Д. М. Ушаков

ИНФОРМАТИКА

НОВЫЙ ПОЛНЫЙ

СПРАВОЧНИК

ДЛЯ ПОДГОТОВКИ

К ОГЭ

Москва
Издательство АСТ
2025

УДК 373:002
ББК 32.81я721
У93

Ушаков, Денис Михайлович.

У93 Информатика : Новый полный справочник для подготовки к ОГЭ / Д.М. Ушаков. — Москва: Издательство АСТ, 2025. — 413, [3] с.: ил. — (Самый популярный справочник для подготовки к ОГЭ).

ISBN 978-5-17-164880-0

В справочнике представлен материал курса информатики в объёме, проверяемом на государственной итоговой аттестации.

Структура справочника соответствует официальным требованиям к контрольным измерительным материалам (КИМ) для проведения ОГЭ. Материал сгруппирован по главам, в каждой из которых изучается определённая тема курса информатики и ИКТ, проверяемая на экзамене. Помимо теоретических сведений главы включают разбор экзаменационных заданий и задачи для самостоятельной тренировки с ответами и критериями оценки в конце пособия.

Книга будет незаменимым помощником при подготовке к экзамену в формате ОГЭ, при изучении нового материала, повторении пройденных тем.

УДК 373:002
ББК 32.81я721

ISBN 978-5-17-164880-0

Содержание

Предисловие

Данное учебное пособие предназначено для подготовки учащихся к сдаче основного государственного экзамена по информатике.

Предлагаемый материал справочника позволит девятиклассникам самостоятельно проверить свои знания и готовность к выполнению экзаменационной работы, а преподавателям организовать успешную подготовку к итоговой аттестации.

Пособие написано на основе большого педагогического опыта автора по подготовке учащихся к экзаменам по информатике (ОГЭ и ЕГЭ).

Структура справочника соответствует официальным требованиям к контрольным измерительным материалам (КИМ) для проведения ОГЭ. Материал сгруппирован по главам, в каждой из которых изучается определённая тема курса информатики и ИКТ, проверяемая на экзамене. Главы включают в себя:

- теоретический материал, который необходим для понимания изучаемой темы,
- примеры экзаменационных заданий с подробным разбором, обсуждением нескольких вариантов решения и рекомендациями по выбору нужного,
- примеры задач для самостоятельной отработки решений задач, разобранных в соответствующей главе.

В конце пособия помещены ответы к заданиям для самостоятельного решения.

Автор надеется, что это пособие окажется полезным Вам, дорогой читатель.

В связи с возможными изменениями в формате и количестве заданий рекомендуем в процессе подготовки к экзамену обращаться к материалам сайта официального разработчика экзаменационных заданий — Федерального института педагогических измерений: www.fipi.ru.

Успехов на экзамене!

1. Информация. Язык как способ представления и передачи информации: естественные и формальные языки

Существует много различных определений термина **информация**. Автору больше всего нравится такое: «Информация — это сведения об окружающем нас мире». Если придирчиво отнестись к этому понятию, получается, что термин «информация» определен через термин «сведения». А если попытаться дать определение термину «сведения», то следует сослаться на термин «данные». В любом случае окажется, что это синонимы, то есть, термин «информация» — это некоторое первичное понятие.

Существует несколько основных действий, которые можно совершать с информацией:

- хранение информации,
- передача информации,
- обработка информации.

Для осуществления этих действий удобнее всего использовать некоторую форму/способ, при помощи которой информация хранится–передается–обрабатывается. Вероятнее всего, первичным действием при этом имеет смысл считать процесс передачи информации. Для выживания люди объединяются в группы, а для совместных действий нужно между собой договариваться. Отсюда и возникает проблема «общего языка» — средства общения между людьми.

За тысячелетия эволюции люди придумали множество различных языков, которые использовались для общения. Эти языки называются **естественными**. Они развились естественным образом в процессе эволюции. Необходимость хранить информацию привела к тому, что люди стали записывать языки разными способами: придумали отдельные буквы, которыми обозначали произносимые звуки. Эти буквы образовали *алфавит*.

