

# ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

## Лабораторная работа 1. Создание консольного приложения в среде Microsoft Visual Studio

**Цель работы:** изучить основные принципы создания консольных приложений на языке C# с использованием среды Microsoft Visual Studio: ввод и вывод данных, выполнение вычислений.

### Задание к работе

1. Изучить главы 1, 2, 3 учебника «Visual C# 2010: полный курс».
2. Изучить структуру консольного приложения на C#, основные возможности класса `System.Console` для организации ввода и вывода данных.
3. Изучить функции класса `System.Math` для выполнения математических вычислений.
4. Решить представленные задачи в соответствии с вариантом.

### Задача 1

Вычислить значения  $a$  и  $b$  по заданным вещественным числам  $x, y, z$ , используя доступные функции модуля `System.Math`, за минимальное количество операций и дополнительных переменных.

$$1. \quad a = \frac{2\sqrt[3]{\sin^2 x - 1.5 \cdot 10^{-3} y}}{\frac{1}{3} + e^{-2z}}$$

$$b = x - \frac{z^2}{5} + \frac{1}{3} + e^{-2z}$$

$$3. \quad a = \frac{\lg^2(1 + |\operatorname{tg} z|) + 0.95x}{1 + \sin^2\left(y - \frac{\pi}{5}\right) - 0.01x}$$

$$b = \lg^2(1 + |\operatorname{tg} z|) + 3$$

$$5. \quad a = 1.2 \left( \operatorname{tg} \frac{x}{7} - 1 \right)$$

$$b = \frac{y \operatorname{tg}^2 \frac{x}{7} + 5.8}{z - \sqrt[5]{x - 2.3}}$$

$$7. \quad a = \frac{\ln^3(1 + \cos|z - 1|) + 0.71y}{2y^x + 0.005y}$$

$$b = 2y^x + \ln^3(1 + \cos|z - 1|)$$

$$9. \quad a = \frac{3 \arcsin^2 x - \sqrt{\lg(1 + x^2)}}{3z - 0.00000071}$$

$$b = x \cos y - 0.8 \arcsin x + 0.83$$

$$2. \quad a = \left| \sqrt[3]{\frac{x}{y}} - 2y^{\frac{3}{2}} \right|$$

$$b = \frac{1 + e^{2z}}{\left| \sqrt[3]{\frac{x}{y}} - 2y^{\frac{3}{2}} \right| + 7.8}$$

$$4. \quad a = \frac{\sqrt[3]{x} - 2 \sin 3y}{1.7 + \left| \ln^3 \frac{z}{2} \right|}$$

$$b = \frac{\sqrt[3]{x} - 2 \sin 3y}{2 \ln^3 \frac{z}{2}}$$

$$6. \quad a = \frac{8.01 \sin^2 3x - e^{-2z}}{\sqrt[3]{\lg y} + 2z}$$

$$b = \sqrt[3]{\lg y} - 5y$$

$$8. \quad a = \frac{\sqrt[5]{(x-7)^3} + 2 \operatorname{tg} 5y}{10 \log_2 z}$$

$$b = 3.7x - \sqrt[5]{(x-7)^3}$$

$$10. \quad a = \frac{\sqrt[4]{e^{3x}} + 0.8 \operatorname{tg}^2 2x}{\sqrt[3]{y-3} + 0.5 \operatorname{tg} 2x}$$

$$b = 1.35 \ln(\operatorname{tg}^2 2x) + 7.5z$$

$$11. \quad a = \frac{\lg(2 + \sin^2 x) + \sqrt[5]{3 - y}}{1 + 3z}$$

$$b = e^{-z} + \lg(2 + \sin^2 x)$$

$$13. \quad a = \frac{0,5 \operatorname{tg}^3 x - \sqrt[4]{x^{-zy}}}{y + 8 + 0,1 \operatorname{tg} x}$$

$$b = x^{-zy} - \sqrt[5]{(y + z)^4}$$

$$15. \quad a = \frac{\cos(x + 0,0002y) - \lg(8 + z)}{4z^5 - 4,0008}$$

$$b = \lg(\cos(x + 0,0002y)) - 11z^{\frac{3}{2}}$$

$$17. \quad a = \left| \sqrt[5]{x - z} - 5y^{3z} \right| + 0,000006$$

$$b = \frac{2 - e^{1,5z}}{\left| \sqrt[5]{x - z} - 5y^{3z} \right| + 3,002}$$

$$19. \quad a = \frac{\sqrt[3]{x + z} - 2 \cos 0,03y}{0,00007 + \left| \lg^3 \frac{z}{2} \right|}$$

$$b = \frac{\sqrt[5]{x} - 2 \cos 0,03y}{2 \lg^3 \frac{z}{2}}$$

$$12. \quad a = \frac{\lg(2 + e^x) + \sqrt[3]{8 + y}}{1 + \cos(2z - 1)}$$

$$b = \cos(2z - 1) + e^x$$

$$14. \quad a = \frac{\ln^3(1 + \cos|z - 1|) + 0,71y}{2x^{-3} + 0,005zy}$$

$$b = \ln^3(1 + \cos|z - 1|) + 5$$

$$16. \quad a = \frac{0,0004 \sqrt[3]{\cos^5 x} - 8,5 \cdot 10^{-3} z}{\frac{1}{8} + e^{x+y}}$$

$$b = x - \frac{z^2}{y - 5} + \frac{1}{8} + e^{x+y}$$

$$18. \quad a = \frac{\ln^3(2 + |\cos z|) + 0,00037x}{1 + \operatorname{tg}^2\left(y - \frac{\pi}{7}\right) + 2,05x}$$

$$b = \ln^3(2 + |\cos z|) + 8x$$

$$20. \quad a = 3,2 \left( \cos \frac{2x}{7} + 0,00001 \right)$$

$$b = \frac{y^z \cos \frac{2x}{7} + 8,71}{z + \sqrt[3]{x + y}}$$

## Задача 2

1. Из трёхзначного числа  $x$  вычли его последнюю цифру. Когда результат разделили на 10, а к частному слева приписали последнюю цифру числа  $x$ , то получилось число  $n$ . По заданному  $n$  найти число  $x$  (значение  $n$  вводится с клавиатуры,  $10 < n < 999$  и при этом число десятков в  $n$  не равно нулю).

2. В трёхзначном числе  $x$  зачеркнули первую цифру. Когда полученное число умножили на 10, а произведение сложили с первой цифрой числа  $x$ , то получилось число  $n$ . По заданному  $n$  найти число  $x$  (значение  $n$  вводится с клавиатуры,  $1 < n < 999$ ).

3. В трёхзначном числе  $x$  зачеркнули его вторую цифру. Когда к образованному при этом двузначному числу слева приписали вторую цифру числа  $x$ , то получилось число  $n$ . По заданному  $n$  найти число  $x$  (значение  $n$  вводится с клавиатуры,  $10 < n < 999$  и при этом число десятков в  $n$  не равно нулю).

4. В трёхзначном числе  $x$  зачеркнули его вторую цифру. Когда к образованному при этом двузначному числу справа приписали вторую цифру числа  $x$ , то получилось число  $n$ . По заданному  $n$  найти число  $x$  (значение  $n$  вводится с клавиатуры,  $100 < n < 999$ ).

5. В трёхзначном числе  $x$  зачеркнули его последнюю цифру. Когда в оставшемся двузначном числе переставили цифры, а затем приписали к ним слева последнюю цифру числа  $x$ , то получилось число  $n$ . По заданному  $n$  найти число  $x$  (значение  $n$  вводится с клавиатуры,  $1 < n < 999$  и при этом число единиц в  $n$  не равно нулю).

## Задача 3

1. Даны цифры двух целых чисел: двузначного  $a_2 a_1$  и однозначного  $b$ , где  $a_1$  — число единиц,  $a_2$  — число десятков. Получить цифры числа, равного сумме заданных чисел (известно, что это число двузначное). Слагаемое — двузначное число и число-результат не определять; условный оператор не использовать.

2. Даны цифры двух двузначных чисел, записываемых в виде  $a_2 a_1$  и  $b_2 b_1$ , где  $a_1$  и  $b_1$  — число единиц,  $a_2$  и  $b_2$  — число десятков. Получить цифры числа, равного сумме заданных чисел (известно, что это число

двузначное). Слагаемое — двузначное число и число-результат не определять; условный оператор не использовать.

3. Даны цифры двух десятичных целых чисел: трёхзначного  $a_3a_2a_1$  и двузначного  $b_2b_1$ , где  $a_1$  и  $b_1$  — число единиц,  $a_2$  и  $b_2$  — число десятков,  $a_3$  — число сотен. Получить цифры числа, равного сумме заданных чисел (известно, что это число трёхзначное). Числа-слагаемые и число-результат не определять; условный оператор не использовать.

4. Даны целое число  $k$  ( $1 \leq k \leq 180$ ) и последовательность цифр 10111213...9899, в которой выписаны подряд все двузначные числа. Определить:

- номер пары цифр, в которую входит  $k$ -я цифра;
- двузначное число, образованное парой цифр, в которую входит  $k$ -я цифра;
- $k$ -ю цифру.

5. Даны целое число  $k$  ( $1 \leq k \leq 150$ ) и последовательность цифр 101102103...149150, в которой выписаны подряд все трёхзначные числа от 101 до 150. Определить  $k$ -ю цифру.

6. Даны целые числа  $h, m, s$  ( $0 < h \leq 23, 0 \leq m \leq 59, 0 \leq s \leq 59$ ), указывающие момент времени:  $h$  часов,  $m$  минут,  $s$  секунд. Определить угол (в градусах) между положением часовой стрелки в начале суток и в указанный момент времени.

