6 субъектам. Для уменьшения смещения экспериментатор принимает решение сначала рандомизировать порядок планируемых 12 применений препаратов, а затем распределить их между 12 добровольцами. Индексы препаратов и испытуемых субъектов рандомизируют каждый отдельно, используя метод 1. Полученные рандомизированные перечни препаратов {В, В, А, В, А, А, В, А, А, В, В, А} и испытуемых {3, 7, 12, 5, 1, 9, 11, 4, 10, 2, 8, 6}. Препараты теперь могут быть назначены соответствующим испытуемым на основе перечня {В3, В7, А12, В5, А1, А9, В11, А4, А10, В2, В8, А6}. Следовательно, препарат А должен быть применен для субъектов {1, 4, 6, 9, 10, 12}, а препарат В — для субъектов {2, 3, 5, 7, 8, 11}.

Пример 2 — Экспериментатору необходимо провести повторный эксперимент, проверяя каждую из 5 единиц три раза, в случайном порядке. Случайная перестановка набора {1, 2, 3, 4, 5} дала три набора: {2, 1, 5, 4, 3}, {1, 5, 2, 3, 4}, и {4, 3, 5, 2, 1}.Этим трем наборам присваивают номера (1, 2 и 3) соответственно, а экспериментатор выполняет проверки единиц в соответствии с номером проверки и в порядке, соответствующем набору с этим номером.

8.15 Случайный латинский квадрат

Латинский квадрат порядка n представляет собой матрицу $n \times n$, содержащую символы некоторого алфавита, расположенные так, что каждый символ появляется в каждой строке и в каждом столбце только

один раз. Латинский квадрат используют при планировании экспериментов. Приведенный алгоритм позволяет получить случайный латинский квадрат порядка n (см. [4]).

Описание алгоритма.

a) Создают массивы $A[1{:}n, 1{:}n]$ и $C[1{:}n]$.

b) Устанавливают $R = 1$.

c) Вводят целые числа от 1 до n в массив $C[1{:}n]$.

d) Устанавливают $J = N$.

e) Устанавливают $C = 1$.

f) Устанавливают $I = 0$.

g) Генерируют случайное целое число X в интервале от 1 до J включительно.

h) Устанавливают $H = 1$.

i) Если $I > 50$, переходят к выполнению c).

j) Если $A[H, C] = C[X]$, увеличивают значение I на 1 и переходят к выполнению g).

k) Увеличивают значение H на 1.

l) Если $H \le (R - 1)$, переходят к выполнению i).

m) Устанавливают $A[R, C] = C[X]$ и уменьшают значение J на 1.

n) Если $X > J$, переходят к выполнению r).

o) Устанавливают $K = X$.

p) Устанавливают $C[K] = C[K + 1]$.

q) Увеличивают K на 1; если $K \le J$, переходят к выполнению p).

r) Увеличивают C на 1; если $C \le n$, переходят к выполнению f).

s) Увеличивают R на 1; если $R \le n$, переходят к выполнению c).

t) В массиве A получают случайный латинский квадрат.

Примечание — Алгоритм генерации равномерно распределенных случайных латинских квадратов приведен в [5]. В дополнение необходимо отметить, что имеется связь между случайным латинским квадратом и генерацией случайных перестановок отдельных строк и колонок латинского квадрата и перестановок всех предыдущих строк и колонок.

Приложение А
(обязательное)

Таблицы случайных чисел

А.1 Описание

В данном приложении приведены две таблицы случайных чисел, которые могут быть использованы в тех случаях, когда невозможно применение компьютерных алгоритмов генерации случайных чисел. В каждой таблице приведено 3600 случайных чисел от 0 до 9, каждое из которых появляется с равной частотой. Каждая таблица состоит из 60 строк по 60 колонок в строке, что облегчает использование текущего времени суток для установки начальной точки. Таблицы получены с использованием алгоритмов, описанных в разделе 7.

А.2 Использование

А.2.1 Количество знаков и интерпретация

А.2.1.1 Определяют количество знаков (цифр числа) m, необходимых для представления объема партии N. Количество знаков равно количеству знаков в объеме партии, за исключением ситуации, когда объем партии является степенью 10. В этом случае число, состоящее из нулей, количество которых равно количеству нулей в объеме партии, следует интерпретировать как число, равное объему партии (например, если $N = 100$, значение 00, при появлении, интерпретируют как 100).

А.2.1.2 Если объем партии меньше или равен половине 10^m, начальное значение для использования таблицы может быть определено на основе трансформации наблюдаемого значения в интервале от 1 до N при условии, что в процесс не введено смещение. Этого легко достичь, отбрасывая все значения, превышающие kN, где $k = \lfloor 10^m/N \rfloor$, до трансформации в соответствии с уравнением $v_2 = 1 + (v_1 - 1) \bmod N$, где v_1 — начальное значение, v_2 — значение из заданного интервала.

А.2.2 Начальная точка

А.2.2.1 До применения таблиц случайных чисел необходимо определить стратегию выбора начальной точки. Таблицы разработаны так, что позволяют использовать для выбора начальной точки показания часов, в том числе часов, показывающих время с точностью до секунды. Приведенный алгоритм реализует применение метода основанного на использовании показаний часов для выбора начальной точки.

Описание алгоритма:

a) Делают запись текущего времени в формате «час: мин: сек».

b) Используют значение секунд для определения номера строки, интерпретируя 00 как 60.

c) Используют значение минут для определения номера колонки, интерпретируя 00 как 60.

d) Используют показания часов для определения номера таблицы (А.1 или А.2) в зависимости от четного или нечетного часа.

