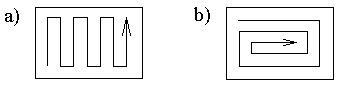
**Задача 1**

**Выполнить следующие задания, используя процедуры и функции.**

1. Будем называть соседями элемента с индексами *i*, *j* некоторой матрицы такие элементы, индексы которых отличатся от *i*, *j* не более чем на единицу. Для данной целочисленной матрицы *А* (*aij*) размерности *m*×*m* найти матрицу *В*, состоящую из нулей и единиц, элемент которой *bij* равен единице, когда среди соседей *aij* есть не менее двух совпадающих с *aij*.
2. Элемент матрицы назовем седловой точкой, если он является наименьшим в своей строке и одновременно наибольшим в своем столбце или, наоборот, является наибольшим в своей строке и наименьшим в своем столбце. Для заданной целой матрицы размером *N* × *M* напечатать индексы всех ее седловых точек.
3. Для заданной вещественной матрицы определить, образуют ли ее элементы упорядоченную последовательность следующего вида:



1. В квадратной матрице определить количество строк

a) упорядоченных по возрастанию;

b) упорядоченных по убыванию;

c) состоящих из равных элементов;

d) неупорядоченных.

Использовать функцию, определяющую свойство каждой строки.

1. Задано конечное множество имен жителей некоторого города, причем для каждого из жителей перечислены имена его детей. Жители *А* и *Б* называются родственниками, если:

либо *А* — ребенок *Б*

либо *Б* — ребенок *А*

либо существует некий *В* такой, что *А* является родственником *В*, а *В* является родственником *Б*.

Перечислить все пары жителей города, которые являются родственниками.

1. В матрице *А*(*n*×*n*) определить количество строк, элементы которой образуют арифметическую прогрессию. Использовать подпрограмму проверки строки.
2. В заданной матрице *А*(*n*×*n*) найти максимум из всех минимальных элементов матрицы по столбцам.
3. В заданной матрице *А*(*n*×*n*) найти минимум всех сумм абсолютных величин элементов матрицы по столбцам. Для нахождения суммы абсолютных величин столбца использовать функцию.
4. Подсчитать количество строк матрицы *А*(*n*×*n*), элементы которых образуют монотонную последовательность. Для определения монотонности использовать процедуру.
5. Уплотнить матрицу *А*(*n*×*n*) влево и вверх. Для выявления нулевых строк и столбцов и столбцов использовать подпрограмму.
6. Матрица *А*(*n*×*n*) просматривается сверху вниз про строкам. Найти скалярное произведение строки и столбца, соответствующих строке с первым найденным отрицательным элементом и последним нулевым.
7. Упорядочить строки матрицы *A*(*m*, *n*) по неубыванию суммы элементов строк.
8. Упорядочить строки матрицы *A*(*m*, *n*) по неубыванию их евклидовых норм.
9. Упорядочить строки матрицы *A*(*m*, *n*) по неубыванию количества их нечетных элементов.
10. Упорядочить строки матрицы *A*(*m*, *n*) по неубыванию сумм цифр элементов строк.
11. Проверить, есть ли в матрице *А*(*n*×*n*) строки, не содержащие более двух отрицательных элементов.
12. Дана матрица *А*(*n*×*n*). Построить вектор, каждый элемент которого содержит наименьший по абсолютной величине элемент строки.
13. Составить программу поиска минимального элемента, расположенного под главной диагональю, и максимального элемента, расположенного над главной диагональю заданной вещественной матрицы *А*(*n*×*n*).
14. Заполнить квадратную матрицу *N*×*N* следующим образом: элементы, расположенные на главной диагонали, принять равными 1; выше главной диагонали — сумме индексов; ниже — их разности.
15. Заполнить квадратную матрицу *N*×*N* единицами и нулями в шахматном порядке, начиная с верхнего левого угла.
16. Дана матрица *N*×*M*. Определить сумму элементов, кратных 3, и количество отрицательных элементов.
17. Задана квадратная матрица *N*×*N*. Определить, где больше четных элементов: выше или ниже главной диагонали.
18. Дана квадратная целочисленная матрица *N*×*N*. Найти суммы элементов строк, имеющих четные элементы на главной диагонали.
19. Даны две квадратных матрицы *A*(*N*×*N*) и *B*(*M*×*M*). Определить сумму элементов, расположенных на главных диагоналях.
20. Дана матрица *N*×*M*. Определить четные элементы, имеющие нечетную сумму индексов. Показать индексы этих элементов.
21. Задана квадратная матрица *N*×*N*. Найти суммы элементов тех строк, у которых элементы, расположенные на главной диагонали, равны нулю.
22. Дана действительная квадратная матрица *N*×*N*. Найти максимальный элемент на главной диагонали и вывести строку, в которой он находится.
23. Задана квадратная матрица *N*×*N*. Найти номер столбца *К* и строки *L* с максимальным произведением. Сформировать вектор *В* (2*n*), нечетные элементы которого равны сумме, а четные — разности элементов *К*-го столбца и *L*-й строки матрицы.
24. В матрице *A*(*N*×*M*) найти строку с максимальной суммой элементов и строку с минимальной суммой элементов. Далее сформировать вектор *B*(*2*×*M*), элементы которого чередовались бы с элементами максимальной и минимальной строк.
25. Дана действительная квадратная матрица *N*×*N*. Требуется переставить строки матрицы по возрастанию первых элементов строк.
26. Дана действительная квадратная матрица *N*×*N*. Требуется преобразовать матрицу: поэлементно вычесть последнюю строку из всех строк, кроме последней.
27. Задана прямоугольная матрица *A*(*N*×*M*). Найти *k* — номер строки с максимальной суммой элементов. Далее сформировать матрицу *B*(*N*×*M*), каждый элемент строки которой равнялся бы элементу соответствующей строки матрицы *А*, деленному на соответствующий элемент *k*-й строки.
28. Дана квадратная матрица *A*(*N*×*N*). За один просмотр элементов матрицы *A*(*N*×*N*) сформировать вектор *С*(*N*), каждый *j*-й элемент которого равен произведению элементов *j*-го столбца исходной матрицы, и вектор *D*(*N*), каждый *j*-й элемент которого равен сумме соответствующей строки матрицы *А*.
29. Задана квадратная матрица *A*(*N*×*N*). Найти *k* — номер столбца с максимальной суммой элементов и номер строки *l* c минимальной суммой элементов, а также элемент с минимальным значением в матрице *А*. Далее сформировать вектор *Р*(*N*), каждый элемент которого равен разности соответствующих элементов *k*-столбца и *l‑*строки, деленной на минимальный элемент матрицы *А*.
30. Задана квадратная матрица *A*(5×*5*). Сформировать вектор *x*(5), каждый элемент которого равен наибольшему значению элементов соответствующей строки исходной матрицы. Вычислить *x*1*x*5+ *x*2*x*4+ *x*3*x*3+ *x*4*x*2+ *x*5*x*1.
31. В данной целочисленной квадратной матрице *A*(*N*×*N*) указать индексы всех элементов, имеющих наибольшее значение.
32. Даны две прямоугольные матрицы *A*(*N*×*M*) и *B*(*N*×*M*). Найти матрицу *C*(*N*×*M*), элементы которой равны сумме соответствующих элементов матриц *А* и *В*, после чего произвести транспонирование полученной матрицы *С*.
33. Дана квадратная матрица *A*(*N*×*N*). За один просмотр найти строку с минимальной суммой элементов и строку с максимальной суммой элементов и образовать произведение этих строк.

