

Федеральное агентство по образованию
Озерский технологический институт (филиал)
ГОУ ВПО «Московский инженерно-физический институт
(государственный университет)»

Кафедра прикладной математики

ТПМ

Требования к программным модулям

Методические указания

Озерск, 2006

УДК 681.3
П 56

Вл. Пономарев.
ТПМ. Требования к программным модулям.
Методические указания. Ред. 1.
Озерск: ОТИ МИФИ, 2006. — 40 с.

Методические указания содержат требования к текстам программных модулей учебных работ студентов, нормы их оценки и правила печатного оформления. В конце документа приведён краткий русско-английский словарь, призванный облегчить подбор имени объекта программы (ок. 1000 слов), а также таблицы символов ASCII.

Указания предназначены для студентов начальных курсов, обучающихся по специальности «Программное обеспечение вычислительной техники и автоматизированных систем». Могут быть полезны также студентам старших курсов.

Содержание

0. Введение	4
1. Требования к текстам.....	5
1. Файловый комментарий.....	5
2. Функциональный комментарий	6
3. Операторный комментарий	7
4. Структурный отступ.....	8
5. Структурная лесенка	9
6. Имя объекта модуля	10
7. Имя локального объекта	11
8. Литерал.....	12
9. Пустая строка.....	13
10. Пробел	14
2. Нормы оценки.....	15
1. Своевременность	15
2. Отлаженность	15
3. Структурированность.....	16
4. Оптимизированность	17
5. Простота	17
6. Порядок	17
7. Защита	17
3. Печатное оформление.....	18
Приложения.....	19
Правила оформления операторов управления	19
Примеры на языке Си (C++).....	19
Примеры на языке Visual Basic	20
Примеры на языке Pascal	22
Словарь.....	24
Способы образования имени	37
Таблицы символов.....	38

0. Введение

1. Настоящий документ устанавливает требования, предъявляемые к текстам зачетных учебных программ, нормы их оценки и правила печатного оформления.
2. Зачетная учебная программа — это один или несколько файлов, составляющих в целом текст(ы) учебной программы, предъявляемые для зачета (защиты).
3. Каждый отдельный файл, предъявляемый к зачету, именуется в дальнейшем как «программный модуль».
4. К зачету принимается программный модуль, который удовлетворяет требованиям, описанным в разделе 1.
5. Оценка программного модуля производится в соответствии с нормами, приведенными в разделе 2.
6. Печатное оформление программного модуля должно выполняться по правилам, описанным в разделе 3.
7. В тексте документа термин «функция» относится также ко всем видам процедур.
8. Требования и нормы распространяются на тексты программ, написанных на языках высокого уровня, таких, как Си, С++, Pascal, ObjectPascal и Visual Basic. Требования к специфичным программным модулям, таким, как тексты на языках ассемблер, Prolog, Lisp, а также тексты скриптов, устанавливаются другими документами.

1. Требования к текстам

Требования устанавливают правила оформления текстов, которым должны соответствовать предъявляемые к зачету программные модули.

1. Файловый комментарий

1. В начале каждого программного модуля должен располагаться файловый комментарий из 8 обязательных строк:
 1. Имя файла программного модуля
 2. Наименование организации
 3. Наименование учебной группы
 4. Фамилия Имя Отчество автора
 5. Наименование учебной дисциплины
 6. Название (имя) программы (проекта)
 7. Назначение модуля
 8. Дата начала работы над программным модулем
2. Имя программы должно быть единым для всех программных модулей данной программы.
3. Назначение программного модуля должно пояснять его функциональные обязанности в программе. Для программ, состоящих из одного модуля, его назначение определяется словами «Основной модуль». Для программ, состоящих из нескольких программных модулей, модуль, содержащий стартовую (главную, начальную) функцию, должен иметь название «Основной модуль».

Пример

```
/* файл rcnt.c          */
/* ОТИ МИФИ           */
/* 1ПО-99Д             */
/* Петров Иван Семенович */
/* Алгоритмические языки и программирование */
/* Программа RCNT      */
/* Основной модуль     */
/* 01.03.2006          */
```

4. Файловый комментарий может быть расширен дополнительной информацией, например, пояснением к программе, авторским знаком и т.п.

2. Функциональный комментарий

1. Каждой функции программного модуля должен предшествовать комментарий, описывающий назначение, входные и выходные данные, возвращаемое значение и, возможно, метод и/или алгоритм. Исключение составляют процедуры событий.
2. Входные и выходные данные описываются в порядке их объявления в заголовке функции.

Пример

```
/* функция FileProcess */
/* Назначение: */
/*  читает файл построчно и считает число страниц */
/* Входные данные: */
/*  File - файловая переменная */
/* Выходные данные: */
/*  отсутствуют */
/* Возвращаемое значение: */
/*  отсутствует */
void FileProcess(FILE* File) {
    /* счетчик строк */
    int n = 0;
    /* построчно читаем файл */
    while (fgets(Buffer, 128, f)) {
        /* если прочитано PAGE_LEN строчек */
        if (++n == PAGE_LEN) {
            /* добавляем одну страницу */
            CountPages(&PageCounter, 1);
            /* обнуляем счетчик строк */
            n = 0;
        }
    }
    /* если счетчик строк не равен нулю */
    if (n > 0) {
        /* добавляем последнюю страницу */
        CountPages(&PageCounter, 1);
    }
    /* выводим количество страниц */
    printf("\nTotal Pages: %d\n", PageCounter);
}
```

3. Основной функции должен предшествовать комментарий «Основная функция» с пояснением параметров и возвращаемого значения. Пример см. пункт 3.2.

3. Операторный комментарий

1. Отдельный оператор программы сопровождается описанием его назначения. Комментарий должен пояснять причину и/или цель появления оператора, и не должен дублировать выполняемое оператором действие.
2. Операторный комментарий располагается в строке непосредственно перед поясняемым оператором. Допускается комментарий на нескольких строчках. Длина одной строки комментария не должна превышать 70–78 знаков.

Пример

```
/* Основная функция */
/* Входные данные: */
/* argn - количество параметров в командной строке */
/* args - массив параметров командной строки */
void main(int argn, char* args[]) {
    /* Проверяем число аргументов командной строки */
    /* Первый аргумент - сама программа */
    /* Второй аргумент - обрабатываемый файл */
    if (argn != 2) {
        /* Разрешен только один параметр */
        printf("Only one parameter is allowed.\n");
        return;
    }
    /* пробуем открыть и обработать файл */
    try {
        /* Открывая файл, проверяем результат
        if (f = OpenFileForRead(args[1])) {
            /* Обрабатываем файл */
            FileProcess(f);
        } else {
            /* Не удалось открыть */
            printf("\nFile not found: %s\n", args[1]);
        }
    }
    /* обработчик исключений */
    catch (...) {

        printf("\nSomething is wrong.\n");
        return;
    }
}
```

4. Структурный отступ

1. Структурные единицы — это операторы управления, такие, как условный оператор, оператор цикла, оператор выбора, а также функции. Структурные единицы отличаются наличием последовательностей операторов, составляющих *тела* или *ветки* структурных единиц.
2. Текст тела или ветки должен иметь смещение вправо, называемое *структурным отступом*.
3. Величина структурного отступа принимается равной 4 знакам «пробел».

Примеры

Отступ тела функции

```
void main(void) {  
    [ Тело функции ]  
}
```

Отступ веток условного оператора

```
if (условие) {  
    [ Ветка «Истина» ]  
} else {  
    [ Ветка «Ложь» ]  
}
```

Отступ тела оператора цикла

```
while (условие) {  
    [ Тело цикла ]  
}
```

Отступы вложенных структурных единиц

```
int foo(параметры) {  
    while (условие) {  
        if (условие) {  
            [ Ветка «Истина» ]  
        }  
    }  
}
```


5. Структурная лесенка

1. Тело или ветка структурной единицы заключается в *операторы* или *скобки* так, чтобы программный текст составлял непрерывную структурную «лесенку». Для обеспечения «лесенки» структурные единицы должны иметь оформление, предписанное приложением «Правила оформления операторов управления».

Пример

```
/* ... */
void foo(параметры) {
    /* ... */
    int имя;
    /* ... */
    while (условие) {
        /* ... */
        if (a > b) {
            /* ... */
            for (выражение ; выражение ; выражение) {
                /* ... */
                if (условие) {
                    /* ... */
                    оператор(ы)
                } else {
                    /* ... */
                    оператор(ы)
                }
            }
        }
    }
}
do {
    /* ... */
    if (условие) {
        /* ... */
        оператор(ы)
    } else {
        /* ... */
        оператор(ы)
    }
    /* ... */
} while (1)
}
```

6. Имя объекта модуля

1. Объект модуля — это внешняя переменная, константа, перечисление, тип, прототип функции или переменная, описанная в начале модуля, а также функция.
2. Имя объекта модуля должно отражать его назначение полными словами английского языка.
3. Имя объекта модуля (за исключением функций) может начинаться с префикса, отражающего его принадлежность к объекту модуля и/или тип (например, при помощи *венгерской нотации*).