Пример — Экспериментатору необходимо выбрать начальную точку в таблицах для отбора случайной выборки из партии в 100 единиц. Текущее время 10:35:13. Таким образом, начальная точка находится в таблице А.2. (т. к. 10 — четное число) в 13-й строке, 35-й колонке. Здесь расположена цифра 6, но т. к. для отбора выборки из 100 единиц необходимы две цифры, использована цифра из колонки 36, таким образом, первая единица выборки имеет номер 66.

А.2.2.2 Может быть использован любой другой метод, обеспечивающий получение случайных равномерно распределенных целых чисел из интервала от 1 до 60 включительно, такой, как метод монеты или кубика (см. 5.2) или компьютерный алгоритм (см. раздел 7), позволяющий получить длинный список случайных координат начального числа, которые могут быть использованы в последующих случаях. Кроме того, в конце отбора выборки из таблицы в конкретном случае координаты начальной точки могут быть зарегистрированы и затем использованы как начальная точка для следующего отбора выборки.

Пример — В качестве продолжения примера А.2.2.1, предположим, что необходим случайный отбор 10 единиц. Получена выборка {66, 13, 10, 45, 32, 22, 41, 49, 22, 99}. Координаты следующего значения — строка 23 и колонка 35 (число 10). Эти значения могут быть зарегистрированы и использованы как начальная точка для следующего отбора выборки при необходимости.

А.2.2.3 Координаты исходной точки также могут быть определены путем установления начальной цифры на основе номеров ряда и колонки в соответствии с А.2.2.1 или А.2.2.2 и дополнительных цифр на основе номера текущей строки и дополнительных номеров колонок, основанных на случайных числах, генерированных без возвращения из интервала от 1 до 60 включительно для оставшихся ($m - 1$)-го знаков. Полученное многозначное число должно быть сформировано в том же порядке, в каком были генерированы номера колонок.

Пример — Необходима случайная выборка из партии в 1000 единиц. Начальные строка и колонка определены как 5 и 11 соответственно в таблице А.1. Необходимы еще две цифры, и они были сгенерированы как 1 и 30. В результате получены следующие координаты первой, второй и третьей цифр соот-

ветственно: (5, 11), (5, 1) и (5, 30). Принято решение о перемещении сверху вниз. Поэтому первое число 511, следующие 943, 419, 413, 899, 209 и т. д.

А.2.3 Действия на границах таблицы

А.2.3.1 При формировании числа, составленного из *m* цифр, цифры необходимо брать справа от первой цифры из колонки, номер которой не больше 60. Если *m* цифр не набрано, а номер колонки более 60, следующую цифру необходимо брать из колонки с номерами на 60 меньше, т. е. с номером 1, 2, и так далее соответственно. Это правило можно применять в пределах существующей таблицы или переходить в другую таблицу случайных чисел, рассматривая ее как расширение первой таблицы.

А.2.3.2 Обычно применяют правило, состоящее в том, чтобы получать последующие случайные числа, увеличивая номер строки на единицу и читая *m* цифр числа в соответствии с номерами колонок и их порядком. В тех случаях, когда номер строки превышает 60, ряд с номером 1 используют вместо ряда 61 и увеличивают номер колонки на *m* в случае последовательно используемых колонок, или на один в случае, когда применяют А.2.2.3 для определения колонки и продолжения определения чисел. Это правило может быть применено в пределах существующей таблицы или при использовании другой таблицы случайных чисел в качестве расширения первой таблицы, если А.2.3.1 не использовано.

А.2.4 Отчет для аудита

Если отчеты необходимы для целей аудита или иных проверок, выполняемых ответственными лицами или третьей стороной, в отчете записывают объем партии и объем выборки.

Кроме того, относительно таблиц и их использования записывают:

a)номер начальной строки;

b)номер колонки(ок) и их последовательность;

c)направление, выбранное в таблице;

d)начальная использованная таблица и способ ее расширения с помощью другой таблицы, если это применялось;

e)использованную трансформацию.

Таблица А.1 — Случайные числа (для нечетных значений)

Номер строки *i*	Номер колонки *j*											
	5	10	15	20	25	30	35	40	45	50	55	60
	95183	14683	96585	84761	65044	65183	55567	28734	19802	56410	79127	02879
	08509	97009	47525	88791	93751	70490	17749	32927	65085	94970	55541	89466
	45448	66819	86936	95349	08657	75106	97487	85268	59208	43206	14898	29083
	02230	00022	46390	76658	91934	64676	42429	96812	30560	99913	72809	66736
5	13275	96798	51425	67147	15216	71831	16229	25862	22090	91420	24352	03550
	44439	33385	95151	92374	14683	00323	57667	78341	09004	80139	81182	87552
	17629	80967	42144	58190	24550	62189	94525	44967	15860	85739	93323	87043
	14328	77127	40397	78105	75031	99553	84296	01482	25738	32761	85035	68873
	96896	02466	86706	09507	66840	68509	38033	90785	75831	98886	00905	48343
10	09725	80938	27971	01243	29232	28799	88456	99618	20071	79865	63584	69087
	55021	37184	69480	56317	19944	56756	37514	86439	69831	15172	81398	69574
	06492	95014	54908	21591	13771	35967	78637	29918	47923	61404	63378	72394
	20604	54145	27781	35157	50127	61025	57344	36615	07766	83959	34546	67011
	20202	58870	67569	71756	76284	30909	87763	21951	67756	82597	15210	04291
15	27160	01595	64831	07126	25821	81524	12585	76273	36256	41879	33287	84361
	95089	78572	87167	65888	93358	23879	84496	16147	31130	96978	80361	85195
	74825	21529	24660	33314	64512	80550	51712	23057	53841	32470	36790	60455
	80338	94074	65731	39470	03807	72355	40407	86049	81583	06786	16673	06017
	16596	43179	42026	94264	28301	29514	60657	21732	21548	28693	15241	68944
20	34134	42056	40153	00994	14179	44447	99399	86963	71862	01306	15489	00515
	01118	98623	33695	49221	97197	21424	91691	09365	62483	98893	22106	45399
	67371	71659	30505	71239	56944	35898	02207	93274	40142	98319	41218	43739
	03485	55173	68477	12348	76971	64800	86498	42059	08942	32931	73896	27772
	33328	74045	25331	37635	39081	28786	20843	32565	24316	17888	47626	69199
25	84302	10060	25334	84920	30270	09722	61706	52863	03417	95658	74490	00143
	94775	52191	94552	99265	55079	64517	16803	13037	50984	14886	04385	67907
	51700	63604	96771	34444	30002	67975	93167	16746	97842	25589	12568	81785
	75920	13260	44283	27735	31134	97100	36706	24404	56970	44575	68832	42374
	32385	28423	46784	59222	17776	57726	56449	32109	11825	57995	91217	12802
30	13424	00587	12231	44543	62984	58391	22054	16134	73790	59050	24893	62342