Алфавит — набор символов, использующийся для записи слов языка. Со временем образовалось довольно большое количество алфавитов, которыми люди продолжают пользоваться до сих пор (вы наверняка знаете кириллицу, латиницу и греческий алфавиты). Часть народов использовала для записи отдельные слоги или даже целые слова (например, иероглифы). Для удобной обработки информации люди придумали более сложные обозначения, например, цифры для записи чисел.

При дальнейшей эволюции оказалось, что в некоторых случаях использование естественных языков неудобно. Например, через некоторое время после изобретения компьютера оказалось, что человеку общаться с компьютером при помощи естественного языка затруднительно (компьютер был не настолько умен, чтобы понимать все особенности и тонкости какого-нибудь естественного языка). И одним из решений этой проблемы было изобретение формальных языков, которые были бы понятны компьютеру и в то же время которым можно было бы относительно просто научить человека.

Например, к этой категории относятся **языки программирования**. Это формальные языки с весьма ограниченным набором слов, которые можно перевести на язык машинных кодов (в котором работает компьютер), но при этом стараются подобрать такие слова, чтобы общение на них было бы понятно человеку.

Программы-переводчики с языка программирования на язык машинных кодов, называются **трансляторами**. Если человек умеет записывать свои мысли на этом формальном языке (языке программирования), то он имеет возможность общения с компьютером. Такие специалисты называются **программистами**.

2. Формализация описания реальных объектов и процессов, моделирование объектов и процессов

Для обработки данных на компьютере все процессы, объекты, явления, должны быть записаны в виде чисел, так как компьютеры умеют работать только с числами. Все остальные виды информации, которые обрабатывает компьютер, преобразовываются в числовой вид. Существует даже специальный термин — **оцифровка**. Он означает, что информация преобразовывается в числа и может после этого обрабатываться, передаваться и храниться при помощи компьютеров. Люди придумали множество устройств, которые умеют делать оцифровку. Наиболее часто используемые из них — это цифровые фотоаппараты и камеры, сканеры, клавиатуры, мыши, планшеты.

Итак, все данные, которые компьютер обрабатывает, записывают в виде чисел. Иногда это достаточно очевидно. Например, для обработки информации о том, в каком месте экрана вы щелкнули мышью, сохраняется местоположение этой точки на экране (координаты по осям X и Y). Когда вы нажимаете клавишу на клавиатуре, фиксируется номер (код) этой клавиши. По нему определяется, какой символ соответствует клавише, и номер этого символа запоминается. При хранении величин, которые уже имеют числовое значение, это значение просто переводится в двоичную систему счисления (с некоторой точностью, если оно не целое).

Кроме хранения различных величин, при обработке объектов и процессов на компьютере возникает также задача компьютерного описания обрабатываемых объектов и процессов. Основной прием для этого аналогичен тому, который мы только что описали для хранения различных величин. Люди описывают состояние объектов или процессов, перечисляя те состояния или значения характеристик, которыми объекты или процессы могут обладать. После этого составляют таблицу всех возможных значений, нумеруют их и хранят номера в двоичном виде.

Рассмотрим пример, на котором покажем способ компьютерного хранения векторного графического изображения, то есть, изображения, описываемого набором объектов. Например, будем считать, что изображение может состоять только из прямых линий, прямоугольников и эллипсов. Линии, не являющиеся прямыми, будем хранить при помощи последовательности большого количества маленьких прямых отрезков.

Замечание. Для хранения кривых линий придумана специальная технология, называющаяся **кривые Безье**.

Итак, в нашем примере изображение состоит из объектов трех видов: прямой отрезок, прямоугольник, эллипс. Пронумеруем эти виды объектов: отрезок — 0, прямоугольник — 1, эллипс — 2. Расположение каждого из этих объектов можно описать при помощи двух точек. Местоположение точек будем хранить как их координаты на плоскости (два числа — проекция на ось X и проекция на ось Y). Для отрезка эти точки задаются координатами, из которой и в которую он проведен. Для прямоугольника — левый верхний и правый нижний угол. Для эллипса — координаты левого верхнего и правого нижнего углов того прямоугольника, в который вписан эллипс. Если мы хотим хранить прямоугольники и эллипсы, не параллельные осям координат, то для объектов хранится еще угол поворота объекта относительно своего центра. Получается, что для хранения каждого объекта достаточно хранить 6 чисел:

- целое число от 0 до 2 (вид объекта),
- две пары чисел, хранящих координаты точек на плоскости,
- угол поворота (например, в градусах).