**Задача 2**

1. Дано целое число *N* и текстовый файл. Создать новый файл, содержащий все слова длины *N* из исходного файла (знаки препинания, расположенные в начале и в конце слов, не учитывать). Если исходный файл не содержит слов длины *N*, оставить результирующий файл пустым.
2. Дан символ *C* (прописная русская буква) и текстовый файл. Создать строковый файл, содержащий все слова из исходного файла, начинающиеся1|оканчивающиеся2 этой буквой (как прописной, так и строчной). Знаки препинания, расположенные в начале и в конце слов, не учитывать. Если исходный файл не содержит подходящих слов, оставить результирующий файл пустым.
3. Дано число *N* и текстовый файл. Удалить из файла абзац с номером *N* (абзацы отделяются друг от друга одной или несколькими пустыми строками и нумеруются от 1). Пустые строки, предшествующие и следующие за удаляемым абзацем, не удалять. Если абзац с данным номером отсутствует, то оставить файл без изменений.
4. Дано число *N* и текстовый файл. Удалить из файла абзац с номером *N* (абзацы выделяются с помощью красной строки (5 пробелов) и нумеруются от 1). Пустые строки между абзацами не учитывать и не удалять. Если абзац с данным номером отсутствует, то оставить файл без изменений.
5. Дан текстовый файл, каждая строка которого изображает целое число, дополненное слева и справа несколькими пробелами. Вывести сумму этих чисел и их количество.
6. Дан текстовый файл, каждая строка которого изображает целое или вещественное число, дополненное слева и справа несколькими пробелами (вещественные числа имеют ненулевую дробную часть). Вывести сумму целых1|вещественных2 чисел и их количество.
7. Дан текстовый файл, каждая строка которого содержит изображения нескольких вещественных чисел, разделенных пробелами. Создать файл вещественных чисел, содержащий эти числа в том же порядке.
8. Даны два текстовых файла с именами *Name1* и *Name2*. Добавить в начало1|конец2 каждой строки файла *Name1* соответствующую строку файла *Name2*. Если файл *Name2* короче файла *Name1*, то оставшиеся строки файла *Name1* не изменять.
9. Дан текстовый файл *NameT* и файл целых чисел *NameN*. Добавить в начало1|конец2 каждой строки файла *NameT* изображение соответствующего числа из файла *NameN*. Если файл *NameN* короче файла *NameT*, то оставшиеся строки файла *NameT* не изменять.
10. Дан текстовый файл с именем *NameT*. В каждой его строке первые 60 позиций отводятся под текст, а оставшаяся часть — под вещественное число. Создать два файла: строковый файл с именем *NameS*, содержащий текстовую часть исходного файла, и файл вещественных чисел с именем *NameR*, содержащий числа из исходного файла.
11. Даны два файла целых чисел одного размера с именами *Name1* и *Name2*. Создать текстовый файл с именем *NameT*, содержащий изображения этих чисел, расположенные в два столбца шириной по 30 символов: первый содержит числа из файла *Name1*, второй — из файла *Name2*. В начале и конце каждой строки текстового файла ввести разделитель " | " (код 124). Числа выравниваются по левому1|правому2 краю столбца.
12. Даны вещественные числа *A*, *B* и целое число *N*. Создать текстовый файл, содержащий таблицу значений функции *f*(*x*) = [*sin*(*x*)]1|[*cos*(*x*)]2|[*exp*(*x*)]3 на промежутке [*A*, *B*] с шагом   
    (*B*–*A*)/*N*. Таблица состоит из двух столбцов: с аргументами *x* (10 позиций, из них 3 под дробную часть) и со значениями *f*(*x*) (15 позиций, из них 8 под дробную часть). Столбцы выравниваются по правому краю и разделяются 10 пробелами.
13. Дан текстовый файл с именем *NameT*, содержащий таблицу из трех столбцов вещественных чисел. Ширина столбцов таблицы и способ их выравнивания являются произвольными. Специальных символов-разделителей таблица не содержит. Создать файлы вещественных чисел с именами *Name1*, *Name2* и *Name3*, каждый из которых содержит числа из соответствующего столбца таблицы.
14. Дан текстовый файл, представляющий собой таблицу, состоящую из трех столбцов с целыми числами. В начале и в конце каждой строки таблицы, а также между ее столбцами располагается символ-разделитель. Ширина столбцов таблицы и способ их выравнивания являются произвольными. Создать файл целых чисел, содержащий сумму чисел из каждой строки исходной таблицы.
15. Дан текстовый файл. Создать новый файл, содержащий все символы, встретившиеся в тексте, включая пробел и знаки препинания (без повторений). Символы располагать в порядке [возрастания их кодов]1|[убывания их кодов]2|[их первого появления в тексте]3.
16. Дан текстовый файл с именем *NameT*. Подсчитать число повторений в нем строчных русских букв ("*а*"—"*я*") и создать строковый файл с именем *NameS*, элементы которого имеют вид: «<буква>—<число повторений данной буквы>». Буквы, отсутствующие в тексте, в файл не включать. Строки упорядочить по [возрастанию кодов букв]1|[убыванию числа повторений букв, а при равном числе повторений — по возрастанию кодов букв]2.
17. Дано целое число *N* и текстовый файл с именем *Name1*, содержащий один абзац текста, выровненный по левому краю. Отформатировать текст так, чтобы его ширина не превосходила *N* позиций, и выровнять текст по левому1|правому2 краю. Пробелы в конце строк удалить. Сохранить отформатированный текст в новом текстовом файле с именем *Name2*.
18. Дано целое число *N* и текстовый файл *Name1*, содержащий текст, выровненный по левому краю. Абзацы текста отделяются друг от друга одной пустой строкой. Отформатировать текст так, чтобы его ширина не превосходила *N* позиций, и выровнять текст по левому1|правому2 краю, сохранив деление на абзацы. Пробелы в конце строк удалить. Сохранить отформатированный текст в новом текстовом файле *Name2*.