Окончание таблицы А.1

Номер строки i	Номер колонки j											
	5	10	15	20	25	30	35	40	45	50	55	60
	90896	00608	31377	53338	84813	76825	92192	24937	81481	01866	22641	21817
	60682	38700	34039	93512	38596	40004	71447	97193	52407	44146	77116	99965
	38746	34667	84499	70915	91391	25660	12328	35273	08135	04799	14489	19984
	51658	18422	05732	91001	98070	13591	44468	88460	66964	24038	93987	66335
35	49174	12449	97583	85835	82313	96349	92721	64617	06030	22312	94263	80291
	44215	21953	21844	44114	93162	51028	29551	66121	63959	97789	44259	90865
	33877	94654	10025	84935	94630	49660	23473	89644	67212	75851	83767	45647
	83411	43288	47832	40488	89085	69731	00790	60182	71358	22571	94204	64211
	27135	82404	52031	44648	97600	72166	70830	27701	01755	00523	01837	31304
40	29475	31431	46863	70098	98659	72035	21538	12923	76963	78288	59083	18839
	56886	38711	66126	16504	87900	74055	46028	84821	83323	35962	27522	87875
	55061	35916	15955	28228	36994	73167	17137	36572	48592	60721	97714	61215
	01646	62126	37253	24997	53016	96515	40536	39311	64151	93960	24053	87645
	41789	28167	90577	84499	25059	90583	09422	87357	55416	81135	41286	92320
45	26066	80119	44259	94514	21211	44302	29023	28138	03693	50650	38450	61118
	63559	20927	12881	25582	07872	28073	59006	55666	68690	59772	25162	87924
	35054	84077	02504	10800	75293	86466	92406	56289	79807	55271	73177	70568
	32826	26937	75563	14290	30078	70820	58639	64900	61699	34974	11738	64065
	07860	90064	91220	46786	45994	47375	70140	35592	05990	58470	82014	05265
50	79276	72512	19525	27397	88975	77137	40032	06205	06997	53504	07760	62546
	41093	04332	68677	27073	94104	58532	53616	32156	66153	00264	36374	15230
	27382	27938	91695	64013	46719	61629	33668	32391	35411	68209	33885	64050
	38984	47230	59448	97802	37987	22733	52199	12325	18625	01271	84870	10911
	51411	44221	93363	48654	42656	99464	08481	98128	66677	89441	66019	75095
55	80310	18848	23722	30788	27435	03780	85737	05561	57203	07316	98597	73621
	40082	39571	89790	65382	01447	15984	60854	72833	87320	95245	40678	89785
	86922	84354	81939	32180	32891	52704	84659	95442	86204	44040	51613	15984
	07932	69932	18796	87070	82202	05372	93506	60697	48535	89027	45719	51567
	20383	07288	50265	48321	63056	35861	80864	86357	51567	68151	11723	06990
60	45471	31340	30187	23899	36361	96780	55823	37743	06957	77884	78061	36603

Таблица А.2 — Случайные числа (для четных значений)

Номер строки i	Номер колонки j											
	5	10	15	20	25	30	35	40	45	50	55	60
	51326	91644	88971	00664	10776	90888	97107	00930	87438	23714	33246	72109
	96821	43647	41707	91062	20037	47660	34652	82087	29652	51614	11015	66187
	67104	67934	95662	85259	08546	77032	28958	26815	66683	50539	65189	22993
	93315	93110	01022	16079	74364	97582	34309	18699	24437	12839	48005	46478
5	45823	44862	86472	77354	95916	36599	52350	89866	98532	69704	33735	13696
	32573	58513	70797	53560	23115	38325	74380	99917	21721	00323	29402	27080
	74553	41700	25357	17428	66708	83630	42360	16842	40782	21345	88668	95845
	06794	48960	75160	18552	00424	25976	69852	35837	69810	64014	03045	53378
	75606	34848	74458	71100	01512	89662	01391	49140	18048	59282	77344	72419
10	97535	55169	47044	81940	16507	12220	16375	65306	38691	75019	55186	47269
	89809	48811	09494	06576	58258	73155	46712	11106	38382	70893	39503	42937
	88095	48475	07399	02165	98038	97247	48441	93066	20276	72169	49706	56553
	48854	75803	34089	45335	13056	44509	77886	65036	42619	95564	66315	49084
	97951	73015	36585	89007	12069	96858	17241	37189	44193	28270	81624	75256
15	48953	47945	81153	10121	69935	60894	23151	00113	40597	71164	61331	10930
	83334	44992	30210	99370	60425	87391	63244	54828	12155	64082	43773	66172
	31105	74546	74875	51747	41021	22338	49403	20173	75454	62122	37227	35790
	94245	19396	03648	28548	42987	78668	09292	23073	21116	07199	21398	18770
	19221	70541	58195	21383	57877	27812	06504	18289	18606	17680	09218	58984
20	33503	45776	88514	10094	28589	47548	64714	96174	10026	87111	29333	77885