7. С начала суток часовая стрелка повернулась на  $y$  градусов ( $0 \leq y \leq 360, y$  — вещественное число). Определить число полных часов и число полных минут, прошедших с начала суток.

8. Часовая стрелка образует угол  $y$  с лучом, проходящим через центр и через точку, соответствующую 12 часам на циферблате,  $0 < y \leq 2\pi$ . Определить значение угла для минутной стрелки, а также количество полных часов и полных минут.

9. Даны целые числа  $h, m$  ( $0 < h \leq 12, 0 \leq m \leq 59$ ), указывающие момент времени:  $h$  часов,  $m$  минут. Определить наименьшее время (число полных минут), которое должно пройти до того момента, когда часовая и минутная стрелки на циферблате:

- совпадут;
- будут расположены перпендикулярно друг другу.

10. Даны два целых числа  $a$  и  $b$ . Если  $a$  делится на  $b$  или  $b$  делится на  $a$ , то вывести 1, иначе — 0. Условные операторы и операторы цикла не использовать.

#### Варианты заданий

№ варианта	Задача 1	Задача 2	Задача 3	№ варианта	Задача 1	Задача 2	Задача 3
1	1	1	1	11	11	3	1
2	2	2	2	12	12	4	2
3	3	3	3	13	13	5	3
4	4	4	4	14	14	1	4
5	5	5	5	15	15	2	5
6	6	2	6	16	16	4	6
7	7	3	7	17	17	5	7
8	8	4	8	18	18	1	8
9	9	5	9	19	19	2	9
10	10	1	10	20	20	3	10

## Лабораторная работа 2. Операторы ветвления и циклы

**Цель работы:** изучить особенности операторов ветвления и циклов в языке C#.

#### Задание к работе

- Изучить главу 4 учебника «Visual C# 2010: полный курс».
- Решить представленные задачи в соответствии с вариантом.

#### Задача 1

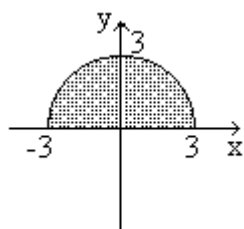
Написать программу, которая определяет:

1. максимальное значение для двух различных вещественных чисел;
2. является ли заданное целое число четным;
3. является ли заданное целое число нечетным;
4. если целое число  $M$  делится на целое число  $N$ , то на экран выводится частное от деления, в противном случае выводится сообщение « $M$  на  $N$  нацело не делится»;
5. оканчивается ли данное целое число цифрой 7;
6. имеет ли уравнение  $ax^2 + bx + c = 0$  решение, где  $a, b, c$  — данные вещественные числа;
7. какая из цифр двухзначного числа больше: первая или вторая;
8. одинаковы ли цифры данного двухзначного числа;
9. является ли сумма цифр двухзначного числа чётной;
10. является ли сумма цифр двухзначного числа нечётной;
11. кратна ли трём сумма цифр двухзначного числа;
12. кратна ли числу  $A$  сумма цифр двухзначного числа;
13. какая из цифр трёхзначного числа больше: первая или последняя;
14. какая из цифр трёхзначного числа больше: первая или вторая;
15. какая из цифр трёхзначного числа больше: вторая или последняя;
16. все ли цифры трёхзначного числа одинаковые;
17. существует ли треугольник с длинами сторон  $a, b, c$ ;
18. является ли треугольник с длинами сторон  $a, b, c$  прямоугольным;
19. является ли треугольник с длинами сторон  $a, b, c$  равнобедренным;
20. является ли треугольник с длинами сторон  $a, b, c$  равносторонним.

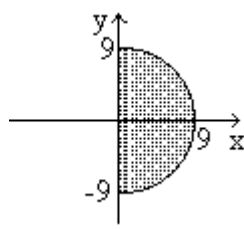
## Задача 2

Дана точка на плоскости с координатами  $(x, y)$ . Составить программу, которая определяет, лежит ли точка внутри заштрихованной области, вне заштрихованной области или на её границе.

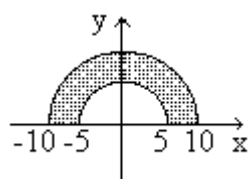
1.



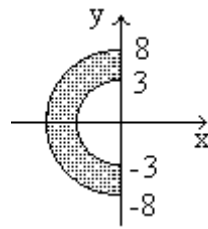
2.



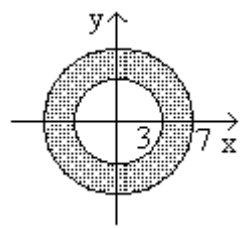
3.



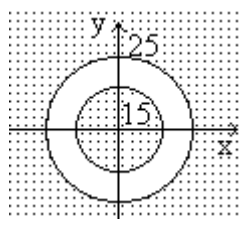
4.



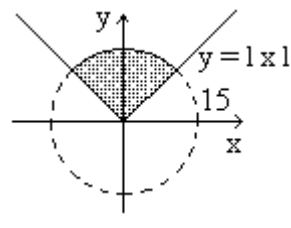
5.



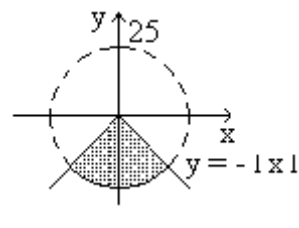
6.



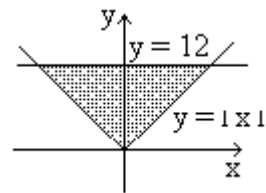
7.



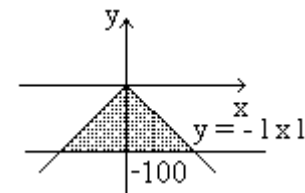
8.



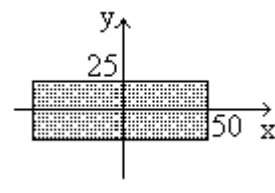
9.



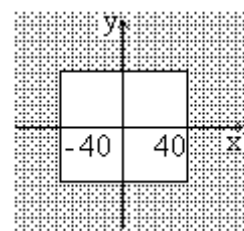
10.



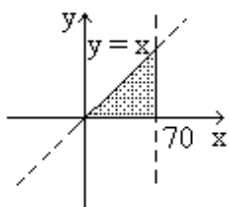
11.



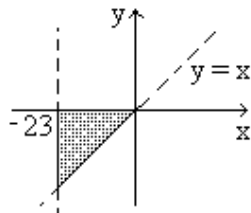
12.



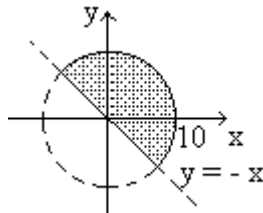
13.



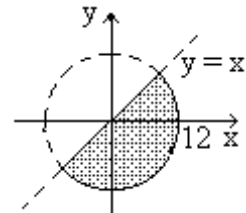
14.



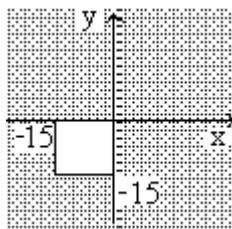
15.



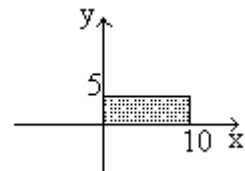
16.



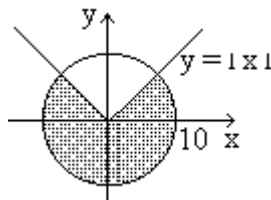
17.



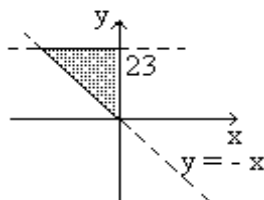
18.



19.



20.



### Задача 3

Вывести на экран:

1. целые числа 1, 3, 5, ..., 21 в строчку через пробел;
2. целые числа 10, 12, 14, ..., 60 в обратном порядке в столбик;
3. квадраты всех целых чисел от 10 до  $b$  (значение  $b$  вводится с клавиатуры,  $b > 10$ );
4. третьи степени всех целых чисел от  $a$  до 50 (значение  $a$  вводится с клавиатуры,  $a < 50$ );
5. таблицу соответствия между весом в фунтах и весом в килограммах для значений 1, 2, 3, ..., 10 фунтов (1 фунт = 453 г);
6. таблицу перевода 5, 10, 15, ..., 120 долларов США в рубли по текущему курсу (значение курса вводится с клавиатуры);
7. таблицу стоимости для 10, 20, 30, ..., 100 штук товара, при условии, что одна штука товара стоит  $x$  руб (значение  $x$  вводится с клавиатуры);
8. таблицу перевода расстояний в дюймах в сантиметры для значений 2, 4, 6, ..., 12 дюймов (1 дюйм = 25,4 мм);
9. кубы всех целых чисел из диапазона от  $A$  до  $B$  ( $A \leq B$ ) в обратном порядке;
10. все целые числа из диапазона от  $A$  до  $B$  ( $A \leq B$ ), оканчивающиеся на цифру  $X$ ;
11. все целые числа из диапазона от  $A$  до  $B$  ( $A \leq B$ ), оканчивающиеся на цифру  $X$  или  $Y$ ;
12. все целые числа из диапазона от  $A$  до  $B$  ( $A \leq B$ ), оканчивающиеся на любую чётную цифру;
13. только положительные целые числа из диапазона от  $A$  до  $B$  ( $A \leq B$ );
14. все целые числа из диапазона от  $A$  до  $B$  ( $A \leq B$ ), кратные трём;
15. все чётные числа из диапазона от  $A$  до  $B$  ( $A \leq B$ ), кратные трём;
16. только отрицательные чётные числа из диапазона от  $A$  до  $B$  ( $A \leq B$ );
17. все двухзначные числа, в записи которых все цифры разные;
18. все двухзначные числа, в которых старшая цифра отличается от младшей не больше чем на 1;
19. все трёхзначные числа, которые начинаются и заканчиваются на одну и ту же цифру;
20. все трёхзначные числа, в которых хотя бы две цифры повторяются.