19. Дана строка *K*, состоящая из 10 цифр, и файл с русским текстом. Зашифровать файл, выполнив циклическую замену каждой русской буквы, стоящей на *i*-й позиции строки, на букву того же регистра, расположенную в алфавите на *K*[*i*]-м месте после шифруемой буквы (символы строки *K* также перебираются циклически: для *i* = 11 снова используется смещение *K*[*1*] и т.д.). Букву ‘ё’ в алфавите не учитывать, знаки препинания и пробелы не изменять.
20. Дана строка  *S1* и файл с русским текстом, зашифрованным по правилу, описанному в задании 39. Строка *S1* представляет собой первую расшифрованную строку текста. Расшифровать остальные строки и заменить в файле зашифрованный текст на расшифрованный. Если информации для расшифровки недостаточно, то исходный файл не изменять.
21. Прочитать текстовый файл. На отдельной форме вывести информацию о количестве слов, состоящих из одного символа, двух символов и т.д.
22. Прочитать текстовый файл. На отдельной форме вывести словарь, используемых слов с указанием частоты их использования.
23. Прочитать текстовый файл. На отдельной форме вывести текст, из которого удалены незначащие пробелы.
24. Прочитать текстовый файл. На отдельной форме вывести этот же текст, но в котором каждое новое предложение написано с новой строки (предложения заканчиваются точкой).
25. Прочитать текстовый файл. Реализовать функцию замены одного слова на другое.
26. Написать программу для кодирования и декодирования текстового файла с помощью слова-пароля.
27. Прочитать текстовый файл. На отдельной форме вывести текст, в котором все слова из латинских символов и цифр выделены другим цветом.
28. Прочитать текстовый файл. На отдельной форме вывести тот же текст, но в котором все числа записаны словами.
29. В текстовом файле записан английский /немецкий текст. Необходимо выделять сплошной фрагмент текста и определять, все ли буквы латинского алфавита в нём задействованы, отметить незадействованные.
30. Прочитать текстовый файл, в котором хранятся статьи уголовного кодекса. Разбить весь текст на отдельные статьи, если известно, что признаком начала статьи является предложение, состоящее из слова <Статья>, за которым следует номер статьи с точкой, написанное в начале строки.
31. Прочитать текстовый файл, в котором хранятся одна статья уголовного кодекса. Разбить эту статью на пункты, если известно, что признаком начала пункта является предложение, состоящее из номера пункта с точкой, написанное в начале строки.
32. Прочитать текстовый файл, в котором хранятся одна статья уголовного кодекса, состоящая из одного единственного пункта. Отделить от этой статьи санкцию, если известно, что она начинается со слов <наказывается> или <наказываются>.
33. Описать функцию *getInt*(*Name*, *k*) целого типа, возвращающую *k*-й элемент файла целых чисел с именем *Name* (элементы нумеруются от 0). Если файл не существует или не содержит *k*-го элемента, то функция возвращает 0. С помощью этой функции вывести пять элементов данного файла с указанными номерами.
34. Описать функцию *getLine*(*Name*, *k*) строкового типа, возвращающую *k*-ю строку текстового файла с именем *Name* (строки нумеруются от 0). Если файл не существует или не содержит *k*-й строки, то функция возвращает пустую строку. С помощью этой функции вывести пять строк данного файла с указанными номерами.
35. Описать функцию *IntFileSize*(*Name*) целого типа, возвращающую размер файла целых чисел с именем *Name*. Если файл не существует, то функция возвращает –1. С помощью этой функции определить размер трех файлов с данными именами.
36. Описать функцию *TextSize*(*Name*) целого типа, возвращающую число строк в текстовом файле с именем *Name*. Если файл не существует, то функция возвращает –1. С помощью этой функции определить размер трех файлов с данными именами.
37. Описать процедуру *InvertIntFile*(*Name*), меняющую порядок следования элементов файла целого типа с именем *Name* на противоположный. Если файл не существует или содержит менее двух элементов, то процедура не выполняет никаких действий. Обработать с помощью этой процедуры три файла с данными именами.
38. Описать процедуру *SplitIntFile*(*Name0*, *k*, *Name1*, *Name2*), копирующую первые *k* (>= 0) элементов существующего файла целых чисел с именем *Name0* в файл *Name1*, а остальные элементы — в файл *Name2* (прежнее содержимое результирующих файлов стирается). Один из результирующих файлов может оказаться пустым. Применить эту процедуру к файлу *Name0*, используя указанные значения *Name1*, *Name2* и *k*.
39. Описать процедуру *SplitText*(*Name0*, *k*, *Name1*, *Name2*), копирующую первые *k* (>= 0) строк существующего текстового файла с именем *Name0* в файл *Name1*, а остальные элементы — в файл *Name2* (прежнее содержимое результирующих файлов стирается). Один из результирующих файлов может оказаться пустым. Применить эту процедуру к файлу *Name0*, используя указанные значения *Name1*, *Name2* и *k*.
40. Описать процедуру *ConcatFile*(*NameA*, *NameB*, *NameAB*), позволяющую объединить содержимое двух двоичных файлов *NameA* и *NameB* одного и того же типа в новом файле *NameAB*. Использовать процедуры *BlockRead* и *BlockWrite*. Применить эту процедуру к парам исходных файлов *Name1*—*Name2*, *Name1*—*Name3* и *Name2*—*Name3*, создав файлы с именами *Name12*, *Name13*, *Name23*.