Окончание таблицы А.2

Номер строки *i*	Номер колонки *j*											
	5	10	15	20	25	30	35	40	45	50	55	60
	03602	99374	41918	43875	11258	83646	46042	26986	11003	94756	89972	00805
	43277	65791	12217	23767	09833	66504	82359	95754	40249	71472	61588	04428
	31966	42142	71410	28139	60147	68496	89021	01615	36565	85598	18048	32584
	19193	31952	87947	89521	65225	97987	14794	90695	69314	10359	27881	38183
25	98935	48606	27475	73462	66692	11151	21709	32836	92997	33682	12722	38906
	21880	10506	78478	85067	30375	82944	06660	40750	84252	20463	37184	27248
	81695	06183	08147	01241	16278	54886	36468	21315	49106	10291	16837	40481
	68883	60266	66180	05680	38799	40481	73524	73255	79950	42007	56334	54332
	59547	60741	56065	71467	07193	38784	08169	07389	64049	21355	86589	06583
30	91709	73154	86898	20234	05773	47157	72305	20819	57301	89018	74851	50560
	57195	97283	25156	59277	33608	73937	19341	17262	63955	41678	36229	54204
	03657	71909	82018	83110	21722	03455	30654	57890	18530	60458	57145	08764
	93373	88795	11353	44726	66989	24389	93445	53752	98703	55276	18391	53513
	35158	50868	45055	12180	29993	69555	69613	69358	96861	01667	47738	11964
35	38056	14298	10431	53147	76843	32128	46844	23407	62423	01712	46033	64425
	49152	05010	84942	25483	52825	17485	67614	12493	88626	39589	56044	21968
	80000	90734	70131	19986	34949	76990	48325	39323	66921	89134	21853	18973
	53597	22379	94302	15425	62185	27894	37281	38876	97902	34008	45051	05607
	09151	34061	64751	96631	50373	61603	84917	56084	57647	80898	55489	24602
40	99734	68144	63963	73011	22832	98145	31523	60195	34172	40637	60940	51237
	84547	89655	53120	95599	04602	07968	85748	74914	76227	07158	24432	22963
	18815	26665	25301	67754	88457	19913	96787	71084	14867	03077	89575	66834
	14169	38336	41192	56208	29069	87045	32135	25975	71643	74200	52556	30213
	90528	60501	73201	72999	30355	86428	39401	72077	48056	17853	24894	19838
45	99055	42696	14376	24907	06082	61789	03963	64664	09132	87218	64755	46107
	62530	10183	38149	70004	74983	02092	40704	01062	17000	61170	99026	24025
	74196	77214	89483	43933	80953	81268	46485	23647	98173	55947	96727	86378
	27293	56047	73998	19996	94427	09157	62999	88803	81272	22315	92708	07343
	94220	93209	32369	82003	82433	85790	47632	36285	68771	06006	37556	51601
50	68430	23169	58879	97812	39399	71469	40835	04924	30336	59222	06350	45656
	61949	23031	50698	85772	85990	36942	11098	06636	57547	73247	46229	52551
	57248	90383	23502	22642	80722	38164	12160	51707	22075	20624	91644	08780
	10777	53979	65288	39116	80635	49653	36903	33854	79873	67823	23256	31643
	06717	92287	42775	79274	90874	44006	27312	15909	25276	59863	75607	22277
55	09519	67689	13829	30992	44921	67375	94754	95322	25501	78486	99059	62524
	57335	48704	79426	49770	32989	22640	88230	66598	27685	29719	99930	26181
	14911	08271	21662	40886	53783	76430	41233	44057	28385	21751	51476	64387
	04837	08929	81607	33210	61894	17240	37617	56753	61251	49433	65644	63758
	14430	20139	15027	52208	16440	59911	57566	22227	60109	95260	21388	96686
60	68896	64599	91227	55882	60220	70202	73354	34776	55530	20599	45720	75145

Приложение В
(справочное)

Текст программы генерации случайных чисел

В.1 Краткое описание

Приведенные программы написаны на языке программирования Си (см. [6]) и соответствуют алгоритмам, описанным в разделе 7. Ссылки на соответствующие подразделы раздела 7 включены в качестве комментариев в текст программ, что позволяет составить программу на другом языке программирования.

В.2 Демонстрационная программа

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
/* Прототипы функций */
int SeedGen(void);
double U(void);
/* Глобальные переменные */
long ij, Seed, Seed2, S2k;
char str[20];
const long m1=2147483563, m2=2147483399, mm1=2147483562,
      a1=40014, a2=40692, q1=53668, q2=52774, r1=12211, r2=3791;
const double ufac=4,6566130573917691e-10;
/* Демонстрационная программа */
int main(void)
{
   long i, nn, n, Seed1, *A, *ptr;
   char yn[4];
   time_t tnow;

   system("CLS");
   printf("Демонстрационная программа для ISO 24153: -\n");
   printf("Процедуры рандомизации и отбора случайной выборки (Раздел 7)\n");
   printf("\n Алгоритмы генерации случайного числа и начального числа, \n");
   printf("используемые для отбора единственной выборки \n");