Пример

```
long m_lngTotalPagesCounter;
```

4. Рекомендуются регистр имен переменных, типов и функций — «Первая Буква Слова Прописная». Разрешен регистр «нижний» для однословных имен в программах на Си (C++).

Пример

```
long TotalPagesCounter;
```

5. Регистр имен констант без префикса — ВЕРХНИЙ.

Пример

```
#define MAX_PAGES 30
```

6. Регистр имен констант с префиксом — «Первая Буква Слова Прописная».

Пример

```
Const cMaxPages = 30
```

7. Префиксы констант: **c** — префикс числовой константы, **cs** — префикс строковой константы.

Пример

```
Public Const csPlayer = "Player"
```

8. Между словами можно вставлять знак подчеркивания.

Пример

```
long m_lng_Page_Counter;
```

9. Все имена во всех модулях должны иметь единый стиль.
10. Данное требование может уточняться для отдельных видов программных модулей другими документами.

7. Имя локального объекта

1. Локальный объект — это константа или переменная, описанная в теле функции, а также параметр функции.
2. Имя локального объекта должно состоять из одной буквы и в некоторых случаях из двух или трех [одинаковых] букв.

Пример

```
void main(void)
{
    int i, j, k;
}
```

3. Рекомендуемые имена локальных переменных

<i>Имя</i>	<i>Тип</i>
A, a	вещественный, символьный
B, b	вещественный, символьный
C, c	вещественный, символьный
D, d	вещественный, символьный
E, e	вещественный, символьный
F, f	логический
G, g	логический
H, h	логический
I, i	параметр цикла
J, j	параметр цикла
K, k	параметр цикла
L	целый
M, m	целый
N, n	целый
P, p	целый
o	объектный
Q, q	объектный
R, r	структура, запись
S, s	строковый
T, t	строковый
U, u	строковый
V, v	строковый
W, w	строковый
X, x	вещественный
Y, y	вещественный
Z, z	вещественный

4. Имена параметров функции подчиняются также требованию 6.4.

8. Литерал

1. Литерал — это явная запись значения.

Примеры литералов (подчеркнуты)

```
void main(void) {
    int i = 5;
    float a = 1.0;
    char* s = "Пример строкового литерала";
}
```

2. Текст программного модуля должен содержать литералы только в описаниях констант.
3. В тексте программы разрешаются символьные литералы при инициализации.

Пример

```
void main(void) {
    char c = 'A';
}
```

4. В тексте программы разрешаются строковые литералы при инициализации.

Пример

```
void main(void) {
    char* s = "Пример допустимого литерала";
}
```

5. Числовые литералы 0, 1 и 2 допускаются в операторах инкремента, декремента и в качестве начального значения.

Пример

```
void main(void) {
    /* ... */
    int i = 1;
    /* ... */
    for ( ; i < MAX_VALUE; ) {
        /* ... */
        i += 2
    }
}
```

9. Пустая строка

1. Фрагмент текста — это блок описаний внешних переменных, блок описаний констант, описание или блок описаний перечислений, описание или блок описаний типов, блок прототипов функций, блоки описаний переменных, функции вместе с предшествующим комментарием.
2. Между отдельными фрагментами текста должна быть ровно одна пустая строка.

Пример

```
#include <stdio.h>

/* Размер страницы в строчках */
#define PAGE_LEN 60

/* файловая переменная */
FILE* f = NULL;
/* Буфер для строки */
char Buffer[128];
/* Счетчик страниц */
int PageCounter = 0;

/* Процедура CountPages */
/* Назначение: */
/* подсчет страниц */
/* Входные данные: */
/* Counter - счетчик страниц */
/* Pages - число обработанных страниц */
/* Выходные данные: */
/* Counter - счетчик страниц */
/* Возвращаемое значение: */
/* отсутствует */
void CountPages(int* Counter, int Pages) {
    /* */
    *Counter += Pages;
}
```

3. В конце текста должна быть ровно одна пустая строка.

Для проверки требования 9.3 следует ввести сочетание клавиш «Конец текста» (Ctrl+End или Ctrl+PageDown). Точка вставки (курсор) должна находиться точно в начале строки, которая следует непосредственно за последней программной строкой.

10. Пробел

1. Пробелы, наряду со структурным отступом, улучшают зрительное восприятие текста, обеспечивая легкое его понимание.
2. Символы двухместных операций должны быть окружены пробелами.

Пример

```
f = !((a || b) && (a || c));
```

3. Символы операций . (точка), -> (стрелка) и :: (разрешение видимости), соединяющие идентификаторы (имена), пробелами не окружаются.

Пример

```
n = List::Counter;
```

5. Символы одноместных операций примыкают к типу или к переменной.

Пример

```
int* a[2];  
if (++n == PAGE_LEN) ...
```

6. После знака «запятая» следует пробел.

Пример

```
while (fgets(Buffer, 128, f)) ...
```

7. После ключевого слова следует пробел.

Пример

```
while (fgets(Buffer, 128, f)) {  
    /* если прочитано PAGE_LEN строчек */  
    if (++n == PAGE_LEN) {  
        /* добавляем одну страницу */  
        CountPages(&PageCounter, 1);  
        /* обнуляем счетчик строк */  
        n = 0;  
    }  
}
```

8. Двойные пробелы не допускаются за исключением случаев выравнивания текста для улучшения зрительного восприятия.

2. Нормы оценки

1. Нормы устанавливают критерии оценки программных модулей и программ в целом.
2. Первоначально программный модуль оценивается в 50 баллов при условии выполнения *всех* требований пункта 1. В отдельных случаях программа принимается к оценке без выполнения требований пункта 1 с начальной оценкой 0 баллов.
3. Оценка программного модуля либо увеличивается, либо уменьшается применением одной из норм. Если оценка становится меньшей или равной нулю, программа может быть аннулирована с выдачей нового задания.
4. Изменения, которым программа может подвергнуться во время оценки, не принимаются во внимание. Программа оценивается такой, какой она была предъявлена для зачета.
5. Оценка программы в целом может быть получена как среднее арифметическое оценок программных модулей.
6. Оценка, полученная после применения пунктов 2, 3 и 4, может быть приведена к пятибалльной системе по следующей шкале:

< 26	26 — 50	51 — 75	>76
2	3	4	5

1. Своевременность

1. Программному модулю, принятому в установленный срок или раньше, начисляется 10 баллов.
2. Программному модулю, принятому позже установленного срока, начисляется –1 балл за каждый день просрочки.

2. Отлаженность

1. Отлаженная программа содержит меньше ошибок. Отлаженность достигается тщательным тестированием и пошаговой проверкой работы отдельных частей и программы в целом.
2. Логические ошибки есть следствие неправильного алгоритма работы программы. Логические ошибки могут вызывать также «зацикливание». За наличие в программе логических ошибок ей начисляется –5 баллов.

3. Неправильный результат работы программы возникает в случае, если в программе не проверяются входные данные, а также вследствие потери точности вычислений. За неправильный результат работы программы ей начисляется –5 баллов.
4. Критические ошибки вызывают прекращение исполнения программы, а также «зависание» программы или операционной системы. Критические ошибки являются следствием непродуманности операторов программы. За наличие в программе критических ошибок ей начисляется –10 баллов.
5. При отсутствии ошибок программе начисляется 10 баллов.

3. Структурированность

1. Структурированность — это разделение алгоритма и элементов данных программы на составляющие её части — структуры данных, модули и функции.
2. Структуры данных описывают предметную область программы при помощи встроенных и пользовательских типов данных.
3. Программный модуль — это структурный элемент программы, объединяющий функции и/или элементы данных определенного назначения.
4. Функция — это структурный элемент программы, предназначенный для реализации конкретной и независимой цели. Функции располагаются внутри программных модулей.
5. Программе может быть начислено от 1 до 10 баллов, если:
 - структуры данных выражены четко и ясно и описаны с применением оправданных и правильных типов данных;
 - текст программы разделен на модули (для достаточно сложных программ), а реализация алгоритма структурирована при помощи тщательно подобранных функций, не имеющих зависимостей по данным.
6. Программному модулю, который содержит функции размером свыше 25 строк (исключая функциональный комментарий), может быть начислено от –1 до –10 баллов.
7. При наличии в программе оператора GOTO программе начисляется –10 баллов.