**Задача 3**

Создать приложение, удовлетворяющее требованиям, приведенным в задании. Наследование применять только в тех заданиях, в которых это логически обосновано. Аргументировать принадлежность классу каждого создаваемого метода. Для каждого класса должно быть не менее 2 свойств.

1.

Создать объект класса Текст, используя классы Предложение, Слово.

Методы: дополнить текст, вывести текст, заголовок текста.

2.

Создать объект класса Автомобиль, используя классы Колесо, Двигатель.

Методы: ехать, заправляться, менять колесо, вывести марку автомобиля.

3.

Создать объект класса Самолет, используя классы Крыло, Шасси, Двигатель. Методы: летать, задавать маршрут, вывести маршрут.

4.

Создать объект класса Государство, используя классы Область, Район, Город. Методы: вывести столицу, количество областей, площадь, областные центры.

5.

Создать объект класса Планета, используя классы Материк, Океан, Остров. Методы: вывести название материка, планеты, количество материков.

6.

Создать объект класса Звездная система, используя классы Планета, Звезда, Луна.

Методы: вывести количество планет в звездной системе, название звезды, добавление планеты в систему.

7.

Создать объект класса Компьютер, используя классы Винчестер, Оперативная память, Процессор, Видеокарта. Методы: включить, выключить, проверить на вирусы, вывести размер винчестера.

8.

Создать объект класса Квадрат, используя классы Точка, Отрезок.

Методы: задание размеров, растяжение, сжатие, поворот, изменение цвета.

9.

Создать объект класса Круг, используя классы Точка, Окружность.

Методы: задание размеров, изменение радиуса, определение принадлежности точки данному кругу.

10.

Создать объект класса Текстовый файл, используя классы Файл, Директория.

Методы: создать, переименовать, вывести содержимое, дополнить, удалить.

11.

Создать объект класса Одномерный массив, используя классы Массив, Элемент. Методы: создать, вывести, выполнить операции (сложить, вычесть, перемножить).

12.

Создать объект класса Простая дробь, используя класс Число.

Методы: вывод, сложение, вычитание, умножение, деление.

13.

Создать объект класса Дом, используя классы Окно, Дверь.

Методы: закрыть на ключ, вывести на консоль количество окон, дверей.

14.

Создать объект класса Фотоальбом, используя классы Фотография, Страница.

Методы: задать название фотографии, дополнить фотоальбом фотографией, вывести количество фотографий.

15.

Создать объект класса Год, используя классы Месяц, День.

Методы: задать дату, вывести день недели по заданной дате, рассчитать количество дней, месяцев в заданном временном промежутке.

16.

Создать объект класса Сутки, используя классы Час, Минута.

Методы: вывести текущее время, рассчитать время суток (утро, день, вечер, ночь).

17.

Новогодний подарок.

Определить иерархию конфет и прочих сладостей. Создать несколько объектов-конфет. Собрать детский подарок с определением его веса. Провести сортировку конфет в подарке на основе одного из параметров. Найти конфету в подарке, соответствующую заданному диапазону содержания сахара.

18.

Домашние электроприборы.

Определить иерархию электроприборов. Включить некоторые в розетку. Подсчитать потребляемую мощность. Провести сортировку приборов в квартире на основе мощности. Найти прибор в квартире, соответствующий заданному диапазону параметров.

19.

Звукозапись.

Определить иерархию музыкальных композиций. Записать на диск сборку. Подсчитать продолжительность. Провести перестановку композиций диска на основе принадлежности к стилю. Найти композицию, соответствующую заданному диапазону длины треков.

20.

Камни.

Определить иерархию драгоценных и полудрагоценных камней. Отобрать камни для ожерелья. Подсчитать общий вес (в каратах) и стоимость. Провести сортировку камней ожерелья на основе ценности. Найти камни в

ожерелье, соответствующие заданному диапазону параметров прозрачности.

21.

Транспорт.

Определить иерархию подвижного состава железнодорожного транспорта. Создать пассажирский поезд. Подсчитать общую численность пассажиров и багажа. Провести сортировку вагонов поезда на основе уровня комфортности. Найти в поезде вагоны, соответствующие заданному диапазону параметров числа пассажиров.

22.

Авиакомпания.

Определить иерархию самолетов. Создать авиакомпанию. Посчитать общую вместимость и грузоподъемность. Провести сортировку самолетов компании по дальности полета. Найти самолет в компании, соответствующий заданному диапазону параметров потребления горючего.

23.

Таксопарк.