   printf("Объем партии:          ");
   scanf("%d", &nn);
   printf("Объем выборки:         ");
   scanf("%d", &n);
   printf("\n Ручной ввод (Y/N):   ");
   scanf("%s", yn);
   if (yn[0] == 'Y' || yn[0] == 'y') {                    /* ручной ввод начального числа                    */
      printf("\n Введите целое из интервала от 1 до 2147483398 включительно: ");
      scanf("%d", &Seed);
      Seed1 = Seed;                                       /* сохранение начального числа, см. 7.1.3          */
      S2k = 0;
      tnow = time(NULL);
      strftime(str, 20, "%Y-%m-%d %H:%M:%S", localtime(&tnow));
      }
   else {
   Seed = SeedGen();                                      /* функция вызова начального числа, см. 7.2        */
   Seed1 = Seed;                                          /* сохранение начального числа, см. 7.2.5          */
   }
   Seed2 = Seed;                                          /* RNG Параметры функции начального числа          */
   ij = -1;                                               /* RNG Параметр функции инициализации              */
   /* создание массива для выборочных значений */
   A = (long *) calloc(n, sizeof(long));
```

```
    if (A == NULL) {
        printf("Массив ошибок размещения\n");
        exit(1);
    }
    /* Отбор случайной выборки с возвращением */
    ptr = A;
    for (i = 0; i < n; i++)
        *(ptr++) = 1 + (long)floor(U() * nn);          /*масштабированный выход (1;nn) см. 7.3.9          */

    /* выход с проверкой деталей, см.7.4 */
    printf("\n Объем партии:                %d\n", nn);
    printf("Объем выборки:                  %d\n", n);
    printf("Дата и время:                   %s\n", str);
    printf("Количество прошедших секунд:    %d\n", S2k);
    printf("начальное число:                %d\n", Seed1);
    printf("отбор выборки: \n");
    for (i = 0; i < n; i++) printf("%8d", A[i]);
    printf("\n");
    system("Пауза");
    return 0;
}
```

В.3 Генератор случайного начального числа

Если требуется ручной ввод начального числа (для проверки, аудита и т. п.), данную функцию вызывают в соответствии с 7.1.3.

```
int   SeedGen(void)
{
/* В.1.3 Функция генерации случайного начального числа                */
/* Установление даты и времени компьютерной системы см. 7.2.2 a)  */
   long i, j, k;
   struct tm t, *ptr;
   time_t tnow, tref;

   /*Время отсчета: 2000-01-01 00:00:00 */
   t.tm_year = 2000 - 1900;
   t.tm_mon = 0; t.tm_mday = 1; t.tm_hour = 0;
   t.tm_min = 0; t.tm_sec = 0; t.tm_isdst = 0;

   tref = mktime(&t);
   tnow = time(NULL);
   ptr = localtime(&tnow);
   strftime(str, 20, "%Y-%m-%d %H:%M:%S", ptr);

/* количество дней до компьютерной даты с 2000-01-01 00:00:00 см. 7.2.2 b)     */
/* количество секунд с 2000-01-01 00:00:00 см. 7.2.2 c)                        */
   S2k = (long)difftime(tnow, tref);

/* генератор квазислучайного начального числа                                  */
   Seed = S2k;                                  /* инициация начального числа   см. 7.2.2 d)        */
   j = S2k - (S2k / 100) * 100 + 1;             /* пробное значение   см. 7.2.2 e)                  */
   for (i = 1; i <= j; i++) {                   /* см.7.2.4                                         */
       k = Seed / q2;                           /* RNG #2  см. 7.2.3 b)                             */
       Seed = a2 * (Seed - k * q2) - k * r2;    /* RNG #2  см. 7.2.3 c)                             */
       if (Seed < 0) Seed += m2;                /*см. 7.2.3 d)                                      */
   }
   return Seed;                                 /* начальное число см. 7.2.5                        */
/* функция дополнительно модифицирует глобальные переменные str[] и S2k */
}
```

В.4 Функция генерации случайных чисел

Если необходим ручной ввод начального числа (для проверки, аудита и т. п.) (см. 7.1.3), данную функцию не вызывают.

```
double U(void)
{
/*  B.1.4 функция генерации случайного числа */
  int j, k, i1;
  static long k1, Shuffle[32];

