**Задание 11.**

В интегрированной среде *Microsoft Visual Studio* на языке *Visual C#* в консольном режиме составить программное обеспечение для решения типовых задач программирования по тематике «Методы» («*Methods*»), заданных по варианту.

Следить за обозначенной классификацией методов («типизированный» / «нетипизированный» [«пустой»]) в условиях индивидуальных задач, выданных по варианту.

Следить за обозначенной классификацией параметров методов: «возвращаемые по ссылке» («*referenced*») / «инициализируемые методом» («*output*»).

**Элементы рабочего задания:**

1. **Учесть все возможные ограничения**, накладываемые не только на переменные, но и на участвующие в расчёте функциональные зависимости**[[1]](#footnote-1)**.

2. **Ввести контроль исходных данных**. Реализовать схему «ввод до победного» без возможности прерывания. При некорректном вводе исходных данных заставлять пользователя вводить сведения до тех пор, пока он не введёт их корректно. Продумать побуждающие сообщения-подсказки, направляющие пользователя ко вводу корректных значений.

Использовать для контроля исходных данных различные состояния метода «*TryParse*», входящего в перечень доступных компонентов интересующего значащего (*valuable*) типа данных, например: *int, byte, float, double*.

3. **Подобрать корректные тестовые примеры в необходимом количестве[[2]](#footnote-2).** Принять во внимание тот факт, что все необходимые для тестирования значения могут быть занесены сразу в один единственный многомерный массив (рациональная размерность массива для тестирования – не менее трёх элементов в каждом из направлений). Помнить, что для решения задачи в общем виде размерность не должна оказаться одинаковой в каждом из измерений, кроме случаев, когда с точки зрения математики принципиальны квадратные и кубические массивы.

4. **Помнить о точности производимых вычислений**, например:

|  |  |
| --- | --- |
| **«При заданной точности  , но ,**  **то есть , а ».** | (1) |

5. Код составленного программного обеспечения **сопроводить комментариями** (элементы программистской этики)**[[3]](#footnote-3)**.

6. **Отчёт по решённой задаче составить** по схеме, изложенной в методических указаниях**[[4]](#footnote-4)**. Для удобства работы обучающихся далее представлена памятка основных разделов отчёта:

0. Титульный лист

1. Цель работы

2. Формулировка задачи

3. Блок-схема алгоритма

4. Подбор тестовых примеров

5. Листинг (код) программы

6. Расчёт тестовых примеров на ПК

7. Вывод по работе

7. **Схему алгоритма оформлять** согласно правилам**[[5]](#footnote-5)**. Помнить, что при работе с методами необходимо вводить нумерацию блоков и рассматривать содержимое методов в отдельных схемах алгоритма. Помнить, что оконечные блоки методов помимо надписей «Начало» и «Конец» должны содержать сигнатуры методов: либо краткие, в которых присутствует только наименование метода без перечисления формальных параметров (все параметры перечисляются далее в блоке «Данные» внутри метода); либо полные, в которых перечисляются все формальные параметры и их типы данных (в этом случае блок «Данные» внутри метода не размещается и не заполняется).

8. Материалы, подготавливаемые **к отправке по электронной почте**, оформляются согласно **Приложению 4** Сборника задач**[[6]](#footnote-6)**. Если материалы направляются на проверку через платформу *GitHub*, то **Приложением 4** разрешается не руководствоваться.

9. Настоятельно рекомендовано перед решением задачи **ознакомиться с рекомендациями**, собранными в **Приложении 3** Сборника задач**[[7]](#footnote-7)**.