## Лабораторная работа 3. Функции и массивы

**Цель работы:** изучить особенности использования функций и массивов в языке C#.

### Задание к работе

1. Изучить главы 5, 6 учебника «Visual C# 2010: полный курс».
2. Решить представленные задачи в соответствии с вариантом, оформив самостоятельные подзадачи в виде функций.

## Задача 1

Дана последовательность из  $n$  вещественных чисел.

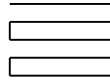
1. Подсчитать количество максимальных элементов.
2. Вывести на экран номера всех минимальных элементов.
3. Заменить все максимальные элементы нулями.
4. Заменить все минимальные элементы на противоположные.
5. Поменять местами максимальный элемент и первый.
6. Вывести на экран номера всех элементов, не совпадающих с максимальным.
7. Найти номер первого минимального элемента.
8. Найти номер последнего максимального элемента.
9. Подсчитать сумму элементов, расположенных между максимальным и минимальным элементами (минимальный и максимальный элементы в массиве единственные). Если максимальный элемент встречается позже минимального, то выдать сообщение об этом.
10. Найти номер первого максимального элемента.
11. Найти номер последнего минимального элемента.
12. Подсчитать сумму элементов, расположенных между первым максимальным и последним минимальными элементами. Если максимальный элемент встречается позже минимального, то выдать сообщение об этом.
13. Поменять местами первый минимальный и последний максимальный элементы.
14. Найти максимум из отрицательных элементов.
15. Найти минимум из положительных элементов.
16. Найти максимум из модулей элементов.
17. Найти количество пар соседних элементов, разность между которыми равна заданному числу.
18. Подсчитать количество элементов, значения которых больше значения предыдущего элемента.
19. Найти количество пар соседних элементов, в которых предыдущий элемент кратен последующему.
20. Найти количество пар соседних элементов, в которых предыдущий элемент меньше последующего.

## Задача 2

Дан массив  $A$  размером  $n \times n$ , элементами которого являются целые числа.

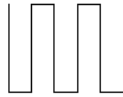
1. Подсчитать среднее арифметическое нечётных элементов, расположенных выше главной диагонали.
2. Подсчитать среднее арифметическое чётных элементов, расположенных ниже главной диагонали.
3. Подсчитать сумму элементов, расположенных на побочной диагонали.
4. Подсчитать среднее арифметическое ненулевых элементов, расположенных над побочной диагональю.
5. Подсчитать среднее арифметическое элементов, расположенных под побочной диагональю.
6. Поменять местами столбцы по правилу: первый с последним, второй с предпоследним и т.д.
7. Поменять местами две средних строки, если количество строк чётное, и первую со средней строкой, если количество строк нечётное.
8. Поменять местами два средних столбца, если количество столбцов чётное, и первый со средним столбцом, если количество столбцов нечётное.
9. Если количество строк в массиве чётное, то поменять строки местами по правилу: первую строку со второй, третью — с четвертой и т.д. Если количество строк в массиве нечётное, то оставить массив без изменений.
10. Если количество столбцов в массиве чётное, то поменять столбцы местами по правилу: первый столбец со вторым, третий — с четвёртым и т.д. Если количество столбцов в массиве нечётное, то оставить массив без изменений.
11. Вычислить  $A^k$ , где  $k$  — натуральное число.
12. Подсчитать норму матрицы по формуле  $\|A\| = \sum_i \max_j a_{i,j}$ .
13. Подсчитать норму матрицы по формуле  $\|A\| = \sum_j \max_i a_{i,j}$ .

14. Вывести элементы матрицы в следующем порядке:



15. Выяснить, является ли матрица симметричной относительно главной диагонали.

16. Заполнить матрицу числами от 1 до  $n$  (где  $n = m \times k$ ,  $m$  — количество строк,  $k$  — количество столбцов



прямоугольной матрицы) следующим образом:

17. Определить, есть ли в данном массиве строка, состоящая только из положительных элементов.

18. Определить, есть ли в данном массиве столбец, состоящий только из отрицательных элементов.

19. В каждой строке найти максимум и заменить его на противоположный элемент.

20. В каждом столбце найти минимум и заменить его нулём.

### Задача 3

Дан массив  $A$  размером  $n \times n$ , элементами которого являются целые числа.

1. Найти максимальный элемент в каждой строке и записать данные в новый массив.
2. Найти минимальный элемент в каждом столбце и записать данные в новый массив.
3. Чётные столбцы таблицы заменить на вектор  $X$ .
4. Нечётные строки таблицы заменить на вектор  $X$ .
5. Вычислить  $A \cdot X$ , где  $A$  — двумерная матрица,  $X$  — вектор.
6. Для каждой строки подсчитать количество положительных элементов и записать данные в новый массив.
7. Для каждого столбца подсчитать сумму отрицательных элементов и записать данные в новый массив.
8. Для каждого столбца подсчитать сумму чётных положительных элементов и записать данные в новый массив.
9. Для каждой строки подсчитать количество элементов, больших заданного числа, и записать данные в новый массив.
10. Для каждого столбца найти первый положительный элемент и записать данные в новый массив.
11. Для каждой строки найти последний чётный элемент и записать данные в новый массив.
12. Для каждого столбца найти номер последнего нечётного элемента и записать данные в новый массив.
13. Для каждой строки найти номер первого отрицательного элемента и записать данные в новый массив.
14. Для каждой строки найти сумму элементов с номерами от  $k_1$  до  $k_2$  и записать данные в новый массив.
15. Для каждого столбца найти произведение элементов с номерами от  $k_1$  до  $k_2$  и записать данные в новый массив.
16. Для каждой строки подсчитать сумму элементов, не попадающих в заданный интервал, и записать данные в новый массив.
17. Подсчитать сумму элементов каждой строки и записать данные в новый массив. Найти максимальный элемент нового массива.
18. Подсчитать произведение элементов каждого столбца и записать данные в новый массив. Найти минимальный элемент нового массива.
19. Для каждой строки найти номер первой пары неравных элементов. Данные записать в новый массив.
20. Для каждого столбца найти номер первой пары одинаковых элементов. Данные записать в новый массив.

## Лабораторная работа 4. Символы и строки

**Цель работы:** изучить особенности использования символьного типа `char` и класса для обработки строк `string`.

### Задание к работе

1. Изучить главы 5, 6 учебника «Visual C# 2010: полный курс».
2. Решить представленные задачи в соответствии с вариантом, используя любые доступные возможности класса `string`.

## Задача 1

Разработать программу, которая для заданной строки  $s$ :

1. вставляет символ  $x$  после каждого вхождения символа  $y$ ;
2. меняет местами первую букву со второй, третью с четвёртой и т.д.
3. определяет, какой из двух заданных символов встречается чаще в строке;
4. подсчитывает общее число вхождений символов  $x$  и  $y$ ;
5. подсчитывает количество букв в строке;
6. определяет, имеются ли в строке два соседствующих одинаковых символа;
7. удаляет среднюю букву, если длина строки нечётная, и две средних, если длина строки чётная;
8. удваивает каждое вхождение заданного символа  $x$ ;
9. удаляет все символы  $x$ ;
10. удаляет все подстроки *substr*;
11. заменяет все вхождения подстроки *substr1* на подстроку *substr2*;
12. подсчитывает сумму всех содержащихся в ней цифр;
13. подсчитывает количество содержащихся в ней цифр;
14. находит порядковые номера первого и последнего вхождения символа  $x$ ;
15. заменяет все группы стоящих рядом точек на многоточие;
16. выводит на экран последовательность символов, расположенных до первого двоеточия;
17. выводит на экран последовательность символов, расположенных после последнего двоеточия;
18. удаляет из неё последовательность символов, расположенных между круглыми скобками (считается, что в строке ровно одна пара круглых скобок).
19. удаляет из неё последовательность символов, расположенных между двумя запятыми (считается, что в строке ровно две запятые);
20. определяет, сколько различных символов встречается в строке.

## Задача 2

Дана строка, в которой содержится осмысленное текстовое сообщение. Слова сообщения разделяются пробелами и знаками препинания.

1. Вывести только те слова сообщения, в которых содержится заданная подстрока.
2. Вывести только те слова сообщения, которые содержат не более чем  $n$  букв.
3. Вывести только те слова сообщения, которые начинаются с прописной буквы.
4. Вывести только те слова сообщения, которые содержат хотя бы одну цифру.
5. Удалить из сообщения все слова, которые заканчиваются на заданный символ.
6. Удалить из сообщения все слова, содержащие данный символ (без учёта регистра).
7. Удалить из сообщения все однобуквенные слова (вместе с лишними пробелами).
8. Удалить из сообщения все повторяющиеся слова (без учёта регистра).
9. Подсчитать сколько раз заданное слово встречается в сообщении.
10. Подсчитать сколько слов, состоящих только из прописных букв, содержится в сообщении.
11. Найти самое длинное слово сообщения.
12. Найти все самые длинные слова сообщения.
13. Найти самое короткое слово сообщения.
14. Найти все самые короткие слова сообщения.
15. Вывести на экран все слова-палиндромы, содержащиеся в сообщении.
16. По правилу расстановки знаков препинания перед каждым знаком препинания пробел отсутствует, а после него обязательно стоит пробел. Учитывая данное правило, проверьте текст на правильность расстановки знаков препинания и, если необходимо, внесите в текст изменения.
17. Вывести только те слова, которые встречаются в тексте ровно один раз.
18. Вывести только те слова, которые встречаются более  $n$  раз.
19. Вывести слова сообщения в алфавитном порядке.
20. Вывести слова сообщения в порядке возрастания их длин.