  if (ij < 0) {                          /* создание массива в соответствии с 7.3.1 a)                    */
    for (j = 39; j >= 0; j- -) {         /*заполнение массива перестановок в соответствии с 7.3.4         */
      k = Seed / q1;                     /* RNG #1  см. 7.3.3 b)                                          */
      Seed = a1 * (Seed - k * q1) - k * r1;  /* RNG #1  см. 7.3.3 c)                                      */
      if (Seed < 0) Seed += m1;          /* RNG #1  см.7.3.3 d)                                           */
      if (j <= 31) Shuffle[j] = Seed;    /*  см. 7.3.4                                                    */
    }                                    /* последнее значение начального числа, вход в 7.3.6 a)          */
    ij = 0;                              /* дальнейшая инициализация невозможна                           */
    k1 = Shuffle[0];                     /* вход в 7.3.6 c)                                               */
  }
    /* генератор комбинированного случайного числа (CRNG) (см. 7.3.6) */
    k = Seed / q1;                       /* RNG #1; вход из 7.3.4 см. 7.3.6 a)                            */
    Seed = a1 * (Seed - k * q1) - k * r1;  /* RNG #1 см. 7.3.6 a)                                         */
    if (Seed < 0) Seed += m1;            /* RNG #1 см. 7.3.6 a)                                           */
    k = Seed2 / q2;                      /* RNG #2; вход из 7.2.5 см. 7.3.6 b)                            */
    Seed2 = a2 * (Seed2 - k * q2) - k * r2;  /* RNG #1 см. 7.3.6 b)                                       */
    if (Seed2 < 0) Seed2 += m2;          /* RNG #1 см. 7.3.6 b)                                           */
    i1 = floor(32.0 * k1 / m1);          /* вычисление индекса массива см. 7.3.6 c)                       */
    k1 = Shuffle[i1] - Seed2;            /* нешкалированные выходные данные см. 7.3.6 d)                  */
    Shuffle[i1] = Seed;                  /* измененный массив перестановок см. 7.3.6 e)                   */
    if (k1 < 1) k1 += mm1;               /* нешкалированные выходные данные см. 7.3.6 f)                  */
    return (k1*ufac);                    /* действительное число из интервала (0;1) см. 7.3.8             */
  }
```

Приложение С
(справочное)

Текст программ для случайного отбора выборки и рандомизации

С.1 Введение

В данном приложении приведен текст программ на языке Си (см. [6], для отбора выборки (раздел 8). Приведенный текст программ не оптимален по структуре и приведен для иллюстрации.

С.2 Текст демонстрационной программы

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* Прототипы функций */
void    RandPermNM(long m, long n, long A[]);
void    RandPermN(long n, long A[]);
void    RandDerang(long n, long A[], long B[]);
void    RSWOR(long n, long m, long A[]);
void    SRSWORIULS(long n, long m, long B[], long A[]);
void    OSRSWORA(long n, long m, long A[]);
void    OSRSWORB(long n, long m, long l, long A[]);
void    RLS(long n, long A[][]);
long    RandIntMN(long m, long n);
long    RandInt1N(long n);
long    C(long m, long n);
double U(void);

/*  Глобальные переменные   */
long  Seed;
const long ka = 16807, kq = 127773, kr = 2836, km = 2147483647;

/* Демонстрационная программа */
int  main(void)
{
  long  i, j, m, n;
  system("CLS");
  printf("Демонстрационная программа. Раздел 8.\n\n");

  Seed = 543210; m = 1; n = 100;
  printf("Случайное целое из интервала от m до n (см. 8.2)\n");
  printf("m, n, i:  %d  %d  %d\n", m, n, RandIntMN(m, n));
  system("Пауза "); printf("\n");

  Seed = 543210; n = 100;
  printf("Случайное целое от 1 до n, см. 8.2. \n");
  printf("n, i:  %d  %d\n", n, RandInt1N(n));
  system("Пауза "); printf("\n");

  Seed = 543210; m = 5; n = 10;
  long A[n];
  for (i = 0; i < n; i++) A[i] = i+1;
  printf("Случайная перестановка по m элементов одновременно, см. 8.3. \n");
  RandPermNM(m, n, A);
  printf("m, n:  %d  %d\n", m, n);
  for (i = 0; i < m; i++) printf("%d ", A[i]);
  printf("\n");
  system("Пауза "); printf("\n");
```

```
Seed = 543210; n = 10;
long A1[n];
for (i = 0; i < n; i++) A1[i] = i+1;
printf("Случайная перестановка всех n элементов, см. 8.3. \n");
RandPermN(n, A1);
printf("n: %d\n", n);
for (i = 0; i < n; i++) printf("%d ", A1[i]);
printf("\n");
system("Пауза "); printf("\n");

n = 10;
long A2[n];
printf("Случайное изменение порядка, см. 8.4 \n");
RandDerang(n, A1, A2);
for (i = 0; i < n; i++) printf("%d ", A2[i]);
printf("\n");
system("Пауза "); printf("\n");

Seed = 543210; n = 5; m = 20;
long A3[n];
printf("Случайный отбор без возвращения, см. 8.6 \n");
RSWOR(n, m, A3);
printf("m, n: %d  %d\n", m, n);
for (i = 0; i < n; i++) printf("%d ", A3[i]);
printf("\n");
system("Пауза "); printf("\n");

Seed = 543210; n = 5; m = 20;
long A4[n], A5[m + 2];
for (i = 0; i < m+2; i++) A5[i] = 0;
for (i = 0; i < m; i++) A5[i] = i+1;
printf("Единичный случайный отбор выборки из партии неизвестного объема, см. 8.9 \n");
SRSWORIULS(n, m, A5, A4);
printf("m, n: %d  %d\n", m, n);
for (i = 0; i < n; i++) printf("%d ", A4[i]);
printf("\n");
system("Пауза "); printf("\n");

Seed = 543210; n = 5; m = 20;
long A6[n];
printf("Метод A: отбор единственной упорядоченной случайной выборки без возвращения, см.8.10 \n");
OSRSWORA(n, m, A6);
printf("m, n: %d  %d\n", m, n);
for (i = 0; i < n; i++) printf("%d ", A6[i]);
printf("\n");
system("Пауза"); printf("\n");

Seed = 543210; n = 5; m = 20;
long A7[n+1], b, d;
b = C(m, n); d = RandInt1N(b);
printf("Метод B: отбор единственной упорядоченной случайной выборки без возвращения, см.8.10 \n");
OSRSWORB(n, m, d, A7);
printf("m, n, d, C(m,n): %d %d %d %d\n", m, n, d, b);
for (i = 1; i <= n; i++) printf("%d ", A7[i]);
printf("\n");
system("Пауза"); printf("\n");

Seed = 543210; n = 8;
long A8[n+1][n+1];
printf("8.15 Случайный латинский квадрат, см. 8.15 \n");
```

```
  RLS(n, A8);
  printf("Порядок: %d\n", n);
  for (i = 1; i <= n; i++) {
    for (j = 1; j <= n; j++) {
      printf("%3d ", A8[i][j]);
    }
   printf("\n");
  }
  printf("\n");
  system("Пауза");
  return 0;
}
```