Все эти величины (кроме номера вида объекта) могут быть целыми или вещественными в зависимости от того, с какой точностью требуется их хранение.

Исходя из подобных соображений описывется любой объект или процесс реального мира. Например, для хранения в компьютере модели трехмерного объекта можно разбить внешнюю поверхность объекта на связанные друг с другом треугольники. Каждый треугольник хранить при помощи трех троек чисел (координаты вершин треугольника в трехмерном пространстве). Для хранения правдоподобного изображения объекта для каждого треугольника хранится изображение **(текстура)**, которое на треугольник будет наложено. О хранении растровых изображений рассказано в одном из последующих разделов.

Например, при хранении процесса движения лифта достаточно перечислить те действия, которые лифт должен уметь делать: не двигаться, открывать двери, закрывать двери, подниматься, опускаться, — пронумеровав действия (от 0 до 4) и присвоив им соответствующие коды-номера. Также, вероятно, потребуется хранить информацию о том, где в текущий момент находится лифт (на каком этаже или между ними), на какой этаж он должен ехать, закрыты у него двери или открыты. Пронумеровав отдельно все состояния, независимые от других, получим компьютерную модель, которую можно использовать для написания программы управления лифтом.

3. Дискретная форма представления информации. Единицы измерения количества информации

При обработке различных данных люди столкнулись со следующей проблемой: компьютеры обрабатывают информацию в числовом виде, т.е. в виде чисел. При этом числа хранятся либо в целом виде, либо с некоторой точностью, которая зависит от количества ячеек памяти **(бит)**, выделенное для хранения числа. Объекты Окружающего нас мира не обладают «целочисленностью» (большинство величин, которые нас окружают, не являются целыми). Целыми числами мы подсчитываем только количество чего-нибудь. Практически все остальные величины окружающего нас мира — **вещественные**. Например, размеры объектов (длина, ширина, высота) могут считаться целыми только с точки зрения цены деления линейки, которой эти они измерены. То же самое касается и большинства других величин: время, масса, скорость, расстояние, сила тока, напряжение, температура.

Измерительные приборы выдают нам значение с некоторой степенью точности **(погрешности)**. При обработке величин размеров на компьютере приходится сталкиваться еще с одной проблемой — потерей точности. Эта проблема возникает при *дискретизации*. **Дискретизация** — запись измеряемой/хранимой величины при помощи определенного, ограниченного количества ячеек памяти (*бит*).

Люди используют три основных способа дискретизации хранимых данных (записи данных при помощи определенного количества двоичных разрядов).

Первый способ. Каждому варианту данных ставится в соответствие своя комбинация двоичных разрядов. Этот способ используется в случае, когда вариантов хранимых данных мало и их все можно выписать и «пронумеровать». Как правило, проще даже не сопоставлять каждому варианту свое отдельное «хитрое» значение двоичных разрядов, а просто выписать весь список вариантов данных в определенном порядке и пронумеровать каждый элемент списка, начиная с нуля. Двоичные значения номеров списка нужно хранить.

Второй способ. Каждому значению сопоставляется определенное целое число. В случае, если хранимые величины при этом не являются целыми числами (например, 6,73 кг), этим способом все равно можно воспользоваться.

Например, пусть мы хотим обрабатывать на компьютере массу продуктов, которые измеряются на весах. Значение массы электронные весы будут хранить в ячейке памяти определенного размера. Размер ячейки памяти не может быть бесконечным. Ничего бесконечного компьютеры обрабатывать не в состоянии. Значит, у ячейки будет определенное ограничение на то количество двоичных разрядов, которое будет для нее выделено. Пусть наибольшая масса, которую мы планируем взвешивать и обрабатывать, составляет 8 кг, то есть от 0 до 8 кг. От того, сколько двоичных разрядов будет выделено внутри электронных весов для хранения массы продукта, зависит, с какой точностью они будут обрабатываться. Пусть, для хранения массы мы выделим всего один бит памяти. Один бит может принимать только два возможных значения — 0 и 1. А мы, напоминаем, планируем взвешивать продукты массой от 0 до 8 кг. Значит, любой продукт, который будет иметь массу меньше 4 кг, мы будем кодировать значением 0, а для остальных значений массы — 1.