## Лабораторная работа 5. Объектная декомпозиция

**Цель работы:** приобретение навыков выполнения объектной декомпозиции, выявления объектов и отношений между ними в заданной предметной области.

### Задание к работе

Для указанных заданий в соответствии с вариантом выполнить объектную декомпозицию, построить диаграмму взаимодействия объектов.

### Варианты заданий

1. Программа для игры в морской бой
2. Программа для игры в шахматы
3. Программа моделирования движения транспорта в населённом пункте
4. Программа для учёта книг в библиотеке
5. Программа для построения графиков функций
6. Программа для моделирования расстановки мебели в помещении
7. Программа для учёта успеваемости студентов в ВУЗе
8. Программа «Редактор графов»
9. Программа учёта денежных средств в банке
10. Программа для воспроизведения музыкальных файлов
11. Игра «Косынка»
12. Программа «Календарь» (учёт и планирования личного времени)
13. Программа учёта торговых операций
14. Игра «Червы»
15. Программа «Адресная книга»
16. Программа обмена сообщениями по локальной сети
17. Графический редактор
18. Программа составления спецификации на персональный компьютер
19. Игра «Сапёр»
20. Игра «Пинбол»

## Лабораторная работа 6. Создание класса

**Цель работы:** приобретение практических навыков создания класса на языке C#.

### Задание к работе

1. Изучить главы 8, 9, 10 учебника «Visual C# 2010: полный курс».
2. По заданному описанию класса в соответствии с вариантом создать класс на языке C#.

### Варианты заданий

1. Создать класс **Point**, разработав следующие элементы класса:
  - a. Поля:
    - `int x, y;`
  - b. Конструкторы, позволяющие создать экземпляр класса:
    - с нулевыми координатами;
    - с заданными координатами.
  - c. Методы, позволяющие:
    - вывести координаты точки на экран;
    - рассчитать расстояние от начала координат до точки;
    - переместить точку на плоскости на вектор (a, b).
  - d. Свойства:
    - получить-установить координаты точки (доступное для чтений и записи);
    - позволяющие умножить координаты точки на скаляр (доступное только для записи).
2. Создать класс **Triangle**, разработав следующие элементы класса:

- a. Поля:
    - `int` `a`, `b`, `c`;
  - b. Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.
  - c. Методы, позволяющие:
    - вывести длины сторон треугольника на экран;
    - рассчитать периметр треугольника;
    - рассчитать площадь треугольника.
  - d. Свойства:
    - позволяющее получить-установить длины сторон треугольника (доступное для чтения и записи);
    - позволяющее установить, существует ли треугольник с данными длинами сторон (доступное только для чтения).
3. Создать класс `Rectangle`, разработав следующие элементы класса:
- a. Поля:
    - `int` `a`, `b`;
  - b. Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.
  - c. Методы, позволяющие:
    - вывести длины сторон прямоугольника на экран;
    - рассчитать периметр прямоугольника;
    - рассчитать площадь прямоугольника.
  - d. Свойства:
    - получить-установить длины сторон прямоугольника (доступное для чтения и записи);
    - позволяющее установить, является ли данный прямоугольник квадратом (доступное только для чтения).
4. Создать класс `Money`, разработав следующие элементы класса:
- a. Поля:
    - `int` `first`; //номинал купюры
    - `int` `second`; //количество купюр
  - b. Конструктор, позволяющий создать экземпляр класса с заданными значениям полей.
  - c. Методы, позволяющие:
    - вывести номинал и количество купюр;
    - определить, хватит ли денежных средств на покупку товара на сумму  $N$  рублей.
    - определить, сколько шт товара стоимости  $n$  рублей можно купить на имеющиеся денежные средства.
  - d. Свойства:
    - позволяющее получить-установить значение полей (доступное для чтения и записи);
    - позволяющее рассчитать сумму денег (доступное только для чтения).
5. Создать класс для работы с одномерным массивом целых чисел. Разработать следующие элементы класса:
- a. Поля:
    - `int` `[]` `intArray`;
    - `int` `n`.
  - b. Конструктор, позволяющий создать массив размерности  $n$ :
  - c. Методы, позволяющие:
    - ввести элементы массива с клавиатуры;
    - вывести элементы массива на экран;
    - отсортировать элементы массива в порядке возрастания.
  - d. Свойства:
    - возвращающее размерность массива (доступное только для чтения);
    - позволяющее домножить все элементы массива на скаляр (доступное только для записи).
6. Создать класс для работы с двумерным массивом целых чисел. Разработать следующие элементы класса:
- a. Поля:
    - `int` `[,]` `intArray`;
    - `int` `n`.
  - b. Конструктор, позволяющий создать массив размерности  $n \times n$ .
  - c. Методы, позволяющие:

- ввести элементы массива с клавиатуры;
  - вывести элементы массива на экран;
  - вычислить сумму элементов  $i$ -того столбца.
- d. Свойства:
- позволяющее вычислить количество нулевых элементов в массиве (доступное только для чтения);
  - позволяющее установить значение всех элементов главной диагонали массива равное скаляр (доступное только для записи).
7. Создать класс для работы с двумерным массивом вещественных чисел. Разработать следующие элементы класса:
- a. Поля:
- `double [][] doubleArray;`
  - `int n, m.`
- b. Конструктор, позволяющий создать массив размерности  $n \times m$ .
- c. Методы, позволяющие:
- ввести элементы массива с клавиатуры;
  - вывести элементы массива на экран;
  - отсортировать элементы каждой строки массива в порядке убывания.
- d. Свойства:
- возвращающее общее количество элементов в массиве (доступное только для чтения);
  - позволяющее увеличить значение всех элементов массива на скаляр (доступное только для записи).
8. Создать класс для работы со строками. Разработать следующие элементы класса:
- a. Поля:
- `StringBuilder line;`
  - `int n.`
- b. Конструктор, позволяющий создать строку из  $n$  символов.
- c. Методы, позволяющие:
- подсчитать количество пробелов в строке;
  - заменить в строке все прописные символы на строчные;
  - удалить из строки все знаки препинания.
- d. Свойства:
- возвращающее общее количество элементов в строке (доступное только для чтения);
  - позволяющее установить значение поля, в соответствии с введенным значением строки с клавиатуры, а также получить значение данного поля (доступно для чтения и записи)
9. Создать класс для работы с регулярными выражениями. Разработать следующие элементы класса:
- a. Поля:
- `Regex r;`
  - `string text;`
- b. Методы, позволяющие:
- определить, содержит ли текст фрагменты, соответствующие шаблону поля;
  - вывести на экран все фрагменты текста, соответствующие шаблону поля;
  - удалить из текста все фрагменты, соответствующие шаблону поля;
- c. Свойства:
- позволяющее установить или получить строковое поле класса (доступно для чтения и записи)
  - позволяющее установить или получить регулярное выражение, хранящееся в соответствующем поле класса (доступно для чтения и записи)
10. Создать класс для работы с датой. Разработать следующие элементы класса:
- a. Поле `DateTime data.`
- b. Конструкторы, позволяющие установить:
- заданную дату;
  - дату 1.01.2009.
- c. Методы, позволяющие:
- вычислить дату предыдущего дня;
  - вычислить дату следующего дня;
  - определить сколько дней осталось до конца месяца.

d. Свойства:

- позволяющее установить или получить значение поле класса (доступно для чтения и записи);
- позволяющее определить год высокосным (доступно только для чтения).

## Лабораторная работа 7. Иерархии классов

**Цель работы:** приобретение практических навыков построения иерархии классов на языке C#.

### Задание к работе

1. Изучить главы 8, 9, 10 учебника «Visual C# 2010: полный курс».
2. По заданному описанию в соответствии с вариантом создать иерархию классов на языке C#.
3. Создать программу для решения поставленной задачи, использующую созданную структуру классов.

### Варианты заданий

#### Вариант 1

1. Создать абстрактный класс **Figure** с методами вычисления площади и периметра, а также методом, выводящим информацию о фигуре на экран.
2. Создать производные классы: **Rectangle** (прямоугольник), **Circle** (круг), **Triangle** (треугольник) со своими методами вычисления площади и периметра.
3. Создать массив  $n$  фигур и вывести полную информацию о фигурах на экран.

#### Вариант 2

1. Создать абстрактный класс **Function** с методом вычисления значения функции  $y = f(x)$  в заданной точке.
2. Создать производные классы: **Line** ( $y = ax + b$ ), **Kub** ( $y = ax^2 + bx + c$ ), **Hyperbola** ( $y = \frac{a}{x} + b$ ) со своими методами вычисления значения в заданной точке.
3. Создать массив  $n$  функций и вывести полную информацию о значении данных функций в точке  $x$ .

#### Вариант 3

1. Создать абстрактный класс **Издание** с методами, позволяющими вывести на экран информацию об издании, а также определить является ли данное издание искомым.
2. Создать производные классы: **Книга** (название, фамилия автора, год издания, издательство), **Статья** (название, фамилия автора, название журнала, его номер и год издания), **Электронный\_ресурс** (название, фамилия автора, ссылка, аннотация) со своими методами вывода информации на экран.
3. Создать каталог (массив) из  $n$  изданий, вывести полную информацию из каталога, а также организовать поиск изданий по фамилии автора.

#### Вариант 4

1. Создать абстрактный класс **Trans** с методами, позволяющими вывести на экран информацию о транспортном средстве, а также определить грузоподъемность транспортного средства.
2. Создать производные классы: **Легковая\_машина** (марка, номер, скорость, грузоподъемность), **Мотоцикл** (марка, номер, скорость, грузоподъемность, наличие коляски, при этом если коляска отсутствует, то грузоподъемность равна 0), **Грузовик** (марка, номер, скорость, грузоподъемность, наличие прицепа, при этом если есть прицеп, то грузоподъемность увеличивается в два раза) со своими методами вывода информации на экран, и определения грузоподъемности.
3. Создать базу (массив) из  $n$  машин, вывести полную информацию из базы на экран, а также организовать поиск машин, удовлетворяющих требованиям грузоподъемности.