4. Оптимизированность

1. Оптимизированность — это минимизация количества вычислительных операций. Неоптимизированность проявляется в наличии повторяющихся вычислений одних и тех же значений.
2. Программе, текст которой содержит неоптимизированный код, наценивается от -1 до -10 баллов.

5. Простота

1. Простота — это логичность и стройность хода решения задачи, лаконичность алгоритма и отдельных операторов программы.
2. Программе, операторы которой отличаются простотой, ясностью и лаконичностью в совокупности с окружающими их комментариями, может быть наценено от 1 до 10 баллов. В эту оценку включается также простое изложение алгоритма.
3. Программе, операторы которой перегружены операциями, затрудняющими восприятие формулы вычисления, а также если алгоритм решения запутан, наценивается от -1 до -10 баллов.

6. Порядок

1. Порядок — это расположение фрагментов текста в программном модуле. Функции модуля располагаются в алфавитном порядке, если их много, или в порядке их использования, если их мало. Объекты модуля располагаются по группам, а в группах — по алфавиту или иному критерию сортировки. Отдельным фрагментам текста программы могут предшествовать специальные обобщающие комментарии, например, «Константы», «Переменные», «Свойства» и т.п.
2. Программному модулю может быть наценено от -10 до 10 баллов за наличие или отсутствие порядка.

7. Защита

1. Учебная программа подлежит защите её разработчиком. В ходе защиты разработчик обязан ответить на вопросы принимающего, касающиеся структуры программы, алгоритма и способов его реализации.
2. В ходе защиты принимающий наценивает программе от -30 до 10 баллов, в зависимости от подготовленности разработчика.

3. Печатное оформление

1. Для оформления текстов программ в печатных документах используется шрифт **Courier New**, начертание **полужирное**.
Масштаб символов принимается равным 100%.
2. Если программный текст располагается в основной части текста документа (как пример кода или как название переменной, объекта, функции и т.п.), размер шрифта принимается на единицу меньше размера основного текста.
3. Если программный текст оформляется как приложение, размер шрифта устанавливается равным 6 (для лазерного или струйного принтера) или 7 пунктов (для матричного принтера). Междустрочное расстояние в этом случае принимается точно равным размеру шрифта.
4. Во всех других случаях размер текста принимается равным 10 ± 1 пунктов.
5. Данное требование не распространяется на дипломные работы.

Приложения

Правила оформления операторов управления

Примеры на языке Си (C++)

Условный оператор if

1. `if (условие) оператор`
2. `if (условие) оператор else оператор`
3. `if (условие) {
 /* комментарий или пустая строка */
 оператор(ы)
}`
4. `if (условие) {
 /* комментарий или пустая строка */
 оператор(ы)
} else {
 /* комментарий или пустая строка */
 оператор(ы)
}`
5. `if (условие) {
 /* комментарий или пустая строка */
 оператор(ы)
} else if (условие) {
 /* комментарий или пустая строка */
 оператор(ы)
} else {
 /* комментарий или пустая строка */
 оператор(ы)
}`

Параметрический цикл for

6. `for (i = 0; i < n; i++) {
 /* комментарий или пустая строка */
 оператор(ы)
}`

Цикл с предусловием while

7. `while (условие) {
 /* комментарий или пустая строка */
 оператор(ы)
}`

Цикл с постусловием do

```
8. do {  
    /* комментарий или пустая строка */  
    оператор (ы)  
    /* комментарий на условие */  
} while (условие);
```

Оператор выбора switch

```
9. switch (выражение) {  
    case значение: {  
        /* комментарий */  
        оператор (ы)  
    }  
    default:;  
        оператор (ы)  
}
```

Структура (класс)

```
10. struct имя {  
    /* комментарий */  
    элемент  
private:  
    /* комментарий */  
    элемент  
};
```

Примеры на языке Visual Basic

Условный оператор If

1. If (Условие) Then оператор
2. If (Условие) Then оператор Else оператор
3. If (Условие) Then
 ' комментарий или пустая строка
 Оператор (ы)
End If
4. If (Условие) Then
 ' комментарий или пустая строка
 Оператор (ы)
Else
 ' комментарий или пустая строка
 Оператор (ы)
End If

```
5. If (Условие) Then
    ' комментарий или пустая строка
    Оператор (ы)
ElseIf (Условие) Then
    ' комментарий или пустая строка
    Оператор (ы)
Else
    ' комментарий или пустая строка
    Оператор (ы)
End If
```

Параметрический цикл For

```
6. For I = 1 To N Step M
    ' комментарий или пустая строка
    Оператор (ы)
Next
```

Цикл Do с предусловием

```
7. Do While (Условие)
    ' комментарий или пустая строка
    Оператор (ы)
Loop
```

Цикл Do с постусловием

```
8. Do
    ' комментарий или пустая строка
    Оператор (ы)
Loop Until (Условие)
```

Оператор выбора Select Case

```
9. Select Case выражение
Case Значение
    ' комментарий
    Оператор (ы)
Case Else
    Оператор (ы)
End Select
```

Тип, перечисление

```
10. Public Type Имя
    ' комментарий
    Элемент (ы)
End Type
```

Примеры на языке Pascal

Условный оператор if

1. `if (условие) then оператор`
2. `if (условие) then оператор else оператор`
3. `if (условие) then begin`
 `{ комментарий или пустая строка }`
 `оператор(ы)`
`end`
4. `if (условие) then begin`
 `{ комментарий или пустая строка }`
 `оператор(ы)`
`end else begin`
 `{ комментарий или пустая строка }`
 `оператор(ы)`
`end`
5. `if (условие) then begin`
 `{ комментарий или пустая строка }`
 `оператор(ы)`
`end else if (условие) then begin`
 `{ комментарий или пустая строка }`
 `оператор(ы)`
`end else if (условие) then begin`
 `{ комментарий или пустая строка }`
 `оператор(ы)`
`end else begin`
 `{ комментарий или пустая строка }`
 `оператор(ы)`
`end`

Параметрический цикл for

6. `for i := 0 to N step M do begin`
 `{ комментарий или пустая строка }`
 `оператор(ы)`
`end`

Цикл с предусловием while

7. `while (условие) do begin`
 `{ комментарий или пустая строка }`
 `оператор(ы)`
`end`

Цикл с постусловием repeat..until

```
8. repeat
    { комментарий или пустая строка }
    оператор(ы)
    { комментарий на условие }
until (условие)
```

Оператор выбора case

```
9. case выражение of
    значение: begin
        { комментарий }
        оператор(ы)
    end;
    значение: begin
        { комментарий }
        оператор(ы)
    end;
end
```

Запись

```
10. Имя = record
    { комментарий }
    элемент
end;
```

Словарь

Краткий словарь приведен для подбора имени объекта модуля и не может служить справочником для перевода текстов.

Способы образования имён приведены в конце словаря.

Латинский алфавит

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122
а	бэ	цэ	дэ	е	эф	жэ	аш	и	жи	ка	эл	эм	эн	о	пэ	ку	эр	эс	тэ	у	вэ	дубльвэ	икс	игрек	зэт