При обратном преобразовании (из бита в килограммы) возникает вопрос, каким значениям в килограммах соответствуют значения, хранимые в битах. Например, значение 0 — это сколько килограммов? Это зависит от того, хотим ли мы точно измерять граничные значения массы (0 и 8 кг) или более точно измерять массы в среднем. Например, можно считать, что 0 — это 0 кг, а 1 — это 8 кг. Тогда эти (граничные) значения будут храниться точно, а остальные — не очень. В худшем случае (например, при массе 4 кг) погрешность дискретизации будет достигать 4 кг.

Можно уменьшить погрешность вдвое, если принять за точные значения 0 и 1 не крайние значения диапазона, а средние в каждом. То есть, считать, что хранимому нулю соответствует масса 2 кг, а хранимой единице — масса 6 кг. Тогда наибольшая погрешность будет происходить при массе 0, 4 или 8 кг и будет составлять 2 кг.

Очевидно, что хранить значение массы при помощи только одного бита (двоичного разряда) никто не будет из-за возникающей погрешности. Но этот пример позволяет лучше понять проблему потери точности при дискретизации.

Предположим, что для хранения массы мы используем два двоичных разряда. Тогда нам доступно всего 4 различных дискретных значений (0, 1, 2 и 3). Если уменьшать потерю точности, этим двоичным значениям будут соответствовать значения массы 1, 3, 5 и 7 кг соответственно. То есть, если измеряемая масса составляет от 0 до 2 кг, храним это как дискретное число 0 (00 двоичное). При обратном преобразовании все эти значения массы будут считаться как 1 кг (наибольшая погрешность — 1 кг). Аналогич-

но, от 2 до 4 кг храним как 1 (01 двоичное) и считаем как 3 кг, от 4 до 6 кг храним как 2 (10 двоичное) и считаем как 5 кг, от 6 до 8 кг храним как 3 (11 двоичное) и считаем как 7 кг.

Нетрудно заметить, что при добавлении одного разряда для хранения массы, погрешность дискретизации уменьшается в два раза. Можем вычислить, какое наименьшее количество двоичных разрядов требуется использовать в данном примере, чтобы потеря точности при дискретизации была не более одного грамма:

Храним, бит	1	2	3	4	5	6	7	8	9	10	11	12
Точность, г	2000	1000	500	250	125	62,5	31,25	15,625	7,8125	3,90625	1,953	0,977

Получается, что для хранения массы от 0 до 8 кг с точностью до 1 грамма требуется не менее 12 бит.

Обратите внимание, что точность 1 грамм означает, что наибольшая ошибка измерений будет составлять не более 1 грамма. Если же требуется хранить значение массы с шагом 1 грамм, точность следует сделать равной 0,5 грамма, то есть, использовать еще один бит (всего 13). Это значение (13 бит) можно получить другим способом: посчитать, какое количество различных значений должны уметь хранить весы для измерений с шагом 1 грамм. Это 8001 (целое значение массы в граммах — от 0 до 8000). Ячейка памяти, состоящая из k бит, может хранить не более чем 2^k различных значений (от 0 до $2^k - 1$). Найдем наименьшее k такое, что $2^k \geqslant 8001$. Оно равно 13 ($2^{13} = 8192$), то есть, 13 бит.

Замечание.
Описанный нами только что способ называется записью вещественных чисел с **фиксированной запятой**. В этом способе погрешность дискретизации одинакова как для больших чисел, так и для маленьких. В случае с весами в магазине, где человек платит за вес с точностью до грамма, это актуально.

Третий способ — использование хранения вещественных чисел с **плавающей запятой**. Его принципиальное отличие от хранения чисел с фиксированной запятой состоит с том, что у числа всегда хранится какое-то гарантированное количество **значащих цифр**. Так, для очень маленьких значений массы (например, 0,00356 грамм) можно обеспечить такое же количество значащих цифр, что и для больших значений массы (например, 3 560 000 кг). Это нужно тогда, когда важна относительная погрешность измерений. Например, измеряя массу планеты, не очень важно, сколько

именно десятков или сотен килограммов она составляет, а, измеряя массу молекулы, точности измерения 1 грамм будет явно недостаточно. Изучить описание того, как устроено хранение вещественных чисел в плавающей запятой, мы предлагаем самостоятельно тем, кого это заинтересовало.