#### Вариант 5

1. Создать абстрактный класс **Persona** с методами, позволяющим вывести на экран информацию о персоне, а также определить ее возраст (на момент текущей даты).

2. Создать производные классы: **Абитуриент** (фамилия, дата рождения, факультет), **Студент** (фамилия, дата рождения, факультет, курс), **Преподаватель** (фамилия, дата рождения, факультет, должность, стаж), со своими методами вывода информации на экран, и определения возраста.
3. Создать базу (массив) из  $n$  персон, вывести полную информацию из базы на экран, а также организовать поиск персон, чей возраст попадает в заданный диапазон.

#### Вариант 6

1. Создать абстрактный класс **Товар** с методами, позволяющим вывести на экран информацию о товаре, а также определить, соответствует ли она сроку годности на текущую дату.
2. Создать производные классы: **Продукт** (название, цена, дата производства, срок годности), **Партия** (название, цена, количество шт., дата производства, срок годности), **Комплект** (названия, цена, перечень продуктов) со своими методами вывода информации на экран, и определения соответствия сроку годности.
3. Создать базу (массив) из  $n$  товаров, вывести полную информацию из базы на экран, а также организовать поиск просроченного товара (на момент текущей даты).

#### Вариант 7

1. Создать абстрактный класс **Товар** с методами, позволяющими вывести на экран информацию о товаре, а также определить, соответствует ли она искомому типу.
2. Создать производные классы: **Игрушка** (название, цена, производитель, материал, возраст, на который рассчитана), **Книга** (название, автор, цена, издательство, возраст, на который рассчитана), **Спорт-инвентарь** (название, цена, производитель, возраст, на который рассчитана), со своими методами вывода информации на экран, и определения соответствия искомому типу.
3. Создать базу (массив) из  $n$  товаров, вывести полную информацию из базы на экран, а также организовать поиск товаров определённого типа.

#### Вариант 8

1. Создать абстрактный класс **Телефонный\_справочник** с методами, позволяющими вывести на экран информацию о записях в телефонном справочнике, а также определить соответствие записи критерию поиска.
2. Создать производные классы: **Персона** (фамилия, адрес, номер телефона), **Организация** (название, адрес, телефон, факс, контактное лицо), **Друг** (фамилия, адрес, номер телефона, дата рождения) со своими методами вывода информации на экран, и определения соответствия искомому типу.
3. Создать базу (массив) из  $n$  товаров, вывести полную информацию из базы на экран, а также организовать поиск в базе по фамилии.

#### Вариант 9

1. Создать абстрактный класс **Клиент** с методами, позволяющими вывести на экран информацию о клиентах банка, а также определить соответствие клиента критерию поиска.
2. Создать производные классы: **Вкладчик** (фамилия, дата открытия вклада, размер вклада, процент по вкладу), **Кредитор** (фамилия, дата выдачи кредита, размер кредита, процент по кредиту, остаток долга), **Организация** (название, дата открытия счёта, номер счёта, сумма на счёту) со своими методами вывода информации на экран, и определения соответствия дате (открытия вклада, выдачи кредита, открытия счёта).
3. Создать базу (массив) из  $n$  клиентов, вывести полную информацию из базы на экран, а также организовать поиск клиентов, начавших сотрудничать с банком в заданную дату.

#### Вариант 10

1. Создать абстрактный класс **Программное\_обеспечение** с методами, позволяющими вывести на экран информацию о программном обеспечении, а также определить соответствие возможности использования (на момент текущей даты).
2. Создать производные классы: **Свободное** (название, производитель), **Условно-бесплатное**

(название, производитель, дата установки, срок бесплатного использования), **Коммерческое** (название, производитель, цена, дата установки, срок использования) со своими методами вывода информации на экран, и определения возможности использования на текущую дату.

3. Создать базу (массив) из  $n$  видов программного обеспечения, вывести полную информацию из базы на экран, а также организовать поиск программного обеспечения, которое допустимо использовать на текущую дату.

## Лабораторная работа 8. Перегрузка операций

**Цель работы:** приобретение практических навыков реализации операций в классах на языке C#.

### Задание к работе

В класс, реализованный в лабораторной работе 6, добавить перегрузку указанных операций в соответствии с вариантом.

### Варианты заданий

1. В класс **Point** добавить перегрузку:
  - операции ++ (--): одновременно увеличивает (уменьшает) значение полей  $x$  и  $y$  на 1;
  - констант **true** и **false**: обращение к экземпляру класса даёт значение **true**, если значение полей  $x$  и  $y$  совпадает, иначе **false**;
  - операции бинарный +: одновременно добавляет к полям  $x$  и  $y$  значение скаляра;
  - преобразования типа **Point** в **string** (и наоборот).
2. В класс **Triangle** добавить перегрузку:
  - операции ++ (--): одновременно увеличивает (уменьшает) значение полей  $a$ ,  $b$  и  $c$  на 1;
  - констант **true** и **false**: обращение к экземпляру класса даёт значение **true**, если треугольник с заданными длинами сторон существует, иначе **false**;
  - операции \*: одновременно домножает поля  $a$ ,  $b$  и  $c$  на скаляр;
  - преобразования типа **Triangle** в **string** (и наоборот).
3. В класс **Rectangle** добавить перегрузку:
  - операции ++ (--): одновременно увеличивает (уменьшает) значение полей  $a$  и  $b$ ;
  - констант **true** и **false**: обращение к экземпляру класса даёт значение **true**, если прямоугольник с заданными длинами сторон является квадратом, иначе **false**;
  - операции \*: одновременно домножает поля  $a$  и  $b$  на скаляр;
  - преобразования типа **Rectangle** в **string** (и наоборот).
4. В класс **Money** добавить перегрузку:
  - операции ++ (--): одновременно увеличивает (уменьшает) значение полей  $first$  и  $second$ ;
  - операции !: возвращает значение **true**, если поле  $second$  не нулевое, иначе **false**;
  - операции бинарный +: добавляет к значению поля  $second$  значение скаляра;
  - преобразования типа **Money** в **string** (и наоборот).
5. Добавить в класс для работы с одномерным массивом целых чисел перегрузку:
  - операции ++ (--): одновременно увеличивает (уменьшает) значение всех элементов массива на 1;
  - операции !: возвращает значение **true**, если элементы массива не упорядочены по возрастанию, иначе **false**;
  - операции бинарный \*: домножить все элементы массива на скаляр;
  - преобразования класса массив в одномерный массив (и наоборот).
6. Добавить в класс для работы с двумерным массивом целых чисел перегрузку:
  - операции ++ (--): одновременно увеличивает (уменьшает) значение всех элементов массива на 1;
  - констант **true** и **false**: обращение к экземпляру класса даёт значение **true**, если двумерный массив является квадратным;
  - операции бинарный +: позволяющей сложить два массива соответствующих размерностей;
  - преобразования класса массив в двумерный массив (и наоборот).
7. Добавить в класс для работы с двумерным массивом вещественных чисел перегрузку:
  - операции ++ (--): одновременно увеличивает (уменьшает) значение всех элементов массива на 1;



- констант `true` и `false`: обращение к экземпляру класса даёт значение `true`, если каждая строка массива упорядочена по возрастанию, иначе `false`;
  - операции `*`: позволяющей умножить два массива соответствующих размерностей;
  - преобразования класса массив в ступенчатый массив (и наоборот).
8. Добавить в класс для работы со строками перегрузку:
- операции унарного `+` (`-`): преобразующей строку к строчным (прописным) символам;
  - констант `true` и `false`: обращение к экземпляру класса даёт значение `true`, если строка не пустая, иначе `false`;
  - операции `&`: возвращает значение `true`, если строковые поля двух объектов посимвольно равны (без учёта регистра), иначе `false`;
  - преобразования класса-строка в тип `string` (и наоборот).
9. Добавить в класс для работы с регулярными выражениями перегрузку:
- операции унарного `-`: удаляет из поля `text` все фрагменты, соответствующие регулярному выражению поля `r`;
  - констант `true` и `false`: обращение к экземпляру класса даёт значение `true`, если поле `text` не пустое, иначе `false`;
  - операции бинарного `+`: дописывает в конец поля `text` строку;
  - преобразования класса регулярного выражения в тип `string` (и наоборот).
10. Добавить в класс для работы с датой перегрузку:
- операции `!`: возвращает значение `true`, если установленная дата не является последним днём месяца, иначе `false`;
  - констант `true` и `false`: обращение к экземпляру класса даёт значение `true`, если установленная дата является началом года, иначе `false`;
  - операции `&`: возвращает значение `true`, если поля двух объектов равны, иначе `false`;
  - преобразования класса даты в тип `string` (и наоборот).

## Лабораторная работа 9. Создание элементов управления

**Цель работы:** приобретение практических навыков создания визуальных элементов интерфейса приложений Windows Forms.

### Задание к работе

Создать визуальный элемент Windows Forms, содержащий необходимые свойства, методы и события для решения поставленной задачи соответствии с вариантом. Выбрать наиболее подходящий способ создания компонента.