С.3 Функции

```
/*************************************************************************/
long RandInt1N(n)
long n;
{
/* Случайное целое из интервала от 1 до n включительно, см. 8.2          */
  return (1 + (long)floor(U() * n));
}
/*************************************************************************/
long RandIntMN(m, n)
long m, n;
{
/* Случайное целое из интервала от m до n включительно;(m < n), см.8.2    */
  return (m + (long)floor(U() * (n - m + 1)));
}
/*************************************************************************/
void RandPermN (n, A)
long n, A[];
{
/* Случайная перестановка N элементов, см. 8.3                            */
/* A[] – массив входа и выхода; выход в A[]                               */
  long j, k, temp;
  for (j = 0; j < n-1; j++) {
    k = RandIntMN(j, n-1);
    temp = A[j]; A[j] = A[k]; A[k] = temp;
    }
  return;
}
/*************************************************************************/
void Rand Perm NM (m, n, A)
long m, n, A[];
{
/* Случайная перестановка по m единиц одновременно из n, см. 8.3          */
/* A[] – массив входа и выхода; выход в A[0:m-1]                          */
  long j, k, temp;
  if (m == n) {m = n - 1;}
  for (j = 0; j < m; j++) {
    k = RandIntMN(j, n-1);
    temp = A[j]; A[j] = A[k]; A[k] = temp;
    }
  return;
}
/*************************************************************************/
void RandDerang (n, A1, A2)
long n, A1[], A2[];
{
```

```
/* Случайное изменение порядка, см.8.4                                    */
/* A1[] — массив входа; A2[] — массив выхода                              */
  long i, iFlag, n1;
  for (i = 0; i < n; i++)
    A2[i] = A1[i];                    /* сохранение */
  iFlag = 1; n1 = n;
  for (;;) {
    RandPermN(n1, A2);
    for (i = 0; i < n1; i++) {
      if (A2[i] == A1[i]) {iFlag = 0; break;}
      }
    if (iFlag == 0) iFlag = 1;
    else break;
  }
return;
}
/*************************************************************************/
void RSWOR (n, m, A)
long n, m, A[];
{
/* Случайный отбор выборки без возвращения, см. 8.6                       */
/* n = объем выборки; m = объем партии                                    */
/* A[] — массив полученной выборки                                        */
  long i, k;
  long B[m]; /* массив для сохранения исходных значений                  */
  for (i = 0; i < m; i++) B[i] = 0;
  i = –1;
  do {
      k = RandInt1N(m);
      if (B[k] == 0) {
      B[k] = 1; i = i + 1; A[i] = k;
      }
  } while (i < n);
  return;
}
/*************************************************************************************/
void SRSWORIULS (n, m, B, A)
long n, m, B[], A[];
{
/* Единичный случайный отбор выборки из партии неизвестного объема, см. 8.9                  */
/* n = объем выборки; m = объем партии; (n < m)                                               */
/* A[] — массив выходных выборочных данных                                                    */
/* B[] — массив для моделирования партии неизвестного объема(0 указывает, что объем партии исчерпан) */
  long k, v;
  m = 0; /* счетчик объема партии */
  for (;;) {
      m = m + 1;
      v = B[m-1];
      if (v == 0) {m = m - 1; break;}
      if (m <= n) A[m-1] = v;
      else {
          k = RandInt1N(m);
          if (k <= n) A[k-1] = v;
      }
  }
  return;
}
/*************************************************************************************/
void OSRSWORA (n, m, A)
long n, m, A[];
```

```
{
/* Метод A: отбор единственной упорядоченной случайной выборки без возвращения, см. 8.10 a)   */
/* n = объем выборки; m = объем партии; (n < m)                                                */
/* A[] — массив, содержащий значения полученной выборки                                        */
  long j, k, k1;
  double p, x;
  k = m - n; k1 = m; j = 0;
OsrsworA1:
  j = j + 1;
  if (j > n) goto OsrsworA3;
  x = U(); p = 1.0;
OsrsworA2:
  p = p * k / k1;
  if (p <= x) {
    A[j-1] = m - k1 + 1; k1 = k1 - 1;
    goto OsrsworA1;
    }
  else {
      k1 = k1 - 1; k = k - 1;
      goto OsrsworA2;
      }
OsrsworA3:
  return;
}
/*****************************************************************************/
void OSRSWORB (n, m, Ix, A)
long n, m, Ix, A[];
{
/* Метод B: отбор единственной упорядоченной случайной выборки без возвращения, см. 8.10 b)   */
/* Поиск сочетаний из m по n                                                                   */
/*для данного лексикографического индекса                                                      */
/* n = объем выборки; m = объем партии; (n < m)                                               */
/* Ix = лексикографический индекс найденного сочетания [1 <= Ix <= C(m,n)]                    */
/* A[] — массив, содержащий значения полученной выборки                                       */
  long i, k, n1, r;
  k = 0; n1 = n - 1;
  for (i = 1; i < n; i++) {
      A[i] = 0;
      if (i != 1) A[i] = A[i-1];
OsrsworB1:
      A[i] = A[i] + 1;
      r = C(m - A[i], n - i);
      k = k + r;
      if (k < Ix) goto OsrsworB1;
      k = k - r;
      }
      A[n] = A[n1] + Ix - k;
      return;
}
/*****************************************************************************/
void RLS (n, A)
long n, A[n+1][n+1];
{
/* Случайный латинский квадрат, см. 8.15                                                      */
/* n = порядок; A[][] — массив, содержащий полученный случайный латинский квадрат             */
  long B[n+1], h, i, j, k, r, c, x;
  for (r = 1; r <= n; r++) {
Rls1:
      for (i = 1; i <= n; i++) B[i] = i;
      j = n;
```

```
      for (c = 1; c <= n; c++) {
        i = 0;
RIs2:
        x = floor(U() * j + 1);
        for (h = 1; h <= r-1; h++) {
          if (i > 50) goto RIs1;                    /* ряд закончен */
          if (A[h][c] == B[x]) {
          i = i + 1; goto RIs2;                     /* колонка закончена */
          }
        }
        A[r][c] = B[x]; j = j - 1;
        for (k = x; k <= j; k++) B[k] = B[k + 1];
}
  }
  return;
}

/*************************************************************************/
long C(m, n)
long m, n;
{
/* Метод B: вспомогательная функция, см. 8.10 b)                          */
/* Вычисление числа сочетаний из m по n.                                  */
  long i, k, x, n1;
  n1 = n; k = m - n1;
  if (n1 < k) {k = n1; n1 = m - k;}
  x = n1 + 1;
  if (k == 0) {x = 1;}
  if (k >= 2) {
      for (i = 2; i <= k; i++)
          x = (x * (n1 + i)) / i;
  }
  return x;
}
/*************************************************************************/
double U(void)
{
/* библиографический источник [12]                                        */
/* RNG основано на: x[i+1] = 16807 * x[i] mod 2147483647                  */
/* Приведено только для иллюстрации                                       */
  long k;
  k = Seed / kq;
  Seed = ka * (Seed - k * kq) - kr * k;
  if (Seed < 0) {Seed = Seed + km;}
  return (1.0 * Seed / km);
}
```