Число

число — *number*

цифра — *digit*

единица — *unity*

ноль (цифра) — *naught*

ноль, нуль — *zero*

нуль — *nil*

бесконечность — *infinity*

1 — *one*

2 — *two*

3 — *three*

4 — *four*

5 — *five*

6 — *six*

7 — *seven*

8 — *eight*

9 — *nine*

10 — *ten*

50 — *fifty*

100 — *hundred*

1000 — *thousand*

первый — *first*

второй — *second*

третий — *third*

энный — *nth*

половина — *half*

треть — *third*

четверть — *quarter*

десятая часть — *tenth*

сотая часть — *hundredth*

тысячная часть — *thousandth*

1234,567

единицы числа (4) — *units*

десятки числа (3) — *tens*

сотни числа (2) — *hundreds*

тысячи числа (1) — *thousands*

десятые доли (5) — *tenths*

сотые доли (6) — *hundredths*

тысячные доли (7) — *thousandths*

виды чисел

простое — *prime*

порядковое — *ordinal*

количественное — *cardinal*

целое — *integer*

вещественное — *real*

комплексное — *complex*

вещественная (часть) — *real (part)*

мнимая (часть) — *imaginary (part)*

положительное — *positive*

отрицательное — *negative*

чётное — *even*

нечётное — *odd*

разное

знак — *sign*

плюс — *plus*

минус — *minus*

определённое — *defined*

неопределённое — *undefined*

точное — *exact*

неточное — *inexact*

правильное — *correct*

неправильное — *wrong*

численный — *numerical*

числовой — *numeric*

целочисленный — *integral*

дробь — fraction

целая (часть) — *integer (part)*

дробная (часть) — *fraction (part)*

правильная дробь — *proper fraction*

неправильная дробь — *improper fraction*

числитель — *numerator*

знаменатель — *denominator*

сокращение — *cancellation*

формат с плавающей запятойплавающая точка — *floating point*мантисса — *mantissa*экспонента — *exponent*младший значащий бит — *LSB*старший значащий бит — *MSB***Счёт**счёт — *counting*считать — *count*счётчик — *counter*инкремент — *increment*декремент — *decrement*счётный — *denumerable*несчётный — *innumerable***система счисления**основание — *radix*двоичный — *binary*восьмеричный — *octal*десятичный — *decimal*шестнадцатеричный — *hexadecimal***Вычисления**больше (чем) — *greater (than)*величина — *quantity*выражение — *expression*вычисление — *calculation*вычислить — *compute*заём — *borrow*знак равенства — *equal sign*значение — *value*меньше (чем) — *less (than)*невязка — *discrepancy*неизвестное — *unknown*неравенство — *inequality*неравный — *unequal*операнд — *operand*оператор — *operator*оператор языка — *statement*операция — *operation*перенос — *carry*переполнение — *overflow*погрешность — *inaccuracy*равенство — *equality*равно — *equal (to)*равный — *equal*разрешить — *resolve*рассчитать — *calculate*результат — *result*решение — *solution*решить — *solve*тождественность — *identity*тождество — *identity*точность — *precision*точный — *precise*уравнение — *equation***сложение** — *addition*суммировать — *sum*слагаемое — *term*сумма — *sum*среднесуммарное — *average***вычитание** — *subtraction*вычесть — *subtract*уменьшаемое — *minuend*вычитаемое — *subtrahend*остаток — *remainder*разность — *difference***умножение** — *multiplication*умножить — *multiply*множимое — *multiplicand*множитель — *multiplier*сомножитель — *factor*произведение — *product***деление** — *division*поделить — *divide*делимое — *dividend*делитель — *divider*частное — *quotient*остаток — *remainder***возведение в степень** — *raising*основание — *base*степень — *power***извлечение корня** — *evolution*корень — *root*квадратный корень — *square root*кубический корень — *cube root***разное**брать производную — *derive*дифференциал — *differential*дифференцирование — *derivation*интеграл — *integral*интегральный — *integral*интегрирование — *integration*интегрировать — *integrate*полином — *polynomial, polynom*полиномиальный — *polynomial*прогрессия — *progression*производная — *derivative*ряд — *series*член (ряда, прогрессии) — *member***Геометрия, пространство**абсцисса — *abscissa*асимптота — *asymptote*Безье — *Bezier*биссектриса — *bisector*вершина — *vertex*вершины — *vertices*внешний угол — *exterior angle*внутренний угол — *interior angle*вписанный — *inscribed*высота (треугольника) — *altitude*

гипербола — *hyperbola*
гипотенуза — *hypotenuse*
грань — *face*
диагональ — *diagonal*
диаметр — *diameter*
директриса — *directrix*
дуга — *arc*
касание — *contact*
касательная — *tangent*
катет — *catheti*
квадрант — *quadrant*
квадрат — *square*
конус — *cone*
координата — *coordinate*
косинусоида — *cosine*
кривая (линия) — *curve*
кривизна — *curvature*
круг, окружность — *circle*
линия (прямая) — *line*
луч — *ray*
многоугольник — *polygon*
многоугольный — *polygonal*
объём — *volute*
описанный — *circumscribed*
ордината — *ordinate*
основание (треугольника) — *base*
ось (координат) — *axe*
парабола — *parabola*
параллелограмм — *rhomboid*
периметр — *circumference*
перпендикуляр — *perpendicular*
пирамида — *pyramid*
плоскость — *plane*
площадь — *area*
поверхность — *surface*
полукруг — *semicircle*
призма — *prism*
пространство — *space*
прямая линия — *straight line*
прямоугольник — *rectangle*
прямоугольный — *rectangular*
радиус — *radius*
ребро — *edge*
ромб — *rhombus*
секущая (секанс) — *secant*
синусоида — *sine*
сплайн — *spline*
сторона — *side*
сфера — *sphere*
сферический — *spherical*
тангенс — *tangent*
тело (сплошное) — *solid*
точка — *point*
точка касания — *point of contact*
трапеция — *trapezium*
треугольник — *triangle*

треугольный — *triangular*
угол (градус) — *angle*
угол (точка) — *corner*
фигура — *figure*
фокус — *foci*
хорда — *chord*
центр — *center, centre*
четырёхугольник — *quadrilateral*
эллипс — *ellipse*
эллиптический — *elliptical*

Множество

множество — *set*
дополнение — *complement*
объединение — *union*
отображение — *mapping, map*
пересечение — *intersection*
элемент — *member*

Время, дата

время — *time*
дата — *date*
сотая доля секунды — *hundredth*
секунда — *second*
минута — *minute*
час — *hour*
день — *day*
неделя — *week*
месяц — *month*
год — *year*
день недели — *day of week*
ежегодный — *annual*
еженедельный — *weekly*
ежедневный — *daily*
ежечасный — *hourly*
ежесекундный — *every second*
ежеминутный — *every minute*

Время года

весна — *spring*
лето — *summer*
осень — *autumn*
зима — *winter*

Месяц

январь — *january*
февраль — *february*
март — *march*
апрель — *april*
май — *may*
июнь — *june*
июль — *july*
август — *august*
сентябрь — *september*
октябрь — *october*
ноябрь — *november*
декабрь — *december*

День недели

понедельник — *monday*
вторник — *tuesday*
среда — *wednesday*
четверг — *thursday*
пятница — *friday*
суббота — *saturday*
воскресенье — *sunday*
выходной — *rest day, holiday*

Пол, род

пол, род — *gender*
мужской — *male*
женский — *female*
средний — *neuter*

Человек

человек — *man, person*
люди — *people*
население — *population*
присущий человеку — *human*
мужчина — *man*
женщина — *woman*
мужчины — *men*
женщины — *women*
мужской — *masculine*
женский — *female*
мальчик — *boy*
девочка — *girl*
отец — *father*
мать — *mother*
сын — *son*
дочь — *daughter*
родители — *parents*
родительский — *parental*
ребёнок — *child*
дети — *children*
семья — *family*
имя — *first name*
фамилия — *surname*
отчество — *patronymic*
старый — *old*
молодой — *young*
старший — *senior*
младший — *junior*
смелый — *bold*
робкий — *shy*
вес — *weight*
рост — *height*
возраст — *age*
части тела
глаз — *eye*
голова — *head*
лицо — *face*
палец — *finger*
рука — *arm*

рука (кисть) — *hand*
сердце — *heart*
тело — *body*
ухо — *ear*

Количество, качество

высокое (качество) — *high*
качество — *quality*
количество — *quantity*
количество (сумма) — *amount*
количество (число) — *number*
мало (качество) — *little*
мало (количество) — *few*
много (качество) — *much*
много (количество) — *many*
низкое (качество) — *poor*

Сторона света

север — *north*
юг — *south*
запад — *west*
восток — *east*
полюс — *pole*

Место

место — *place, site*
верх — *top*
внешность — *exterior*
внутренность — *interior*
зад — *rear, background*
левая сторона — *left, left side*
низ — *bottom*
перед — *front, foreground*
правая сторона — *right, right side*
середина — *middle, midst*
центр — *centre, center*

Положение

положение — *position*
вверху, выше — *above*
верхний — *top, up, upper*
внешний — *external, outer*
внизу — *down*
внутренний — *internal, inner*
внутри — *inside*
впереди, спереди — *ahead*
выше — *above*
за, позади — *behind, beyond*
за, после — *after*
между — *between*
на — *on*
наверху — *on top*
над — *above*
ниже — *below*
нижний — *bottom, down*
перед, до — *before*
под — *under*

позади, сзади — *behind*
самый верхний — *topmost*
середина, срединный — *middle*
середина, центр, центральный — *center*
слева — *left*
снаружи — *outside*
справа — *right*

Расстояние

расстояние — *distance*
ближайший — *nearest*
близкий — *near*
близко — *close*
далёкий — *distant*
далеко — *far*
дальний — *far*
удалённый — *remote*

Направление

направление — *direction*
в — *in*
из — *out, off*
к — *to, towards*
от — *from*
вон — *away*
вверх — *up, upwards*
вниз — *down, downwards*
влево — *to the left, left*
вправо — *to the right, right*
внутри — *inside*
наружу — *outside*
вперёд, прямо — *forward, ahead*
назад — *back, backwards*
направленный вперёд — *forward*
направленный наружу — *outward*
вдоль — *along*
поперёк — *across*