10. Не забывать **давать значимые имена переменным**, проектам, решениям, файлам, методам и другим разрабатываемым компонентам**[[8]](#footnote-8)**. При нежелании использовать значимые имена переменных в отчёте необходимо составить в разделе, содержащем формулировку задачи, таблицу соответствия вида:

Таблица 1 – Пример таблицы соответствия переменных, используемых в программе

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Наименование переменной в программе** | **Смысловое содержание используемой переменной** | **Тип данных** |
| 1 | *a* | Ускорение | Вещественное |
| 2 | *v* | Скорость | Вещественное |
| 3 | *S* | Пройдённый путь | Вещественное |
| 4 | *t* | Затраченное время | Вещественное |
| 5 | *x* | Абсцисса точки на Декартовой плоскости | Целое |
| 6 | *y* | Ордината точки на Декартовой плоскости | Целое |
| 7 | *z* | Аппликата точки в пространстве | Целое |

**Внимание! При выявлении противоречивых условий среди вариантов индивидуального задания не предпринимать самостоятельных решений по устранению противоречий, а обратиться за консультацией к преподавателю, читающему курс «Программирования и основ алгоритмизации».**

**Задание состоит только из Индивидуальной части. Общая часть в нём отсутствует.**

**Вариант №1.** Составить нетипизированный (пустой) метод увеличения/уменьшения размерности исходного двумерного массива вещественных чисел на единицу с сохранением его содержимого. В качестве параметров метода направляются двумерный массив вещественных чисел (вернуть по ссылке) и логический признак направления изменения размерности. При наименовании метода использовать префикс «*\_a*».

**Вариант №2.** Составить типизированный метод для вывода элементов двумерного массива змейкой, начиная с позиции, обусловленной наличием логического признака (флага). Для печати из правого нижнего угла влево использовать значение *true*, для печати из левого верхнего угла – значение *false*. Отвечать логическим признаком наличия/отсутствия не размещённых на экране элементов массива. При наименовании метода использовать префикс «*\_b*».

**Вариант №3.** Составить комбинированный нетипизированный (пустой) метод масштабирования псевдослучайного вещественного числа со смещением диапазона относительно нулевой отметки. В качестве параметров метода подаётся неинициализированная вещественная переменная одинарной точности (вернуть по ссылке), целочисленный коэффициент усиления (масштаба), а также целочисленная величина сдвига относительно нулевой отметки. При наименовании метода использовать префикс «*\_c*».

**Вариант №4.** Составить типизированный метод сортировки по возрастанию/убыванию элементов указанной строки/столбца исходного двумерного массива. В качестве параметров метода подаётся двумерный массив, логический признак (флаг) направления сортировки, индекс указанной размерности массива, логический признак размерности сортировки (*true* – строка, *false* – столбец). Выполнить сортировку всех строк и столбцов без перезаписи исходного массива. При наименовании метода использовать префикс «*\_d*».

**Вариант №5.** Составить нетипизированный (пустой) метод формирования строкового массива по некоторому, указанному разделителю. Метод должен работать по аналогии с имеющимся стандартным методом *Split()* языка *Visual C#*. При разбиении на элементы разделитель сохранять в конце текущей строки, принудительно удалять из массива получающиеся пустые строки. Получившийся массив вернуть по ссылке в объявленную, но не инициализированную переменную. При наименовании метода использовать префикс «*\_e*».

**Вариант №6.** Создать типизированный метод перевода десятичного целочисленного значения в строковый двоичный или шестнадцатеричный эквивалент в зависимости от поступившего на вход логического признака (флага). *True – HEX*, *false – BIN*. При наименовании метода использовать префикс «*\_f*».

**Вариант №7.** Составить нетипизированный (пустой) метод для сортировки по возрастанию/убыванию элементов указанной строки/столбца исходного двумерного массива вещественных чисел. В качестве параметров метода подаётся двумерный массив вещественных чисел (вернуть по ссылке), логический признак направления сортировки, индекс указанной размерности массива, логический признак размерности сортировки (*true* – строка, *false* – столбец). Выполнить сортировку всех строк и столбцов. Сначала строк, затем столбцов. При наименовании метода использовать префикс «*\_g*».

**Вариант №8.** Составить типизированный метод сортировки по возрастанию/убыванию элементов исходного двумерного массива. В качестве параметров метода подаётся двумерный массив, а также логический признак направления сортировки.

Выполнить сортировку элементов трёх различных массивов различной размерности в различных комбинациях:

а) все по возрастанию;

б) все по убыванию;

в) с чередованием возрастание/убывание;

г) с чередованием убывание/возрастание.