### Варианты заданий

1. Создать компонент, служащий для ввода в текстовое поле вещественных чисел в заданном диапазоне. Компонент должен содержать следующие свойства: диапазон допустимых значений, текущее значение, целая часть текущего значения, дробная часть текущего значения, строковое представление текущего значения, шаг изменения при нажатии клавиш вверх и вниз.
2. Создать компонент, содержащий кнопку, при нажатии на которую выполняется ввод заданной цифры в позицию курсора текстового поля или удаление символа, стоящего слева от курсора, если задано значение `-1`. Компонент должен содержать следующие свойства: цифра, которая должна вводиться при нажатии на кнопку (диапазон изменения от `-1` до `9`), текстовое поле для ввода цифры.
3. Создать компонент, выполняющий вывод текущего времени на заданный компонент. При потере приложением фокуса ввода таймер должен отключаться. Компонент должен содержать следующие свойства: компонент для вывода времени, формат для вывода времени, цвет текста.
4. Создать компонент, выполняющий запрет ввода заданных символов в многострочное текстовое поле, если установлен флажок. Компонент должен содержать следующие свойства: компонент для ввода текста, компонент, содержащий строку запрещённых символов. В случае отсутствия запрещённых символов флажок должен быть недоступен.
5. Создать компонент, отображающий результат заданной арифметической операции, если оба операнда введены без ошибок. Компонент должен содержать следующие свойства: два значения операндов, селектор для выбора операции.

## Лабораторная работа 10. Массивы и коллекции

**Цель работы:** приобретение практических навыков работы с типами массивов и коллекций в .NET Framework.

### Задание к работе

1. Создать класс, содержащий заданные в соответствии с вариантом имя и поля. Определить необходимые свойства для доступа к полям. Реализовать интерфейс `Comparable` и его обобщённый вариант.
2. Определить класс, основанный на структуре данных типа «односвязный линейный список», который позволяет хранить объекты заданного класса. Реализовать интерфейсы `IList`, `ICollection`, `IEnumerable` и их обобщённые варианты.
3. Добавить методы для добавления данных в список и удаления элементов из списка.
4. Добавить метод для поиска в соответствии с условием, представленным в задаче. Если элемент с заданным условием не найден, метод должен сгенерировать исключение.
5. Написать программу для решения поставленной задачи, используя созданные классы.

### Вариант 1

1. Определить класс с именем `Student`, содержащий следующие поля:

- фамилия и инициалы;
- номер группы;
- успеваемость (массив из пяти элементов).

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа `Student`; записи должны быть упорядочены по возрастанию номера группы;
- вывод на экран фамилий и номеров групп для всех студентов, включённых в массив, если средний балл студента больше 4.0;
- если таких студентов нет, вывести соответствующее сообщение.

### Вариант 2

1. Определить класс с именем `Student`, содержащий следующие поля:

- фамилия и инициалы;
- номер группы;
- успеваемость (массив из пяти элементов).

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа `Student`; записи должны быть упорядочены по возрастанию среднего балла;
- вывод на экран фамилий и номеров групп для всех студентов, имеющих оценки 4 и 5;
- если таких студентов нет, вывести соответствующее сообщение.

### Вариант 3

1. Определить класс с именем `Student`, содержащий следующие поля:

- фамилия и инициалы;
- номер группы;
- успеваемость (массив из пяти элементов).

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа `Student`; записи должны быть упорядочены по алфавиту;
- вывод на экран фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2;



- если таких студентов нет, вывести соответствующее сообщение.

#### Вариант 4

1. Определить класс с именем `Aeroflot`, содержащий следующие поля:

- название пункта назначения рейса;
- номер рейса;
- тип самолета.

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа `Aeroflot`; записи должны быть упорядочены по возрастанию номера рейса;
- вывод на экран номеров рейсов и типов самолётов, вылетающих в пункт назначения, название которого совпало с названием, введённым с клавиатуры;
- если таких рейсов нет, вывести соответствующее сообщение.

#### Вариант 5

1. Определить класс с именем `Aeroflot`, содержащий следующие поля:

- название пункта назначения рейса;
- номер рейса;
- тип самолёта.

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа `Aeroflot`; записи должны быть размещены в алфавитном порядке по названиям пунктов назначения;
- вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолётом, тип которого введён с клавиатуры;
- если таких рейсов нет, вывести соответствующее сообщение.

#### Вариант 6

1. Определить класс с именем `Worker`, содержащий следующие поля:

- фамилия и инициалы работника;
- название занимаемой должности;
- год поступления на работу.

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа `Worker`; записи должны быть размещены по алфавиту;
- вывод на экран фамилий работников, чей стаж работы в организации превышает значение, введённое с клавиатуры;
- если таких работников нет, вывести соответствующее сообщение.

#### Вариант 7

1. Определить класс с именем `Train`, содержащий следующие поля:

- название пункта назначения;
- номер поезда;
- время отправления.

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа `Train`; записи должны быть размещены в алфавитном порядке по названиям пунктов назначения;
- вывод на экран информации о поездах, отправляющихся после введенного с клавиатуры времени;
- если таких поездов нет, вывести соответствующее сообщение.

#### Вариант 8

1. Определить класс с именем `Train`, содержащий следующие поля:

- название пункта назначения;
- номер поезда;
- время отправления.

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа `Train`; записи должны быть упорядочены по времени отправления поезда;
- вывод на экран информации о поездах, направляющихся в пункт, название которого введено с клавиатуры;
- если таких поездов нет, вывести соответствующее сообщение.

#### Вариант 9

1. Определить класс с именем `Train`, содержащий следующие поля:

- название пункта назначения;
- номер поезда;
- время отправления.

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа `Train`; записи должны быть упорядочены по номерам поездов;
- вывод на экран информации о поезде, номер которого введен с клавиатуры;
- если таких поездов нет, вывести соответствующее сообщение.

#### Вариант 10

1. Определить класс с именем `Marsh`, содержащий следующие поля:

- название начального пункта маршрута;
- название конечного пункта маршрута;
- номер маршрута.

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми объектов типа `Marsh`; записи должны быть упорядочены по номерам маршрутов;
- вывод на экран информации о маршруте, номер которого введен с клавиатуры;
- если таких маршрутов нет, вывести соответствующее сообщение.

#### Вариант 11

1. Определить класс с именем `Marsh`, содержащий следующие поля:

- название начального пункта маршрута;
- название конечного пункта маршрута;
- номер маршрута.

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа `Marsh`; записи должны быть упорядочены по номерам маршрутов;
- вывод на экран информации о маршрутах, которые начинаются или кончаются в пункте, название которого введено с клавиатуры;
- если таких маршрутов нет, вывести соответствующее сообщение.

#### Вариант 12

1. Определить класс с именем `Note`, содержащий следующие поля:

- фамилия, имя;
- номер телефона;
- день рождения (массив из трёх чисел).

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа `Note`; записи должны быть упорядочены по датам дней рождения;
- вывод на экран информации о человеке, номер телефона которого введён с клавиатуры;
- если такого нет, вывести соответствующее сообщение.

#### Вариант 13

1. Определить класс с именем `Note`, содержащий следующие поля:

- фамилия, имя;
- номер телефона;
- день рождения (массив из трёх чисел).

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа `Note`; записи должны быть размещены по алфавиту;
- вывод на экран информации о людях, чьи дни рождения приходятся на месяц, значение которого введено с клавиатуры;
- если таких нет, вывести соответствующее сообщение.

#### Вариант 14

1. Определить класс с именем `Note`, содержащий следующие поля:

- фамилия, имя;
- номер телефона;
- день рождения (массив из трёх чисел).

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа `Note`; записи должны быть упорядочены по трём первым цифрам номера телефона;
- вывод на экран информации о человеке, чья фамилия введена с клавиатуры;
- если такого нет, вывести соответствующее сообщение.

#### Вариант 15

1. Определить класс с именем `Znak`, содержащий следующие поля:

- фамилия, имя;
- знак Зодиака;
- день рождения (массив из трёх чисел).

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа **Znak**; записи должны быть упорядочены по датам дней рождения;
- вывод на экран информации о человеке, чья фамилия введена с клавиатуры;
- если такого нет, вывести соответствующее сообщение.

#### Вариант 16

1. Определить класс с именем **Znak**, содержащий следующие поля:

- фамилия, имя;
- знак Зодиака;
- день рождения (массив из трёх чисел).

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа **Znak**; записи должны быть упорядочены по датам дней рождения;
- вывод на экран информации о людях, родившихся под знаком, наименование которого введено с клавиатуры;
- если таких нет, вывести соответствующее сообщение.

#### Вариант 17

1. Определить класс с именем **Znak**, содержащий следующие поля:

- фамилия, имя;
- знак Зодиака;
- день рождения (массив из трёх чисел).

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа **ZNAK**; записи должны быть упорядочены по знакам Зодиака;
- вывод на экран информации о людях, родившихся в месяце, значение которого введено с клавиатуры;
- если таких нет, вывести соответствующее сообщение.

#### Вариант 18

1. Определить класс с именем **Price**, содержащий следующие поля:

- название товара;
- название магазина, в котором продается товар;
- стоимость товара в рублях.

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа **Price**; записи должны быть размещены в алфавитном порядке по названиям товаров;
- вывод на экран информации о товаре, название которого введено с клавиатуры;
- если таких товаров нет, вывести соответствующее сообщение.

#### Вариант 19

1. Определить класс с именем **Price**, содержащий следующие поля:

- название товара;
- название магазина, в котором продается товар;
- стоимость товара в рублях.

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа **Price**; записи должны быть размещены в алфавитном порядке по названиям магазинов;
- вывод на экран информации о товарах, продающихся в магазине, название которого введено с клавиатуры;
- если такого магазина нет, вывести соответствующее сообщение.

#### Вариант 20

1. Определить класс с именем **Order**, содержащий следующие поля:

- расчётный счёт плательщика;
- расчётный счёт получателя;
- перечисляемая сумма в рублях.

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа **Order**; записи должны быть размещены в алфавитном порядке по расчётным счетам плательщиков;
- вывод на экран информации о сумме, снятой с расчётного счёта плательщика, введённого с клавиатуры;
- если такого расчётного счёта нет, вывести соответствующее сообщение.

## РАСЧЁТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ

### Моделирование информационного процесса средствами объектно-ориентированного программирования

#### Задание к работе

1. Для заданного в соответствии с вариантом описания информационного процесса выполнить объектно-ориентированный анализ и выделить основные типы объектов предметной области.
2. Реализовать выделенные типы объектов в виде классов, выбрав необходимые свойства, методы и события.
3. Разработать структуру приложения Windows Forms для моделирования предметной области. Для каждой формы приложения определить выполняемые функции и разработать алгоритмы их реализации, а также принципы и стратегии взаимодействия объектов.
4. Выполнить реализацию разработанных функций и алгоритмов в виде приложения Windows Forms. При проектировании интерфейса использовать элементы управления DevExpress или ComponentOne.
5. Сравнить функциональность выбранных элементов пользовательского интерфейса с возможностями аналогичных стандартных элементов, сделать выводы о возможности замены выбранных компонентов DevExpress или ComponentOne стандартными (входящими в состав .NET Framework).

#### Варианты заданий

##### Вариант 1

На каждой станции метро расположены информационные табло, которые работают под управлением единой информационной службы метро. На табло отображается текущее время (часы, минуты, секунды) и время, прошедшее с момента отправления последнего поезда (минуты, секунды). Момент прибытия и отправления поезда определяется при помощи датчиков, устанавливаемых на путях. Все табло в метро синхронизованы, текущее время отсчитывается и устанавливается из центральной службы времени, являющейся частью единой информационной службы метро.

На табло высвечивается конечная станция назначения прибывающего поезда. Эти данные содержатся в расписании движения поездов, которое хранится в памяти табло и периодически обновляется единой информационной службой метро.

В "бегущей строке" табло отображается рекламная информация. Память табло хранит до 10 рекламных сообщений. Сообщения отображаются друг за другом с небольшими паузами, циклически. Содержание рекламных сообщений поступает из единой информационной службы метро.

Дополнительная функция табло – по запросу единой информационной службы метро оно пересылает данные о нарушениях расписания (преждевременных отправлениях поездов или опозданиях).

В ходе выполнения задания должна быть создана схема базы данных для хранения рекламных сообщений, расписания и сведений о нарушении расписаний.

## Вариант 2

Банкомат – это автомат для выдачи наличных денег по кредитным пластиковым карточкам. В его состав входят следующие устройства: дисплей, панель управления с кнопками, приемник кредитных карт, хранилище денег и лоток для их выдачи, хранилище конфискованных кредитных карт, принтер для печати справок, сервисная консоль. Банкомат подключен к линии связи для обмена данных с банковской информационной системой, содержащей сведения о счетах клиентов.

Обслуживание клиента начинается с момента помещения пластиковой карточки в банкомат. После распознавания типа пластиковой карточки, банкомат выдает на дисплей приглашение ввести персональный код. Персональный код представляет собой четырехзначное число. Затем банкомат проверяет правильность введенного кода, сверяя с кодом, хранящимся на карте. Если код указан неверно, пользователю предоставляются еще две попытки для ввода правильного кода. В случае повторных неудач карта перемещается в хранилище карт, и сеанс обслуживания заканчивается. После ввода правильного кода банкомат предлагает пользователю выбрать операцию. Клиент может либо снять наличные со счета, либо узнать остаток на его счету, либо осуществить безналичный платеж со своего счёта.

При снятии наличных со счета банкомат предлагает указать сумму (100, 200, 500, 1000, 5000, 10000, 20000, 50000 рублей). После выбора клиентом суммы банкомат запрашивает, нужно ли печатать справку по операции. Затем банкомат посылает запрос на снятие выбранной суммы центральному компьютеру банка. В случае получения разрешения на операцию, банкомат проверяет, имеется ли требуемая сумма в его хранилище денег, и может ли он выдать запрошенную сумму банкнотами, имеющимися в наличии (например, если в банкомате нет банкнот достоинством менее 1000 рублей, он не может выдать никакую сумму не кратную 1000). Если банкомат может выдать деньги, то на дисплей выводится сообщение "Выньте карту". После удаления карточки из приемника, банкомат выдает указанную сумму в лоток выдачи. Банкомат печатает справку по произведенной операции, если она была затребована клиентом.

Если клиент хочет узнать остаток на счету, то банкомат посылает запрос центральному компьютеру банка и выводит сумму на дисплей. По требованию клиента печатается и выдается соответствующая справка.

Если клиент хочет осуществить безналичный платеж, то банкомат запрашивает у него банковские реквизиты получателя платежа и сумму платежа. Затем банкомат посылает запрос банковскому компьютеру на совершение платежа. Варианты ответа на такой тип запроса следующие: платеж осуществлён; ошибка в реквизитах получателя платежа; недостаточно денег на счету клиента для осуществления платежа. В случае успешного совершения платежа, на дисплей выводится сообщение "Выньте карту". После удаления карточки из приемника, банкомат печатает справку по произведенной операции, если она была затребована клиентом. Сервисная консоль, которая используется обслуживающим персоналом, находится в специальном отделении банкомата, закрываемом на замок. С консоли производится управление доступом к хранилищу денег для загрузки банкнот, управление доступом к хранилищу конфискованных карт, конфигурация сетевого соединения с банковским компьютером и проверка его работоспособности.

## Вариант 3

Магазин компьютеров предлагает возможность приобретения своих товаров через Интернет. Клиент может выбрать компьютер на веб-странице магазина. Компьютеры подразделяются на серверы, настольные, портативные и карманные. Заказчик может выбрать стандартную конфигурацию из списка и детально ознакомиться с ней на отдельной веб-странице. Если стандартная конфигурация ему не подходит, он может построить требуемую ему конфигурацию в диалоговом режиме. Компоненты конфигурации (такие, как оперативная память, процессор, жесткий диск и т. п.) представляются как список для выбора из доступных альтернатив. Для каждой новой конфигурации система может подсчитать цену.

Чтобы оформить заказ, клиент должен заполнить электронную форму с адресами для доставки товара и отправки счета-фактуры, а также деталями, касающимися оплаты. Оплата компьютеров осуществляется наличными курьеру, осуществляющему доставку, или банковским переводом на счет Интернет-магазина. После ввода заказа система отправляет клиенту по электронной почте сообщение с подтверждением получения заказа вместе с относящимися к нему деталями (стоимость, номер счета, банковские реквизиты для безналичной оплаты и т. п.). Пока клиент ожидает прибытия компьютера, он может проверить состояние заказа (поставлен в очередь / собран / отправлен). Работник магазина проверяет, поступила ли оплата (в случае безналичного расчета) и делает соответствующую пометку при поступлении денег. Если деньги не поступают в течение 5 банковских дней, заказ аннулируется. После оплаты или в случае оплаты наличными работник печатает счет-фактуру и отправляет ее на склад вместе с требованием заказанной конфигурации. Заказ помечается как поставленный в очередь. Собранный компьютер вместе со счетом-фактурой и накладной передается со склада в отдел доставки, при этом заказ помечается как собранный. Компьютер поставляется клиенту (статус заказа – отправлен). Если заказ оплачивается наличными, курьер по возвращении передает деньги в кассу, заказ помечается как оплаченный.

По окончании работы с заказом, он помечается в системе как выполненный.

#### Вариант 4

Каждый блог принадлежит одному пользователю и состоит из нескольких записей. В момент создания блога в нем записей нет. Пользователь может создавать записи лишь в собственном блоге. Читать записи любого пользователя может каждый пользователь. В каждой записи есть заголовок, дата, текст. На каждой странице блога находятся ссылка на профиль. На начальной странице блога отображаются 10 последних записей (или менее, если в блоге их недостаточно). Если записей в блоге больше 10, то с начальной страницы можно перейти на вторую, где отображается второй десяток записей и т. д. Записи в блоге упорядочены по убыванию даты. Любая запись может быть отредактирована, но дата записи не может быть изменена. Запись может быть удалена автором.

Перейдя на страницу отдельной записи, пользователь может ее комментировать, независимо от того, чужая это запись или его собственная. Комментарий состоит из текста и даты, указывающей, когда он написан. Комментарий может быть дан к записи блога или к ранее написанному комментарию. Пользователь может запретить комментирование любой записи в своем блоге. Комментарии, полученные раньше запрета, сохраняются. Запрет на комментирование может быть снят. Все комментарии к записи отображаются под ее текстом в виде дерева. Каждый комментарий помечен именем его автора. Имя может быть использовано для перехода к начальной странице блога с этим именем.

Комментарии не могут быть отредактированы, но могут быть удалены автором записи. При удалении комментария также стираются все ответы на него, если таковые были.

Любой желающий вести блог должен зарегистрироваться. При регистрации пользователь заполняет форму, указывая желаемое имя, пароль, список своих интересов, дату рождения, контакты, краткие сведения о себе. Если имя не занято, и форма заполнена полностью, система заводит блог нового пользователя.

Сведения из регистрационной формы (кроме пароля) отображаются на странице профиля пользователя. Пользователь может отредактировать собственный профиль, изменив в нем все, кроме имени. Пользователь может изменить свой пароль. Пароль не может быть короче 8 символов.

Зарегистрированный пользователь, осуществляет вход в собственный блог, указывая имя и пароль. Все описанные выше действия с записями, комментариями и собственным профилем, действия с друзьями доступны пользователю только после входа.

Пользователь может полностью удалить свой блог со всеми записями, комментариями и профилем. После удаления блога регистрационные данные стираются из системы.

#### Вариант 5

В каталоге хранится следующая информация о ресурсах: название ресурса, уникальный локатор ресурса (URL), раздел каталога, в котором содержится ресурс, список ключевых слов, краткое описание, дата последнего обновления, контактная информация.