Определить иерархию легковых автомобилей. Создать таксопарк. Подсчитать стоимость автопарка. Провести сортировку автомобилей парка по расходу топлива. Найти автомобиль в компании, соответствующий заданному диапазону параметров скорости.

24.

Страхование.

Определить иерархию страховых обязательств. Собрать из обязательств дериватив. Подсчитать стоимость. Провести сортировку обязательств в

деривативе на основе уменьшения степени риска. Найти обязательство в

деривативе, соответствующее заданному диапазону параметров.

25.

Мобильная связь.

Определить иерархию тарифов мобильной компании. Создать список тарифов компании. Подсчитать общую численность клиентов. Провести сортировку тарифов на основе размера абонентской платы. Найти тариф в компании, соответствующий заданному диапазону параметров.

26.

Налоги.

Определить множество и сумму налоговых выплат физического лица за год с учетом доходов с основного и дополнительного мест работы, авторских вознаграждений, продажи имущества, получения в подарок денежных сумм и имущества, переводов из-за границы, льгот на детей и материальной помощи. Провести сортировку налогов по сумме.

27.

Счета.

Клиент может иметь несколько счетов в банке. Учитывать возможность блокировки/разблокировки счета. Реализовать поиск и сортировку счетов. Вычисление общей суммы по счетам. Вычисление суммы по всем счетам, имеющим положительный и отрицательный балансы отдельно.

28.

Туристические путевки.

Сформировать набор предложений клиенту по выбору туристической путевки различного типа (отдых, экскурсии, лечение, шопинг, круиз и т.д.) для оптимального выбора. Учитывать возможность выбора транспорта, питания и числа дней. Реализовать выбор и сортировку путевок.

29.

Кредиты.

Сформировать набор предложений клиенту по целевым кредитам различных банков для оптимального выбора. Учитывать возможность досрочного погашения кредита и/или увеличения кредитной линии. Реализовать выбор и поиск кредита

30.

Фургон кофе.

Загрузить фургон определенного объема грузом на определенную сумму из различных сортов кофе, находящихся, к тому же, в разных физических состояниях (зерно, молотый, растворимый в банках и пакетиках). Учитывать объем кофе вместе с упаковкой. Провести сортировку товаров на основе соотношения цены и веса. Найти в фургоне товар, соответствующий заданному диапазону параметров качества.

**Задача 4**

Полиморфизм

***Задание:*** построить класс 1-го уровня с указанными в индивидуальном задании полями и методами:

* конструктор;
* функция, которая определяет «качество» объекта – *Q* по заданной формуле (столбец 2);
* вывод информации об объекте.

Построить класс 2-го уровня(класс-потомок), который содержит:

* дополнительное поле *P*;
* функция, которая определяет «качество» объекта класса 2-го уровня **–** Qp, которая перекрывает функцию качества класса 1-го уровня (*Q* ), выполняя вычисление по новой формуле (столбец 3).

Создать проект для демонстрации работы: ввод и вывод информации об объектах классов 1-го и 2-го уровней.