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО 3534-1:2006	—	*
ИСО 3534-2:2006	—	*
ИСО 3534-3:1999	IDT	Р 50.1.040—2002 «Статистические методы. Планирование экспериментов. Термины и определения»
ИСО 80000-2:2009	NEQ	ГОСТ Р 54521—2011 «Статистические методы. Математические символы и знаки для применения в стандартах»

* Соответствующий национальный стандарт отсутствует. До его утверждения рекомендуется использовать перевод на русский язык данного международного стандарта. Перевод данного международного стандарта находится в Федеральном информационном фонде технических регламентов и стандартов.

Примечание — В настоящей таблице использованы следующие условные обозначения степени соответствия стандартов:

- IDT — идентичные стандарты;
- NEQ — неэквивалентные стандарты.

Библиография

[1] BAYS, C. and DURHAM, S.D. Improving a Poor Random Number Generator. ACM Transactions on Mathematical Software, **2** (1), 1976, pp. 59—64

[2] BISSELL, A.F. Ordered random selection without replacement. *Applied Statistics*, 35, 1986, pp. 73—75

[3] BUCKLES, B.P. and LYBANON, M. Algorithm 515, Generation of a Vector from the Lexicographical Index.ACM Transactions on Mathematical Software, **3** (2), 1977, pp. 180—182

[4] BYERS, J.A. Random selection algorithms for spatial and temporal sampling. Computers in Biology and Medicine, 26, 1996, pp. 41—52

[5] JACOBSON, M.T. and MATTHEWS, P. Generating uniformly distributed random Latin squares. Journal of Combinatorial Design, 4, 1996, pp. 405—437

[6] ISO/IEC 9899:1999, Programming languages — C

[7] L'ECUYER, P. An Efficient and Portable Combined Random Number Generator. Communications of the *ACM*, 31 (6), 1988, pp. 742—749, 774

[8] MARSAGLIA, G. Random Number Generators. Journal of Modern Applied Statistical Methods, 2 (1), 2003, pp. 2—13

[9] MCCULLOUGH, B.D. Assessing the Reliability of Statistical Software: Part II. The American Statistician, Vol. 53, No. 2 (May), 1999, pp. 149-159

[10] MCCULLOUGH, B.D. and WILSON, B. On the Accuracy of Statistical Procedures in Microsoft EXCEL 97. Computational Statistics and Data Analysis, **31** (1), 1999, pp. 27—37

[11] MCLEOD, A.I. and BELLHOUSE, D.R. A convenient algorithm for drawing a simple random sample. Applied Statistics, **32,** 1983, pp. 182—184

[12] PARK, S.K. and MILLER, K.W. Random Number Generators: Good Ones are Hard to Find. Communications of the ACM, **31** (10), 1988, pp. 1192—1201

[13] PRESS, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P. Numerical Recipes in Fortran 77: The Art of Scientific Computing, Second Edition (Volume 1 of Fortran Numerical Recipes), Cambridge University Press, Cambridge, UK, 1992, 2001

[14] SOM, R.K. A Manual of Sampling Techniques. Heinemann Educational Books Ltd., London, 1973

УДК 658.562.012.7:65.012.122:006.352 ОКС 03.120.30 Т59

Ключевые слова: совокупность, случайная выборка, случайный отбор выборки, рандомизация, отбор без возвращения, начальное число, расслоенная выборка, слой, кластер, кластерный отбор, многостадийный отбор

Редактор *С. Д. Золотова*
Технический редактор *В. Н. Прусакова*
Корректор *С. В. Смирнова*
Компьютерная верстка *З. И. Мартыновой*

Сдано в набор 27.03.2014. Подписано в печать 03.06.2014. Формат 60×84 1/8. Бумага офсетная. Гарнитура Ариал. Печать офсетная. Усл. печ. л. 4,18. Уч.-изд. л. 3,45. Тираж 62 экз. Зак. 579.

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
gost.e-ecolog.ru gost.e-ecolog.ru
Набрано и отпечатано в Калужской типографии стандартов, 248021 Калуга, ул. Московская, 256.

gost.e-ecolog.ru