Доступ пользователей к каталогу осуществляется при помощи браузера. Пользователи каталога могут добавлять новые ресурсы, информация о которых не была внесена ранее. Ресурсы в каталоге классифицируются по разделам. Полный список ресурсов каждого раздела должен быть доступен

пользователям. Количество ресурсов в разделе может быть большим, поэтому пользователь может выбрать количество, отображаемое на одной странице, например 25, тогда на первой странице раздела отображается список из первых 25 ресурсов, на второй – следующие 25 и т. д. Ресурсы в списке могут быть упорядочены по дате обновления или по названиям (по алфавиту).

Пользователям каталога должны быть предоставлены возможности по поиску ресурсов в каталоге. Поиск осуществляется по ключевым словам. Если пользователь не доволен результатами поиска, он может уточнить запрос (осуществить поиск среди результатов предыдущего поиска). Должна быть возможность выдавать результаты поиска в разной форме (вывод всей информации о ресурсах или частичной). Пользователь может отсортировать результаты поиска по релевантности (соответствию ключевым словам из запроса) или по дате обновления.

Поскольку содержание ресурсов Интернет со временем изменяется необходимо следить за датой последнего обновления, периодически опрашивая веб-сайты, URL которых хранятся в каталоге.

#### Вариант 6

На продуктовом складе для каждого товара фиксируется место хранения (определенная полка), количество товара и его наименование. Разные товары имеют разные единицы измерения: штуки, килограммы, коробки, бутылки и др. Система учета товаров должна обеспечивать добавление информации о новом товаре, изменение или удаление информации об имеющемся товаре, хранение (добавление, изменение и удаление) информации о поставщиках и покупателях, включающей в себя название фирмы, ее адрес и телефон. В системе учитывается приход товаров от поставщиков. В каждом приходе товаров могут содержаться несколько позиций, в каждой позиции указываются наименование товара и его количество. После оформления прихода товара в системе количество товара в инвентаризационной описи соответственно увеличивается.

Товар со склада отпускается покупателям по расходным накладным. В каждой накладной могут содержаться несколько позиций, в каждой позиции указываются наименование товара и его количество. После оформления расхода товара в системе количество товара в инвентаризационной описи соответственно уменьшается. Количество расходуемого товара не может превысить общее количество этого товара на складе.

Система учета по требованию пользователя формирует и выдает на печать следующую справочную информацию:

- список всех товаров;
- инвентаризационную опись товаров, имеющихся в наличии;
- список товаров, количество которых необходимо пополнить;
- список товаров, поставляемых данным поставщиком;
- все позиции в каком-либо приходе товара;
- все позиции в какой-либо расходной накладной.

Система осуществляет поиск информации о клиенте или поставщике по части названия фирмы. Это необходимо, чтобы работники склада могли связаться с фирмой по какому-либо вопросу.

#### Вариант 7

Система обеспечивает составление расписания некоторого учебного заведения, внесение в него изменений, выдачу полного расписания и дополнительной информации (например, по итоговому расписанию составляется расписание указанной группы или указанного преподавателя на заданный день или неделю).

В расписании фиксируются время и место проведения занятия, предмет и преподаватель, проводящий занятие, а также номер группы, для которой это занятие проводится. Некоторые занятия проводятся для нескольких групп: потока или всего курса.

Расписание не должно содержать коллизий (например, разные занятия не должны пересекаться друг с другом по месту и времени их проведения, один преподаватель не может вести одновременно два разных занятия, в одно и то же время, у одной и той же группы не допускается одновременно два различных занятия и т. д.).

Система по требованию пользователя должна осуществлять поиск свободных аудиторий на заданную дату и время.



## Вариант 8

Торговый терминал (автоматизированная касса) обеспечивает расчет с клиентами супермаркета. Терминал снабжен двумя дисплеями (один для кассира, другой для покупателя), устройством чтения штрих-кода (сканером), клавиатурой для ввода данных о покупках, печатающим устройством, выдающим чеки, линией связи с сервером данных супермаркета, ящиком с деньгами.

Подходя к терминалу покупатель предъявляет покупки. Кассир начинает новый чек, при этом дисплеи высвечивают нулевую сумму. Каждый товар снабжен штрих-кодом, который считывается сканером. На дисплее высвечивается наименование и цена товара. Если товар штучный, кассир вводит его количество, стоимость высвечивается и добавляется к сумме чека. Если товар весовой, кассир вводит вес, стоимость высвечивается и добавляется. Покупатель может предъявить дисконтную карту, снабженную штрих-кодом, дающую право на скидки либо по отдельным позициям чека, либо на общую сумму чека. При ошибке кассира или по требованию покупателя последняя (текущая) позиция чека может быть удалена, при этом текущей становится предыдущая позиция. Когда все сведения о покупках введены верно, кассир получает деньги от покупателя, вводит полученную сумму и закрывает чек. При этом терминал рассчитывает сдачу и высвечивает ее на дисплеи, а также печатает чек с указанием всех позиций (для каждой печатается код товара, наименование товара, количество, цена и стоимость, скидка если есть), общего размера скидки, итоговой суммы к оплате, сведений о кассире, полученной от покупателя суммы и сдачи. Кассир выдает чек и сдачу, кладет деньги в ящик и готовится обслужить следующего клиента.

Терминал следит за общей выручкой, и если она достигает определенного значения (200 000), перед обслуживанием очередного клиента он информирует кассира о необходимости сдать выручку. В таком случае кассир должен приостановить обслуживание клиентов, сдать деньги менеджеру, оставив минимальную сумму для расчетов с последующими клиентами (5 000).

В начале работы кассир авторизуется при помощи рабочей карты (со штрих-кодом) и пароля. Только после успешной авторизации он может работать с терминалом и получить доступ к ящику с деньгами. В него он помещает, полученные от менеджера 5 000 для выдачи сдачи покупателям. Эта сумма вводится в терминал как начальная сумма выручки. В конце рабочего дня кассир закрывает день, сдает менеджеру всю выручку, инициализирует передачу сведений о покупках за день на сервер супермаркета и выключает терминал по окончании передачи, заново авторизуясь при помощи пароля.

Данные о товарах, скидках, дисконтных картах покупателей и рабочих картах кассиров и паролях торговый терминал получает с сервера данных супермаркета. В ходе выполнения этого варианта задания должна быть разработана схема базы данных сервера супермаркета. Проектировать ввод и обновление этих данных не следует.

В связи с изменением в российском законодательстве торговый терминал должен особым образом обрабатывать покупку спиртных напитков. Если в покупке было спиртное, при закрытии чека терминал информирует кассира о необходимости проверить возраст покупателя по документам. После 22-00 закрытие чеков с крепкими спиртными напитками не производится.

## Вариант 9

Домофон регулирует доступ в подъезд многоквартирного дома. В подъезде имеется дверь с замком. С наружной стороны двери установлена панель с кнопками на каждую квартиру, микрофон и динамик. В каждой квартире имеется кнопка «СВЯЗЬ», «БЛОКИРОВКА» и «ОТКРЫТЬ». Кроме того, в квартире имеется микрофон и динамик.

Жильцы могут открывать дверь ключом. Посетитель может нажать кнопку квартиры. В квартире раздается звонок. При нажатии кнопки «СВЯЗЬ» внутри квартиры устанавливается звуковое сообщение между квартирой и посетителем. Звуки, произносимые посетителем в микрофон, установленный в подъезде, должны быть слышны в динамике, установленном в квартире. При нажатии на кнопку «ОТКРЫТЬ» дверь в подъезд открывается.

Нажатие на кнопку «БЛОКИРОВКА» отключает подачу звонка в квартиру при нажатии на кнопку квартиры в подъезде. Повторное нажатие на кнопку «БЛОКИРОВКА» вновь включает подачу звонка.

Сведения о посещении дома (когда и из какой квартиры открывали замок домофона) сохраняются в журнале домофона. По прошествии суток журнал домофона пересылается на пульт начальника охраны. После пересылки данных журнал обнуляется.

## Вариант 10

В банке используется информационная система, предназначенная для использования исключительно сотрудниками банка. Система хранит список клиентов и их счетов, открытых в банке. Клиентами банка являются физические лица и организации. Информация о клиенте включает ФИО в случае физического лица/наименование в случае юридического лица, адрес регистрации/юридический адрес, адрес фактического проживания/физический адрес, телефоны, адреса электронной почты, список счетов клиента.

Счет клиента идентифицируется его уникальным номером. Информация о счете включает дату открытия счета; текущий баланс; все начисления и списания (дата и сумма); вид счета; ФИО/наименование клиента, которому принадлежит счет, и отделение, в котором открыт счет.

Вид счета идентифицируется номером и имеет наименование. Информация о виде счета включает максимально доступный кредит при списании и срок, в течение которого кредит должен быть погашен; процент, начисляемый от положительного баланса счета, а также интервал, по истечении которого баланс должен быть проиндексирован на указанный процент.

За работу с данными клиентов банка отвечает менеджер. В его обязанности входит добавление нового клиента и заведение счета для клиента. Клиент может иметь несколько счетов, поэтому менеджер имеет возможность читать, добавлять, удалять и редактировать данные клиента; заводить новый счет, закрывать существующий (со снятием всех денег или переводом их на другой счет клиента в этом же банке), а также читать информацию о счете.

Оформление списаний и начислений осуществляются кассиром по обращению клиента. Кассир может работать с любыми счетами клиентов банка. По каждой проведенной кассиром операции печатается справка, выдаваемая клиенту банка. Учет начислений по процентам система осуществляет автоматически по истечении интервала, определяемого видом счета. По требованию клиента кассир может распечатать справку о текущем состоянии счета этого клиента.