|  |  |  |
| --- | --- | --- |
| №  вар | Поля и функция «качества» (*Q* ) класса 1-го уровня | Поле и функция «качества» *Qp* класса 2-го уровня |
| 1 | Компьютер:  - наименование процессора;  -тактовая частота процессора (МГц); - объем оперативной памяти (Мб). *Q* = (0,1·частота) + память | *P*: объем винчестера (Гб)  *Qp*=*Q*+0,5·*Р* |
| 2 | Оператор мобильной связи:   * название оператора; * стоимость 1 минуты разговора; - площадь покрытия.   *Q* = 100 · площадь покрытия / стоимость 1 минуты разговора | *P*: наличие платы за каждое соединение  *Qp* = 0,7·Q, если *Р* - истина, иначе *Qp* = 1,5·*Q* |
| 3 | Товар на складе:   * наименование; * цена; * количество.   *Q* = цена / количество | *P*: год выпуска товара    *Qp* = *Q* + 0,5· (*Т–Р*), где *Т* - текущий год |
| 4 | Кабель:   * тип; * количество жил кабеля; - диаметр.   *Q* = диаметр / количество жил | *P*: наличие оплетки    *Qp*: если *Р* - истина, то *Qp* = 2·*Q;* иначе *Qp* = 0,7·*Q* |
| 5 | Учебник по программированию:   * название; * количество страниц; - цена.   *Q* = цена / количество страниц | *P:* год издания    *Qp* = *Q* – 0,2· (*Т–Р*),  где *Т* - текущий год |
| 6 | Мобильный телефон:   * марка; * цена; * объем памяти.   *Q* = объем памяти / цена | *P*: количество SIM карт    *Qp* = *Q·Р* |
| 7 | Спутниковая антенна (тарелка):  - диаметр; - материал; - цена.    *Q* = диаметр / цена | *P*: тип подвески (азимутальная, полярная, тороидальная)    *Qp*= *Q*, если тип подвески азимутальный,  *Qp* = 2·*Q*, если тип подвески полярный,  *Qp* = 2,5·*Q*, если тип подвески тороидальный |
| 8 | Экзамен:   * дисциплина; * число студентов на экзамене; - продолжительность экзамена (ч).   *Q* = число студентов / продолжительность | *P*: процент двоек      *Qp* = *Q*· ( 100-*Р*)/100 |
| 9 | Спортсмен:   * фамилия; * число соревнований; * сумма мест, занятых спортсменом в соревнованиях.   *Q* = (число соревнований)/ (сумма мест) | P: занимал ли хотя бы раз первое место    *Qp* = 1,5·*Q*, если *Р* - истина, иначе – *Qp*= *Q*, |
| 10 | Программист:   * фамилия; * число программ, написанных программистом; * число языков программирования, которыми он пишет программы.   Q = (число программ)\*(число языков) | *P*: число программ, которые работают правильно      *Qp* = *Q* · *Р* / (число всех программ) |
| 11 | Спектакль: - название; *n*1 – число зрителей в начале; *n*2 – число зрителей в конце.  *Q* = (*n*2 – *n*1)/ *n*1 | *P*: год написания пьесы    *Qp* = *Q* · (*Т-Р*+1), где Т - текущий год |
| 12 | Алмаз:   * название; * вес (в каратах); * качество огранки в баллах (число).   *Q* = 0,4·вес + 0,6·качество огранки | *P*: цвет (белый, голубой, желтый, и тп)  *Qp*: если цвет голубой, то *Qp*=*Q+*1;  а если желтый, то *Q*p = *Q* **–** 0,5  иначе *Qp*=*Q* |
| 13 | Компьютерная сеть:   * название организации; * число рабочих станций; * среднее расстояние между станциями   (м).  *Q* = число станций · среднее расстояние | *P*: средняя скорость передачи данных в сети (Мб/с)    *Qp* = *Q·Р* |
| 14 | Армия:   * вид войск; * численность (тыс человек): * вооруженность (баллы - число).   *Q*=0,3·численность+0,7·вооруженность | *P*: опыт (число месяцев, на протяжении которых армия вела боевые действия)  *Qp* = *Q* · (*Р*+1) |
| 15 | Автомобиль:   * марка автомобиля; * мощность двигателя (кВт); - число мест.     *Q* = 0,1·мощность·число мест | *P*: год изготовления    *Qp*=*Q*-1,5· (*Т*-*Р*) , где *Т* - текущий год |
| 16 | Партия:   * название; * численность (тыс. членов); * процент голосов на последних выборах.     *Q* = 0,3·численность+0,7· процент гол | *P*: численность партии в прошлом году    *Qp*: если численность в текущем году увеличилась, то *Qp*=1,2·*Q*;  а если сократилась, то  *Qp*=0,8·*Q* |
| 17 | Высшее учебное заведение:   * название заведения; * количество студентов, зачисленных на   1-й курс;   * количество выпускников.   *Q* = количество выпускников / количество зачисленных | *P*: процент выпускников, которые работают по специальности    *Qp* = *Р\*Q* |
| 18 | Солдат:  - фамилия; - рост (м); - вес (кг).  *Q* = рост·вес | *P*: образование (начальное, среднее, высшее)  *Qp*: если образование высшее, то *Qp*=2·*Q*; а если начальное, то *Qp*=0,5·*Q*; иначе *Qp*=*Q* |
| 19 | Телевизор:   * фирма; * диагональ экрана (дюйм); - звуковая мощность (дб).   *Q* = диагональ+(0,05·мощность) | *P*: страна-производитель *Qp:* если страна - Япония, то *Qp*=2·*Q*; а если Сингапур или Корея, то *Qp*=1,5·*Q*; иначе *Qp*=*Q* |
| 20 | Митинг:  - название события; *n*1 – число ораторов; *n*2 – число участников. *Q* = *n*1/*n*2 | *P*: число групп ораторов, которые высказывали одинаковые мысли    *Qp* = *Q* + *Р*/*n*2 |
| 21 | Дом:  - номер дома; - число квартир; - год сооружения.  *Q*=(число квартир)+2· (*Т* –  – год сооружения),  где *Т* - текущий год | P: район (центр, окраина, и тп)    *Qp*: если район - центр, то *Qp*=2·*Q*;  иначе *Qp*=0,5·*Q* |
| 22 | Руководитель:   * фамилия; * самооценка (в баллах - целое число); - оценка другими людьми (в баллах).   *Q* = (оценка другими)/самооценка | *P:* оценка потомками (в  баллах)      *Qp* = 0,3·*Q* + 0,7·*Р* |
| 23 | Студент:  - фамилия; - средний балл; - курс.    *Q* = 0,2 · средний балл · курс | *P*: изучает дисциплины на английском языке    *Qp* = 2·*Q*, если *Р* - истина, иначе *Qp* = 0,9·*Q* |
| 24 | Антенна:   * название; * мощность; - высота (м).   *Q* = мощность +0.5·высота | *P*: коэффициент излучения    *Qp* = *Q* – 0.1·*Р* |
| 25 | Самолет:   * марка; * количество двигателей; - высота полета.     Q = кол-во двигателей · высота полета  /1000 | *P*: страна-производитель    *Qp*: если страна - Россия, то  *Qp*=*Q*+1;  а если Франция, то  *Qp*=*Q*+0.5 |
| 26 | Студент:   * фамилия; * число экзаменов; - число оценок «пять».     *Q* = число оценок «пять»/ число экзаменов | *P*: число оценок «три»      *Qp* = *Q* – 0.5*Р* |
| 27 | Фирма:   * название; * доход (тыс $ ); - рейтинг (в баллах).   *Q* = доход · рейтинг | *P*: инвестиции в фирму  (тыс $ )    *Qp* = *Р*3 + *Q* |
| 28 | Военный корабль:   * название; * длина; * число пушек главного калибра. *Q* = (число пушек)/ длина | *P*: крейсерская скорость  (в морских узлах)    *Qp* = 0.25*Q* + *Р* |
| 29 | Коробка спичек:   * фирма изготовитель; * число спичек в коробке; * время горения одной спички (с).   *Q* = (число спичек) · время | *P*: средний % бракованных  спичек в коробке    *Qp* = ( 100-*Р*)*Q* / 100 |
| 30 | Полководец:  - фамилия; - число битв; - число побед.  *Q* = (число побед)2/(число битв) | *P*: число побед с меньшими, чем у противника,  силами    *Qp* = *Р*2**/**битвы +*Q* |

**Задача 5**

Для интерфейса необходимо определить 1 свойство и 2 метода.

Абстрактный класс должен содержать 3-5 свойств и 3-5 методов(включая унаследованные свойства интерфейса).

Класс должен содержать дополнительно 2 свойства и 2 метода.