Рассмотрим теперь вопрос **измерения количества информации**.

Для измерения количества информации придуманы специальные **единицы измерения информации**.

Основная — *один бит*.

Один бит — это количество информации, уменьшающее неопределенность в два раза.

Что такое неопределенность? Проще всего это понимать как выбор из нескольких вариантов.

Например, Вася должен угадать, в какой из восьми одинаковых коробок, стоящих в ряд, лежит конфета.

Ему говорят, что конфета лежит в одной из левых 4-х коробок.

У Васи был выбор из 8-ми коробок, а остался выбор только их 4-х коробок, то есть, количество вариантов уменьшилось в 2 раза. Неопределенность уменьшилась в 2 раза. Значит, Васе сообщили 1 бит информации. Чтобы прийти к определенности (в какой конкретно коробке лежит конфета), нужно, чтобы остался только один вариант. Значит, нужно оставшееся количество вариантов поделить на 2 (останется 2 варианта, Вася получил еще один бит информации). А затем оставшееся количество вариантов поделить еще раз на 2 (останется 1 коробка, Вася получил еще один бит информации).

Итак, если изначально у нас был выбор из 8-ми коробок, и мы узнали, что конфета лежит в некоей конкретной коробке, мы должны были эту неопределенность (8 коробок) трижды поделить на 2, чтобы осталась только одна коробка. То есть, получить 3 бита информации.

Из этих соображений выведем основную **формулу для вычисления количества информации**.

Пусть у нас есть выбор из N одинаковых объектов. Мы должны выбрать один.

Будем делить количество объектов на 2 (уменьшать неопределенность в 2 раза) столько раз, сколько нужно для получения определенности (чтобы остался только один объект).

Получаем формулу:

$$N/2^i = 1.$$

Заметим, что ровно 1 можно получить только в том случае, если число N является степенью числа 2.

Если же это не так (а это часто не так), нужно понимать, что **бит — минимальная единица измерения информации и она не**

бывает не целой. То есть, если в процессе деления на 2 будут получаться не целые числа, нужно округлять до ближайшего целого (вверх).

Например, при исходных 5 объектах, нужно делить так же 3 раза, как если бы их было 8 (5/2 = 2,5; округляем до 3, 3/2 = = 1,5, округляем до 2; 2/2 = 1. Всего делили 3 раза). Можно это понимать и по-другому: следует считать, что определенность — это когда возможных вариантов остается не больше одного. То есть, нашу формулу правильнее было бы записать так: $N / 2^i \leqslant 1$.

Если перемножить обе части неравенства на 2^i, получим **формулу Хартли**:

$$2^i \geqslant N,$$

где N — количество равновероятных событий, i — количество информации (бит) в сообщении об одном таком событии, при этом i — минимальное целое число.

Заметим, что если у нас есть выбор только из одного варианта, то количество информации в сообщении о таком событии равно нулю (сообщение о событии, которое происходит всегда, не несет в себе информации).

Пример экзаменационной задачи

В магазине продается 30 одинаковых упаковок шоколадных шариков. Известно, что в одной из этих упаковок находится приз. Вася покупает одну упаковку. Какое количество информации содержится в сообщении о том, что приз находится именно в упаковке, купленной Васей?

Решение.

Первый способ. Анализируем исходные данные. Так как все 30 упаковок одинаковые и приз находится только в одной из них, то сообщение, что приз находится именно в упаковке, купленной Васей, — это одно из 30 равновероятных событий. То есть, мы применяем формулу Хартли: $2^i \geqslant N$. Здесь N — количество равновероятных событий (30). Нужно подобрать наименьшее целое i такое, что $2^i \geqslant 30$. Если Вы не знаете наизусть степени числа 2 (что весьма полезно для подготовки и сдачи ОГЭ по информатике), предлагаем подбирать эти степени последовательно, начиная с первой:

$2^1 = 2$. $2 \geqslant 30$? Нет. Берем следующую степень (домножаем на 2).

$2^2 = 2 \cdot 2 = 4$. $4 \geqslant 30$? Нет. Берем следующую степень (домножаем на 2).

$2^3 = 4 \cdot 2 = 8$. $8 \geqslant 30$? Нет. Берем следующую степень (домножаем на 2).