Лучше, хуже

лучше — *better*
лучший — *better, best*
наилучший — *best*

Определители значения

адекватный — *adequate*
альтернативный — *alternative*
асинхронный — *async, asynchronous*
ассоциативный — *assoc, associative*
базовый — *base, basic*
бедный — *poor*
без помех (сигнал) — *noiseless*
безопасный — *safe*
безымянный — *nameless*
бесполезный — *needless*
беспримесный — *pure*

наихудший — *worst*
плохой — *bad*
хороший — *fine*
худший — *worse, worst*
хуже — *worse*

Больше, меньше (размер)

большой — *bigger*
большой — *big*
маленький — *little, small*
меньший — *smaller*
наибольший — *biggest*
наименьший — *smallest*

Больше, меньше (величина)

больше, больший — *greater*
большой — *large, great*
маленький — *small*
меньше — *less*
меньший — *lesser*
наибольший — *greatest*
наибольший (что) — *most?*
наименьший — *smallest*
наименьший (что) — *less?*

Передача, приём

передача — *transmission*
приём — *reception*
передавать — *transmit*
принимать — *receive*
передатчик — *transmitter*
приёмник — *receiver*
посланный — *sent*
принятый — *received*
послать — *send*
принять — *accept*
отправитель — *sender*
получатель — *recipient*
источник (переменная) — *source*
приёмник (переменная) — *destination*
последовательный — *serial*
параллельный — *parallel*

бесчувственный — *insensible*
ближайший — *nearest*
блокированный — *locked*
блокирующий — *lock*
богатый — *rich*
быстрый — *fast*
важный — *important*
вводимый — *input*
вероятный — *probable*
взаимный — *mutual*
видимый — *visible*

включительный — *inclusive*
вложенный — *nested*
вовлечённый — *involved*
возможный — *possible*
временный — *temporal*
врожденный — *native*
вспомненный — *recollected*
вспомогательный — *accessory*
вторичный — *second*
второстепенный — *secondary*
входящий в диапазон — *fit*
выборочный — *selective*
выбранный — *chosen*
выводимый — *output*
выделенный (выбор) — *selected*
выработанный — *workout*
вычисленный — *computed*
главный — *chief*
глобальный — *global*
готовый — *ready*
граничный — *border*
грубый — *rough*
грязный — *dirty*
данный (кем-то) — *given*
действительный — *active*
действительный (матем.) — *real*
действующий — *effective*
длинный — *long*
добытый — *obtained*
доказанный — *argued*
дополнительный — *additional*
дополняющий — *complement*
допустимый — *valid*
допущенный — *admitted*
дорогостоящий — *expensive*
достаточный — *sufficient*
достоверный — *reliable*
доступный — *available*
дохлый — *dead*
дочерний — *child*
другой — *other*
жадный — *greedy*
жаждущий — *eager*
ждуций — *waiting*
ждуций обработки — *pending*
заблокированный — *locked*
забытый — *forgotten*
завершённый — *finished*
зависимый — *dependent*
заданный — *defined*
заключительный — *final*
законный — *legal*
замедленный — *retarded*
заменённый — *replaced*
занятый (несвободный) — *occupied*
занятый (работающий) — *busy*

записанный — *written*
заполненный — *full*
заполненный (массов) — *populated*
запрещённый — *forbidden*
запрошенный — *requested*
зафиксированный — *fixed*
захваченный — *captured*
защищённый — *protected*
знакомый — *familiar*
избранный (выборы) — *elected*
известный — *known*
извлечённый — *extracted*
изменённый — *changed*
изменчивый — *volatile*
инертный (неактивный) — *inert*
исключённый — *excluded*
искомый — *sought*
использованный — *used*
используемый — *inuse*
истёкший (срок) — *expired*
истинный — *true*
исходящий — *outgoing*
исчезающий — *disappearing*
итоговый — *total*
концевой — *end*
конечный (имеющий предел) — *finite*
конкретный — *specific*
корневой — *root*
короткий — *short*
корректирующий — *remedial*
купленный — *bought*
лёгкий (вес) — *light*
лёгкий (простой) — *easy*
ленивый — *lazy*
линейный — *linear*
ложный — *false*
локальный — *local*
медленный — *slow*
младший — *junior*
мнимый — *imaginary*
модифицированный — *modified*
навязчивый — *obsessive*
надёжный — *secure*
найденный — *found*
наполненный — *filled*
начальный — *initial*
не входящий в диапазон — *misfit*
неважный — *unimportant*
невидимый — *invisible*
невозможный — *impossible*
недавний — *recent*
недопустимый — *invalid*
недопущенный — *dismissed*
недостаточный — *insufficient*
недостоверный — *unreliable*
незавершённый — *unfinished*

независимый — *independent*
незаданный — *undefined*
незаконный — *illegal*
незанятый (свободный) — *vacant*
незнакомый — *unfamiliar*
неизбежный — *inevitable*
неизвестный — *unknown*
неизменный — *constant*
неистёкший (срок) — *unexpired*
нейтральный — *neutral*
некоторый — *some*
ненадёжный — *insecure*
ненормальный — *irregular*
необработанный — *raw*
необходимый — *needed*
необычный — *unusual*
необязательный — *optional*
неограниченный — *unlimited*
неожидаемый — *unexpected*
неопределённый — *undetermined*
неподходящий — *improper*
неполный — *incomplete*
неправильный — *wrong*
неприменимый — *inapplicable*
неприятный — *unpleasant*
несущественный — *inessential*
несуществующий — *unexistent*
неточный — *inexact*
нетронутый — *intact*
неуверенный — *uncertain*
неудачный — *unlucky*
неуспешный — *failed*
неустраивающий — *unsuited*
нехороший — *ill*
нечёткий — *indefinite*
нечётный — *odd*
нечувствительный — *insensitive*
неявный — *implicit*
новый — *new*
нормальный — *regular*
нулевой — *null*
обменный — *exchange*
обновлённый — *updated*
обработанный — *handled*
обратный (величина) — *reciprocal*
общий — *common*
обыкновенный — *ordinary*
ограниченный — *limited*
одиночный — *single*
ожидаемый — *expected*
ожидающий — *awaiting*
округленный — *rounded*
определённый — *determined*
организованный — *ordered*
основной — *main*
особенный — *special*

оставшийся — *left*
остаточный — *rest*
отвергнутый — *denied*
отложенный — *deferred*
относящийся — *related*
отправленный — *sent*
отправляемый — *send*
отрицательный — *negative*
отслеживаемый — *traced*
отслеживающий — *tracking*
отсутствующий — *absent*
отчётливый — *distinct*
охраняемый — *guarded*
оценённый — *evaluated*
оценочный — *estimated*
очевидный — *evident*
ошибочный — *mistaken*
первичный — *primary*
первоначальный — *original*
переключаемый — *toggled*
плотный — *dense*
плохой — *bad*
повторный — *repeated*
повышающийся — *rising*
подготовленный — *prepared*
поднимающийся — *ascending*
подозрительный — *suspicious*
подсчитанный — *calculated*
подтверждённый — *confirmed*
подходящий — *proper*
полный (завершённый) — *complete*
положительный — *positive*
полученный (приём) — *received*
пользовательский — *custom*
понижающийся — *sinking*
попыточный — *trial*
порядковый — *ordinal*
последний — *last*
постоянный — *permanent*
потерянный — *missed*
правильный — *correct*
праздный — *idle*
предварительный — *prior, preliminary*
предпочтительный — *preferable*
предыдущий — *previous, prev, pred*
при входе в — *enter*
при выходе из — *exit*
приближающийся — *approaching*
приблизительный — *approximate*
прибывающий — *arriving*
прибывший — *arrived*
придержанный — *suspended*
приложенный — *applied*
применимый — *applicable*
принятый — *accepted*
приобретённый — *purchased*