В программе реализовать работу со списком объектов, который должен содержать объекты типа интерфейса.

1. interface Издание -> abstract class Книга -> class Справочник и Энциклопедия.

2. interface Абитуриент -> abstract class Студент -> class Студент-Заочник.

3. interface Сотрудник -> abstract class Инженер -> class Руководитель.

4. interface Здание -> abstract class Общественное Здание -> class Театр.

5. interface Mobile -> abstract class Siemens Mobile -> class Model.

6. interface Корабль -> abstract class Военный Корабль -> class Авианосец.

7. interface Врач -> abstract class Хирург -> class Нейрохирург.

8. interface Корабль -> abstract class Грузовой Корабль -> class Танкер.

9. interface Мебель -> abstract class Шкаф -> class Книжный Шкаф.

10. interface Фильм -> abstract class Отечественный Фильм -> class Комедия.

11. interface Ткань -> abstract class Одежда -> class Костюм.

12. interface Техника -> abstract class Плеер -> class Видеоплеер.

13. interface Транспортное Средство -> abstract class Общественный Транспорт

-> class Трамвай.

14. interface Устройство Печати -> abstract class Принтер -> class Лазерный Принтер.

15. interface Бумага -> abstract class Тетрадь -> class Тетрадь Для Рисования.

16. interface Источник Света -> abstract class Лампа -> class Настольная Лампа.

17. interface Фигура -> abstract class Геометрическая фигура -> class Ромб.

18. interface Транспортное средство -> abstract class Автомобиль -> class Грузовик.

19. interface Устройство -> abstract class Фотоаппарат -> class Цифровой фотоаппарат.

20. interface Телефон -> abstract class Мобильный телефон -> class Смартфон.

21. interface Музыкальный инструмент -> abstract class Гитара -> class Электрогитара.

22. interface Вид спорта -> abstract class Командный вид спорта -> class Футбол.

23. interface Строительная техника -> abstract class Тяжелая строительная техника -> class Экскаватор.

24. interface Участок -> abstract class Сад -> class Яблочный сад.

25. interface Хозяйство -> abstract class Животноводческая ферма -> class Молочная ферма.

26. interface Космический объект -> abstract class Звезда -> class Красный гигант.

27. interface Вычислительное устройство -> abstract class Компьютер -> class Ноутбук.

28. interface Коллектив -> abstract class Музыкальный коллектив -> class Рок-группа.

29. interface Производство -> abstract class Завод -> class Кирпичный завод.

30. interface Добывающее предприятие -> abstract class Горнодобывающее предприятие -> class Добыча руды.

**Задача 6.1**

Взять за основу задачу 5. Должно быть не менее 3 классов, которые наследуются от абстрактного класса.   
Используя рефлексию реализовать возможность создания и вызова методов данных классов на форме. Написать код, который принимает путь к библиотеке классов и ищет все классы, которые реализуют интерфейс. Далее получают всю информацию о данных классах, и возвращают список из названий классов. На форме реализовать «дроплаун» с названиями классов. При выборе класса на форму должны динамически подгружаться все методы класса с возможностью ввода параметров пользователем. При нажатии кнопки «Выполнить» должен создаваться объект и выполняться выбранный метод.

**Задача 6.2**

В зависимости от задачи необходимо смоделировать ситуацию/процесс. В каждой модели есть набор возможных ситуаций. Для некоторых событий необходимо определить вероятность возникновения данного события. Интерфейс необходимо реализовать, используя 3 и более классов.

Для решения задач необходимо использовать:

1. Делегаты/события.
2. Многопоточность
3. Где необходимо рефлексию

На форме должно быть динамическое изменение моделей – все должно двигаться. Иметь возможность добавлять несколько моделей на форму.

1. Остановка – необходимо смоделировать работу остановки транспорта. Реализовать классы – Остановка, Автобус, интерфейс – Пассажир (реализовать несколько классов). События – автобус остановился на остановке (происходит посадка интерфейса пассажир), автобус переполнен (с некоторой долей вероятности).
2. Пешеходный переход – смоделировать работу пешеходного перехода. Реализовать классы – Авто, Пешеход, интерфейс – аварийная служба. События – зеленый свет для пешехода, аварийная ситуация(с некоторой долей вероятности) – должна приезжать машина аварийной службы.
3. Квадрокоптер – смоделировать полет квадрокоптера. Реализовать классы – Квадрокоптер, Оператор, интерфейс – механик. События – оператор включает пульт управления, на квадрокоптере отключается GPS(с некоторой долей вероятности). При аварийной ситуации квадрокоптер должен сесть, механик должен починить его.
4. Конвейер с деталями – смоделировать работу конвейера производства деталей. Реализовать классы – Конвейер, Погрузчик, интерфейс – Механик. События – в конвейере закончились материалы – погрузчик загружает новую партию, конвейер сломался (с некоторой долей вероятности) – механик чинит конвейер.
5. Гонки – смоделировать гонки. Реализовать классы – Болид, Механик, интерфейс – погрузчик. События – Стерлись покрышки – заезд в боксы смена колес, Столкновение (с некоторой долей вероятности) – приезжает погрузчик.
6. Троллейбус – смоделировать работу троллейбуса. Реализовать классы – Троллейбус, Водитель, интерфейс – Аварийная служба. События - сломался(с некоторой долей вероятности) – приезжает аварийная служба, соскочили штанги(с некоторой долей вероятности) – Водитель сам ставит на место.
7. Станок по производству деталей – смоделировать работу станка. Реализовать классы – Станок, Фрезеровщик, интерфейс – погрузчик. События – работать с новой деталью – Фрезеровщик берет новую деталь и работает с ней, Деталь готова – погрузчик забирает деталь.
8. Торговая точка – смоделировать работу торговой точки. Реализовать классы – Торговая точка, Покупатель, интерфейс – служба доставки. События – покупатель покупает товар, Товар закончился.
9. Спортивные соревнования – смоделировать спортивные соревнования. Реализовать классы – Соревнования, Спортсмен, интерфейс – врач. События – спортсмен получил травму (с некоторой долей вероятности) – приходит врач и лечит, Соревнования выиграл какой-то спортсмен – вручение награды.
10. Добыча нефти – смоделировать добычу нефти. Реализовать классы – Нефтяная вышка, Механик, интерфейс – погрузчик. События - возгорание платформы(с некоторой долей вероятности), отправка нефти – приезжает погрузчик.
11. Строительство дома – смоделировать строительство дома. Реализовать классы – Дом, Каменщик, интерфейс – строительная техника. События - Закончились стройматериалы, Необходимо крыть крышу.
12. Молочная ферма – смоделировать работу молочной фермы. Реализовать классы – Механик, Склад, интерфейс – погрузчик. События – Склад заполнен – приезжает погрузчик и увозит молоко, сломалось оборудование (с некоторой долей вероятности) – механик чинит всё.
13. Доменная печь – смоделировать работу доменной печи. Реализовать классы – Доменная печь, Рабочий, интерфейс – погрузчик. События – закончился материал, перегрев(с некоторой долей вероятности).
14. Производство конфет – смоделировать производство конфет. Реализовать классы – Фабрика, погрузчик, интерфейс – Техника. События - закончился сахар, случилась авария(с некоторой долей вероятности).
15. Ракета – смоделировать запуск ракеты. Реализовать классы – Ракета, Космонавт, интерфейс – страховая компания. События – успешный запуск – космонавты выходят в открытый космос, Авария(с некоторой долей вероятности) – страховая выплачивает много денег.
16. Остановка – необходимо смоделировать работу остановки транспорта. Реализовать классы – Остановка, Автобус, интерфейс – Пассажир (реализовать несколько классов). События – автобус остановился на остановке (происходит посадка интерфейса пассажир), автобус переполнен (с некоторой долей вероятности).
17. Пешеходный переход – смоделировать работу пешеходного перехода. Реализовать классы – Авто, Пешеход, интерфейс – аварийная служба. События – зеленый свет для пешехода, аварийная ситуация(с некоторой долей вероятности) – должна приезжать машина аварийной службы.
18. Квадрокоптер – смоделировать полет квадрокоптера. Реализовать классы – Квадрокоптер, Оператор, интерфейс – механик. События – оператор включает пульт управления, на квадрокоптере отключается GPS(с некоторой долей вероятности). При аварийной ситуации квадрокоптер должен сесть, механик должен починить его.
19. Конвейер с деталями – смоделировать работу конвейера производства деталей. Реализовать классы – Конвейер, Погрузчик, интерфейс – Механик. События – в конвейере закончились материалы – погрузчик загружает новую партию, конвейер сломался (с некоторой долей вероятности) – механик чинит конвейер.
20. Гонки – смоделировать гонки. Реализовать классы – Болид, Механик, интерфейс – погрузчик. События – Стерлись покрышки – заезд в боксы смена колес, Столкновение (с некоторой долей вероятности) – приезжает погрузчик.
21. Троллейбус – смоделировать работу троллейбуса. Реализовать классы – Троллейбус, Водитель, интерфейс – Аварийная служба. События - сломался(с некоторой долей вероятности) – приезжает аварийная служба, соскочили штанги(с некоторой долей вероятности) – Водитель сам ставит на место.
22. Станок по производству деталей – смоделировать работу станка. Реализовать классы – Станок, Фрезеровщик, интерфейс – погрузчик. События – работать с новой деталью – Фрезеровщик берет новую деталь и работает с ней, Деталь готова – погрузчик забирает деталь.
23. Торговая точка – смоделировать работу торговой точки. Реализовать классы – Торговая точка, Покупатель, интерфейс – служба доставки. События – покупатель покупает товар, Товар закончился.
24. Спортивные соревнования – смоделировать спортивные соревнования. Реализовать классы – Соревнования, Спортсмен, интерфейс – врач. События – спортсмен получил травму (с некоторой долей вероятности) – приходит врач и лечит, Соревнования выиграл какой-то спортсмен – вручение награды.
25. Добыча нефти – смоделировать добычу нефти. Реализовать классы – Нефтяная вышка, Механик, интерфейс – погрузчик. События - возгорание платформы(с некоторой долей вероятности), отправка нефти – приезжает погрузчик.
26. Строительство дома – смоделировать строительство дома. Реализовать классы – Дом, Каменщик, интерфейс – строительная техника. События - Закончились стройматериалы, Необходимо крыть крышу.
27. Молочная ферма – смоделировать работу молочной фермы. Реализовать классы – Механик, Склад, интерфейс – погрузчик. События – Склад заполнен – приезжает погрузчик и увозит молоко, сломалось оборудование (с некоторой долей вероятности) – механик чинит всё.
28. Доменная печь – смоделировать работу доменной печи. Реализовать классы – Доменная печь, Рабочий, интерфейс – погрузчик. События – закончился материал, перегрев(с некоторой долей вероятности).
29. Производство конфет – смоделировать производство конфет. Реализовать классы – Фабрика, погрузчик, интерфейс – Техника. События - закончился сахар, случилась авария(с некоторой долей вероятности).
30. Ракета – смоделировать запуск ракеты. Реализовать классы – Ракета, Космонавт, интерфейс – страховая компания. События – успешный запуск – космонавты выходят в открытый космос, Авария(с некоторой долей вероятности) – страховая выплачивает много денег.

**Задача 6.3**

Доработать предыдущую задачу с использованием синхронизации потоков. На форме должно быть не менее 4 моделей. Ограничения накладываются на классы, которые реализуют интерфейсы. Для 4 моделей должно быть 2 объекта данных классов в сумме. При возникновении какого-то события 1 из объектов «лочится» и не доступен для использования в других моделях.