присоединённый — <i>appended</i>	старый — <i>old</i>
присутствующий — <i>present</i>	стёртый — <i>erased</i>
присущий — <i>native</i>	стремящийся — <i>striving</i>
приятный — <i>nice</i>	суммарный — <i>total</i>
проверенный — <i>proved</i>	существенный — <i>essential</i>
проданный — <i>sold</i>	существующий — <i>existent</i>
проигравший — <i>loser</i>	съеденный — <i>eaten</i>
промежуточный — <i>interim</i>	сырой — <i>raw</i>
пропавший — <i>lost</i>	текущий — <i>current</i>
пропущенный — <i>dropped</i>	типичный — <i>typical</i>
простой — <i>simple</i>	толстый — <i>thick</i>
противоположный — <i>opposite</i>	тонкий — <i>thin</i>
противоречивый — <i>conflict</i>	точное соответствие — <i>match</i>
прочитанный — <i>read</i>	точный — <i>exact</i>
пустой — <i>empty</i>	требуемый — <i>required</i>
работающий — <i>working</i>	тщательный — <i>thorough</i>
рабочий — <i>work</i>	тяжёлый (вес) — <i>heavy</i>
разблокированный — <i>unlocked</i>	тяжёлый (трудный) — <i>difficult</i>
разделяемый — <i>shared</i>	увеличенный — <i>increased</i>
разреженный — <i>rarefied</i>	увеличивающийся — <i>increasing</i>
разрешённый — <i>permitted</i>	уверенный — <i>certain</i>
разрешившийся — <i>resolved</i>	угловой — <i>angular</i>
распределённый — <i>distributed</i>	удалённый — <i>removed</i>
растущий — <i>growing</i>	удаляющийся — <i>removing</i>
расширенный — <i>expanded</i>	удачный — <i>lucky</i>
расширяющийся — <i>expanding</i>	улучшенный — <i>improved</i>
редкий — <i>rare</i>	уменьшающийся — <i>decreasing</i>
результатирующий — <i>result</i>	уменьшенный — <i>reduced</i>
решённый — <i>solved</i>	уникальный — <i>unique</i>
родительский — <i>parent, parental</i>	упорядоченный — <i>arranged</i>
ручной — <i>manual</i>	ускоренный — <i>rapid</i>
свободный — <i>free</i>	ускоряющийся — <i>accelerating</i>
сделанный — <i>done</i>	условный — <i>conditional</i>
сильный — <i>strong</i>	успешный — <i>successful</i>
сканированный — <i>scanned</i>	устаревший — <i>obsolete</i>
складированный — <i>stored</i>	устраивающий (годный) — <i>suitable</i>
скорый — <i>quick</i>	ухудшенный — <i>worsen</i>
скрытый — <i>hidden</i>	фактический — <i>actual</i>
слабый — <i>weak</i>	фиксированный — <i>fixed</i>
следующий — <i>next</i>	хороший — <i>fine</i>
сложный — <i>complex</i>	целевой — <i>target</i>
случайный — <i>random</i>	частичный — <i>partial</i>
случающийся — <i>accidental</i>	частый — <i>frequent</i>
сменный (диск) — <i>removable</i>	чёткий — <i>definite</i>
смутный — <i>vague</i>	чётный — <i>even</i>
собранный — <i>collected</i>	численный — <i>numerical</i>
собственный — <i>own</i>	числовой — <i>numeric</i>
сомнительный — <i>doubt</i>	чистый — <i>clean</i>
сортированный — <i>sorted</i>	чувствительный — <i>sensitive</i>
сохранённый — <i>saved</i>	широкий — <i>wide</i>
спорный — <i>arguable</i>	широкий, широковещательный — <i>broad</i>
спускающийся — <i>descending</i>	штрафной — <i>penalty</i>
среднесуммарный — <i>average</i>	экстремальный — <i>extreme</i>
старший — <i>senior</i>	явный — <i>explicit</i>

Действия и глаголы

авторизовать — authorize	задержать — delay
активировать — activate	задержка — delay
блокировать — lock	заказать — order
внедрение — implementation	закончить — terminate
внедрить — implement	закрепить — fix
возбудить (ошибку) — raise	закрепление (фиксация) — fixing
возвратить (отмененное) — redo	закрытие — closing
возвратить (память) — deallocate	закрывать — close
возобновить — resume	замена — replacing
войти (в систему) — login	заменить — replace
восстановить — recover	замкнуть — close
восстановление — recovering	заморозить (зафиксировать) — freeze
вписать в — fit	замыкание — closing
вписывание — fitting	записать — record
вращать — rotate	записывание — recording
вращение — rotation	запись — writing
вставить — insert	заполнение — filling
вставка — insertion	заполнение (массива) — populating
выбор — selection	заполнить (массив) — populate
выбрать — choose	заполнить, наполнить — fill
выгрузить — unload	запрос — request
выгрузка — unloading	запросить — query
выделение — selection	запустить (программу) — run
выделить — select	захват — capture
выйти (из системы) — logout	захватить — capture
выполнение (программы) — execution	зашифровать — encrypt
выполнить (программу) — execute	зашифровка — encrypting
выравнивание — aligning	защита — protection
вырезание — cutting	защитить — protect
вырезать — cut	идентификация — identification
выровнять — align	идентифицировать — identify
выровнять (строку) — justify	извлечение — extraction
вытереть (стереть инф.) — wipe	извлечение информации — retrieving
вытолкнуть (из стека) — pop	извлечь (из) — extract
вычесть — subtract	извлечь (что) — eject
вычитание — subtraction	извлечь информацию — retrieve
дать разрешение — grant	изменение размера — resizing
деактивировать — deactivate	изменение размерности — redimension
декодирование — decoding	изменить — modify
декодировать — decode	изменить размер — resize
демонтировать — dismount	изменить размерность — redim
дешифрация — decrypting	измерение — measurement
дешифровать — decrypt	измерить — measure
добавить — add	инверсия — inversion
добавление — adding	инвертировать — invert
доказательство — proving	инициализация — initialization
доказать — prove	инициализировать — initialize
допуск — admittance	инициировать — invoke
допустить — admit	искать (данные) — search
ждать — wait	искать (позицию) — seek
завершение — completion	исключение — exception
завершить — complete	исключение (недопущение) — exclusion
загрузить — load	исключить (что) — except
загрузка — loading	исключить из — exclude

кодировать — *encode*
компилировать — *compile*
компиляция — *compilation*
копировать — *copy*
котирование — *rating*
лишение правильности — *invalidation*
лишить правильности — *invalidate*
ловить, поймать — *trap*
максимизировать — *maximize*
маскирование — *masking*
маскировать — *mask*
масштабирование — *scaling*
масштабировать — *zoom*
менять, обменивать — *swap*
мигать, мерцать — *blink*
минимизировать — *minimize*
монтировать — *mount*
наблюдать — *watch*
наблюдение — *watching*
нажать, надавить — *press*
назначить, связать — *assign*
найти — *find*
накапливать — *accumulate*
накопление — *accumulation*
напомнить — *remind*
направить — *direct*
направлять — *guide*
настроить вид — *customize*
настроить конфигурацию — *configure*
нормализовать — *normalize*
обеспечить — *provide*
обменивать — *exchange*
обнаружение — *detection*
обнаружить — *detect*
обновить — *update*
обновление — *refresh*
обнулить — *zero*
обработать — *handle*
обработка — *handling*
обрезать — *clip*
обрезка — *clipping*
обслуживание — *service*
обслужить — *serve*
объединение — *join*
объединение — *union*
объединить — *unite*
объединить, собрать — *combine*
объединить, соединить — *join*
ограничение — *restriction*
ограничить — *limit*
ожидание — *awaiting*
оказать содействие (помочь) — *assist*
округлить — *round*
определение (задание) — *definition*
определить (задать) — *define*
оптимизация — *optimization*

оптимизировать — *optimize*
опубликовать — *publish*
опустошить — *empty*
освободить — *free*
освобождение — *release*
ответ — *answer*
ответить — *answer*
отказ — *refusal*
отказать — *refuse*
отключение — *disconnection*
отключить(ся) — *disconnect*
открепить — *detach*
открепление — *detachment*
открывание — *opening*
открыть — *open*
отложить — *defer*
отмена — *cancellation*
отменить — *cancel*
отменить (действие) — *undo*
отображение (состояния) — *indication*
отображение (отношения) — *mapping*
отобразить (состояние) — *indicate*
отобразить (отношение) — *map*
оторвать — *tear*
отослать, направить — *refer*
отпустить — *release*
отражение — *reflection*
отразить — *reflect*
отрезание — *trimming*
отрезать — *trim*
отслеживание — *tracking*
отслеживать путь — *track*
охранять — *guard*
оценить — *estimate*
оценить, вычислить — *evaluate*
оценить, котировать — *rate*
оценка — *estimation*
очистить — *clear*
очистка — *cleaning*
перевернуть — *reverse*
перевести — *translate*
перевод — *translation*
переключить — *toggle*
переместить — *move*
перемещение — *moving*
перенаправить — *redirect*
перенаправление — *redirection*
переопределение — *redefinition*
переопределить — *redefine*
перерисовать — *redraw*
переуступить, передать — *negotiate*
перечисление — *enumeration*
перечислить — *enumerate*
перечислить, вывести список — *list*
печатать (клавиатура) — *type*
печатать (принтер) — *print*

печатать (клавиатура) — *typing*
печатать (принтер) — *printing*
писать — *write*
планировать — *plan*
повернуть — *turn*
поворот — *turning*
повторить, повторять — *repeat*
подготовить — *prepare*
подготовка — *preparation*
поддерживать — *support*
поделить — *divide*
подключение — *connection*
подключить(ся) — *connect*
подписать — *sign*
подрезать (края) — *crop*
подсветить, акцентировать — *highlight*
подстроить, регулировать — *adjust*
подтвердить правильность — *validate*
подтверждение правильн. — *validation*
поиск — *search*
показ — *exhibition*
показать — *show*
покинуть — *quit*
получение копии — *duplication*
получить — *get*
получить копию — *duplicate*
получить точную копию — *replicate*
поместить — *place*
пометить — *mark*
понижение — *reduction*
понизить — *reduce*
понизить (сигнал) — *attenuate*
построение — *building*
построить — *build*
предложение — *proposal*
предложить — *propose*
предупредить — *warn*
предупреждение — *warning*
преобразование — *transformation*
преобразовать — *transform*
прервать — *abort*
прерывание — *abortion*
преследовать — *pursue*
приближение — *approach*
приблизиться — *approach*
прикрепить — *attach*
применение — *application*
применить — *apply*
принятие — *accept*
принять — *accept*
присоединить — *append*
пробовать — *try*
пробудить — *awake*
проверить — *check*
проверить правильность — *verify*
проверка — *checking*

проверка правильности — *verification*
проверять (ревизовать) — *audit*
проводить — *conduct*
продолжать (начатое) — *keep*
продолжить — *continue*
проиграть (запись) — *playback*
производить, выпускать — *produce*
прокрутить (изображение) — *scroll*
проследить — *trace*
прослеживание — *tracing*
прослушивание — *listening*
прослушивать — *listen*
просматривать (бродить) — *browse*
просмотр (брожение) — *browsing*
протолкнуть (в стек) — *push*
прыгать, переходить — *jump*
разбить — *break*
разблокировать — *unlock*
разделить, расщепить — *split*
разделять с кем-л. — *share*
разместить — *dispose*
размещение — *disposition*
разобрать (выражение) — *parse*
разрешить — *permit*
разрушение — *destruction*
разрушить — *destroy*
распаковать — *unpack*
распознавание — *recognition*
распознать — *recognize*
расположение — *locating*
расположить — *locate*
распределить (направить) — *dispatch*
распределить память — *allocate*
распространить — *distribute*
распространить (ошибку) — *propagate*
растянуть — *stretch*
расширить — *widen*
расширить, удлинить — *extend*
реализация — *realization*
реализовать — *realize*
регистрировать — *register*
редактировать — *edit*
решение — *solution*
решить (уравнение) — *solve*
рисование — *drawing*
рисовать — *draw*
рисовать, красить — *paint*
сбросить (на диск) — *flush*
сбросить, обнулить — *reset*
связать (в цепь) — *link*
связать, прикрепить — *bind*
связывание — *binding*
сдвиг — *shift*
сделать доступным — *enable*
сделать недоступным — *disable*
сделать плоским — *flatten*

сделать предположение — guess
сделать ход — move
сигнализировать (ошибку) — alert
сигнализировать — beep
сканирование — scanning
сканировать — scan
склеить — paste
скрывание — hiding
следить — monitor
слить — merge
смешивание — mixing
смешивать — mix
снабжать — supply
собирать — collect
собрать — assemble
советовать — advise
согласиться — agree
создание — creation
создать — create
сообщить — report
сортировать — sort
сохранить — save
сохранить, складировать — store
спать, быть пассивным — sleep
спросить — ask
спрятать, скрыть — hide
сравнение — comparison
сравнить — compare
стереть — erase
сформировать — form
схватить — catch

Разные вещи (существительные)

автомат (машина) — ?machine, ?box
адрес — address
анализ — analysis
анализатор — analyzer
базис — base, basic
балл — point, mark
буква — letter
вариант — option
вес — weight
ветка — branch
вещь — thing, piece
вид, внешность — appearance
вид, изображение — view
владелец — owner
вопрос — question
вхождение — entry
выбор — choice
выигрыш — gain
высота — height
глубина — depth
голос — voice
гость — guest
граница (объекта) — border

сцепить (строки) — concatenate
тащить, перетаскивать — drag
тестировать — test
убрать — remove
уведомить — notify
увеличить — enlarge
удаление — removal
удалить — delete
удлинить — lengthen
удостоверить — assure
удостовериться — certify
уменьшить, снизить — reduce
уничтожить — kill
упаковать — pack
упорядочить — arrange
управлять — control
управляться — manage
усечение — truncation
усечь — truncate
установить — set
установка — setting
учить — teach
фильтрация — filtering
фильтровать — filter
форматировать — format
хэшировать — hash
чередовать — alternate
чертить — plot
чистить, очищать — clean
читать, считывать — read
чтение — reading

граница (масива) — bound
график — graph, chart
данные — data
дерево — tree
диаграмма — chart, diagram
диапазон — range
долг (задолженность) — debt
доля — portion, quota
допуск, отклонение — tolerance
единица изм. — unit
ёмкость, вместимость — capacity
загвоздка — trouble
заготовка — blank
задача — task
задержка — delay
зазор — gap
заключение, вывод — conclusion
запас — stock
запись — record
звук — sound
знак — sign
избыток, излишек — excess
инструмент — tool

интервал — *interval*
кадр — *frame*
канал — *channel*
каркас — *frame, framework*
карта, отношение — *map*
картинка, изображение — *image*
картинка, рисунок — *picture*
каталог — *dir, directory, folder*
клавиша — *key*
клетка таблицы — *cell*
ключ — *key*
ключ, подсказка — *clue, hint*
коллекция — *collection*
колонка таблицы — *col, column*
команда — *command, instruction*
комплект — *set, pack*
компьютер — *computer*
конструктор, мастер — *designer*
копия — *copy*
корень — *root*
коэффициент — *factor*
краска — *dye*
куча — *heap, pile*
лист (дерева) — *leaf*
лист (книги) — *sheet*
маска — *mask*
масштаб — *scale*
мотор, движитель — *engine*
мощность — *power, output*
набор символов — *charset*
намёк — *hint*
неуспех — *failure*
обработчик — *handler*
образец — *sample*
обратная величина — *reciprocal*
объём — *volute*
оператор (языка) — *statement*
отступ (строки) — *indent*
очередь — *queue*
очередь (ход) — *turn*
ошибка — *error, mistake*
пересечение — *intersection*
плотность — *density*
повреждение — *damage*
поле — *field*
половина — *half*
пользователь (программы) — *user*
попытка — *attempt*
порядок — *order*
потеря — *miss, loss*
правило — *rule*
предложение — *sentence*
пример — *example*
причина — *reason*

проход (компилятора) — *pass*
пустышка — *dummy*
путь — *path*
разделитель — *delimiter*
размер — *size*
размерность — *dimension*
расширение (файла) — *extension*
регистр (слова) — *case*
режим — *mode*
родитель — *parent*
семейство — *family*
сила — *force, strength*
символ — *symbol*
скорость — *speed, velocity*
слово — *word*
смещение — *offset*
событие — *event*
сосед — *neighbor*
сообщение — *message*
состояние — *condition, state*
стиль — *style, manner*
сторона — *side*
страница — *page*
строка таблицы — *row*
строка текста — *line*
счёт (в банке) — *account*
счёт (игры) — *score*
счетовод — *accountant*
счётчик (устройство) — *?meter*
таблица — *table*
текст — *text*
тело — *body*
тип — *type*
точка — *point*
точка (знак) — *dot*
угол — *corner*
угол (матем.) — *angle*
удача, успех — *success*
узел — *node*
указатель — *pointer*
условие — *condition*
установки — *settings*
хвост — *tail*
ход, шаг — *move*
цвет — *color*
цель — *target*
член — *member*
часть — *part*
шаблон — *template*
шаг — *step*
ширина — *width*
элемент — *element, item, unit*
ядро — *kernel, core*
язык (лингв.) — *language*

Примечание: другие существительные следует искать в других словарях.

Способы образования имени

Имя объекта должно отражать его назначение. Используется одно или два, реже три и больше слов. Множественное число имени существительного обычно образуется добавлением буквы **s** к концу слова.

Сочетать можно:

- ❑ существительное с существительным:
PagesCounter — счётчик страниц
PagesNumber, NumberOfPages — число страниц
PageNumber, NumberOfPage — номер страницы
- ❑ прилагательное с существительным:
EstimatedValue — оценочное значение
AwaitingPlayer — ожидающий игрок
FinalScore — результирующий счёт
- ❑ глагол с существительным:
EstimateValue — оценить значение
MoveFigure — передвинуть фигуру
CreateNode — создать узел
- ❑ существительное с глаголом:
NodeCreating — создание узла
FigureMove — движение фигуры

Глаголы и действия используются для именования функций, процедур, свойств и методов.

Функции именовываются одним или несколькими словами:

- Assist** — оказать содействие (помочь)
- AskUser** — спросить пользователя
- TurnToTheRight, TurnRight** — повернуть вправо

Функции, которые возвращают логическое значение, а также переменные логического типа, рекомендуется начинать со слова **Is**:

- IsDigit** — (это) цифра?
- IsInside** — (находится) внутри?
- IsWaiting** — ждущий?

Свойства и методы по возможности должны именоваться оди-
ночными словами:

- Appearance** — внешность (свойство)
- Move** — переместить (метод)

Свойства именовываются существительными, методы — глаголами и глаголами в сочетании с прилагательными:

- BorderStyle** — стиль границ (свойство)
- MoveNext** — переместить к следующему (метод)

Таблицы символов

Таблицы символов определяют числовые значения знаков. В колонках *Dec* приведены значения в десятичной системе счисления, а в колонках *Hex* — в шестнадцатеричной.

Таблица ASCII — первая половина (128 символов)

<i>Dec</i>	<i>Hex</i>	<i>Знак</i>	<i>Символ</i>	<i>Dec</i>	<i>Hex</i>	<i>Знак</i>	<i>Dec</i>	<i>Hex</i>	<i>Знак</i>	<i>Dec</i>	<i>Hex</i>	<i>Знак</i>
0	00		NUL	32	20	пробел	64	40	@	96	60	`
1	01	☺	SOH	33	21	!	65	41	A	97	61	a
2	02	☻	STX	34	22	"	66	42	B	98	62	b
3	03	♥	ETX	35	23	#	67	43	C	99	63	c
4	04	♦	EOT	36	24	\$	68	44	D	100	64	d
5	05	♣	ENQ	37	25	%	69	45	E	101	65	e
6	06	♠	ACK	38	26	&	70	46	F	102	66	f
7	07	▪	BEL	39	27	'	71	47	G	103	67	g
8	08	▣	BS	40	28	(72	48	H	104	68	h
9	09	○	HT	41	29)	73	49	I	105	69	i
10	0A	▣	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	♂	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	♀	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	♪	CR	45	2D	-	77	4D	M	109	6D	m
14	0E	♫	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	☼	SI	47	2F	/	79	4F	O	111	6F	o
16	10	▶	DLE	48	30	0	80	50	P	112	70	p
17	11	◀	DC1	49	31	1	81	51	Q	113	71	q
18	12	!!	DC2	50	32	2	82	52	R	114	72	r
19	13	↕	DC3	51	33	3	83	53	S	115	73	s
20	14	ℳ	DC4	52	34	4	84	54	T	116	74	t
21	15	§	NAK	53	35	5	85	55	U	117	75	u
22	16	—	SYN	54	36	6	86	56	V	118	76	v
23	17	↕	ETB	55	37	7	87	57	W	119	77	w
24	18	↑	CAN	56	38	8	88	58	X	120	78	x
25	19	↓	EM	57	39	9	89	59	Y	121	79	y
26	1A	→	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	←	ESC	59	3B	;	91	5B	[123	7B	{
28	1C	└	FS	60	3C	<	92	5C	\	124	7C	
29	1D	↔	GS	61	3D	=	93	5D]	125	7D	}
30	1E	▲	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	▼	US	63	3F	?	95	5F	_	127	7F	DEL

Таблица ASCII — вторая половина (MS-DOS)

Dec	Hex	Знак	Dec	Hex	Знак	Dec	Hex	Знак	Dec	Hex	Знак
128	80	А	160	A0	а	192	C0	┌	224	E0	р
129	81	Б	161	A1	б	193	C1	└	225	E1	с
130	82	В	162	A2	в	194	C2	┘	226	E2	т
131	83	Г	163	A3	г	195	C3	┐	227	E3	у
132	84	Д	164	A4	д	196	C4	-	228	E4	ф
133	85	Е	165	A5	е	197	C5	+	229	E5	ж
134	86	Ж	166	A6	ж	198	C6	┌	230	E6	ц
135	87	З	167	A7	з	199	C7	┐	231	E7	ч
136	88	И	168	A8	и	200	C8	└	232	E8	ш
137	89	Й	169	A9	й	201	C9	┘	233	E9	щ
138	8A	К	170	AA	к	202	CA	┌	234	EA	ъ
139	8B	Л	171	AB	л	203	CB	┐	235	EB	ы
140	8C	М	172	AC	м	204	CC	└	236	EC	ь
141	8D	Н	173	AD	н	205	CD	=	237	ED	э
142	8E	О	174	AE	о	206	CE	┐	238	EE	ю
143	8F	П	175	AF	п	207	CF	└	239	EF	я
144	90	Р	176	B0	░	208	D0	└	240	F0	Ё
145	91	С	177	B1	▒	209	D1	┘	241	F1	ё
146	92	Т	178	B2	▓	210	D2	┐	242	F2	Є
147	93	У	179	B3		211	D3	└	243	F3	е
148	94	Ф	180	B4	┘	212	D4	┌	244	F4	ї
149	95	Х	181	B5	┐	213	D5	└	245	F5	і
150	96	Ц	182	B6	┐	214	D6	┐	246	F6	ı
151	97	Ч	183	B7	┘	215	D7	┐	247	F7	ÿ
152	98	Ш	184	B8	┘	216	D8	┐	248	F8	ÿ
153	99	Щ	185	B9	┐	217	D9	┘	249	F9	°
154	9A	Ъ	186	BA		218	DA	┐	250	FA	•
155	9B	Ы	187	BB	┘	219	DB	█	251	FB	·
156	9C	Ь	188	BC	┘	220	DC	█	252	FC	v
157	9D	Э	189	BD	┘	221	DD	█	253	FD	№
158	9E	Ю	190	BE	┘	222	DE	█	254	FE	¤
159	9F	Я	191	BF	┘	223	DF	█	255	FF	

Таблица ASCII — вторая половина (Windows)

<i>Dec</i>	<i>Hex</i>	<i>Знак</i>	<i>Dec</i>	<i>Hex</i>	<i>Знак</i>	<i>Dec</i>	<i>Hex</i>	<i>Знак</i>	<i>Dec</i>	<i>Hex</i>	<i>Знак</i>
128	80	À	160	A0	à	192	C0	А	224	E0	а
129	81	Á	161	A1	á	193	C1	Б	225	E1	б
130	82	Â	162	A2	â	194	C2	В	226	E2	в
131	83	Ã	163	A3	ã	195	C3	Г	227	E3	г
132	84	Ä	164	A4	ä	196	C4	Д	228	E4	д
133	85	Å	165	A5	å	197	C5	Е	229	E5	е
134	86	Æ	166	A6	æ	198	C6	Ж	230	E6	ж
135	87	Ç	167	A7	ç	199	C7	З	231	E7	з
136	88	È	168	A8	è	200	C8	И	232	E8	и
137	89	É	169	A9	é	201	C9	Й	233	E9	й
138	8A	Ê	170	AA	ê	202	CA	К	234	EA	к
139	8B	Ë	171	AB	ë	203	CB	Л	235	EB	л
140	8C	Ì	172	AC	ì	204	CC	М	236	EC	м
141	8D	Í	173	AD	í	205	CD	Н	237	ED	н
142	8E	Î	174	AE	î	206	CE	О	238	EE	о
143	8F	Ï	175	AF	ï	207	CF	П	239	EF	п
144	90	Ð	176	B0		208	D0	Р	240	F0	р
145	91	Ñ	177	B1		209	D1	С	241	F1	с
146	92	Ò	178	B2		210	D2	Т	242	F2	т
147	93	Ó	179	B3		211	D3	У	243	F3	у
148	94	Ô	180	B4		212	D4	Ф	244	F4	ф
149	95	Õ	181	B5		213	D5	Х	245	F5	х
150	96	Ö	182	B6		214	D6	Ц	246	F6	ц
151	97	×	183	B7		215	D7	Ч	247	F7	ч
152	98	Ø	184	B8		216	D8	Ш	248	F8	ш
153	99	Ù	185	B9		217	D9	Щ	249	F9	щ
154	9A	Ú	186	BA		218	DA	Ъ	250	FA	ъ
155	9B	Û	187	BB		219	DB	Ы	251	FB	ы
156	9C	Ü	188	BC		220	DC	Ь	252	FC	ь
157	9D	Ý	189	BD		221	DD	Э	253	FD	э
158	9E	Ë	190	BE		222	DE	Ю	254	FE	ю
159	9F	ß	191	BF		223	DF	Я	255	FF	я