При наименовании метода использовать префикс «*\_h*».

**Вариант №9.** Составить нетипизированный (пустой) метод замены одной строковой конструкции другой. Метод должен работать по аналогии с имеющимся стандартным методом *Replace()* языка *Visual C#*. Параметры те же, что и в *Replace*, но записаны должны быть в обратной последовательности: на первой позиции конструкция, на которую необходимо заменить, а второй позиции – конструкция, которую необходимо заменить. Перечень параметров должен быть дополнен входной строкой (вернуть по ссылке). При наименовании метода использовать префикс «*\_i*».

**Вариант №10.** Составить типизированный метод поиска чисел Фибоначчи. В качестве параметра метода подаётся *M*-е по счёту число Фибоначчи. Вывести последовательно 7, 4 и 13 числа Фибоначчи и найти сумму цифр всех этих чисел. При наименовании метода использовать префикс «*\_j*».

**Вариант №11.** Составить нетипизировнный (пустой) метод сортировки по возрастанию/убыванию элементов исходного двумерного массива целых чисел. В качестве параметров метода подаётся двумерный массив (вернуть по ссылке), а также логический признак направления сортировки. Выполнить сортировку элементов четырёх различных массивов различной размерности в различных комбинациях: а) все по возрастанию; б) все по убыванию; в) с чередованием возрастание/убывание; г) с чередованием убывание/возрастание. При наименовании метода использовать префикс «*\_k*».

**Вариант №12.** Составить типизированный метод формирования строкового массива по некоторому, указанному разделителю. Метод должен работать по аналогии с имеющимся стандартным методом *Split()* языка *Visual C#*. При разбиении на элементы разделитель сохранять в конце текущей строки, принудительно удалять из массива получающиеся пустые строки. При наименовании метода использовать префикс «*\_l*».

**Вариант №13.** Составить нетипизированный (пустой) метод шифрования и дешифрирования сообщения. В качестве параметров метода подаётся строка (вернуть по ссылке) текста и логический признак направления шифрования. Шифрование заключается в том, что каждый второй символ при движении с начала сообщения обменивается с каждым вторым символом с конца сообщения. При наименовании метода использовать префикс «*\_m*».

**Вариант №14.** Составить типизированный метод расчёта расстояния до точки, заданной своими координатами *x, y, z*. Посчитать расстояния до трёх различных точек с произвольно формируемыми координатами. При наименовании метода использовать префикс «*\_n*».

**Вариант №15.** Составить нетипизированный (пустой) метод шифрования и дешифрирования сообщения. В качестве параметров метода подаётся строка (вернуть по ссылке) текста и логический признак направления шифрования. Шифрование заключается в том, что каждый символ кодируется его порядковым номером в алфавите. Если символ кириллический, то слева от него записывается «*%*», если символ латинский – «*&*». Численные значения в сообщении не изменять и слева от них ставить «*\**». При наименовании метода использовать префикс «*\_o*».

**Вариант №16.** Создать типизированный метод заполнения двумерного массива псевдослучайными числами. Наличие на входе метода определённого значения логического признака указывает на: *true* – формирование уникальных элементов без повторов, *false* – формирование произвольных элементов, включая повторные вхождения. При наименовании метода использовать префикс «*\_p*».

**Вариант №17.** Составить нетипизированный (пустой) метод увеличения/уменьшения размерности исходного одномерного массива символов на единицу с сохранением его содержимого. В качестве параметров метода направляются одномерный массив символов (вернуть по ссылке) и логический признак направления изменения размерности. При наименовании метода использовать префикс «*\_q*».

**Вариант №18.** Составить типизированный метод расчёта корней квадратного уравнения. В качестве входных параметров метода подаются коэффициенты *a, b, с*. Выполнить расчёт для различных исходов решения в рамках одного запуска программы. При наименовании метода использовать префикс «*\_r*».

**Вариант №19.** Составить нетипизированный (пустой) метод определения следа матрицы. Матрица подаётся в качестве параметра метода и дополняется неинициализированной вещественной переменной одинарной точности (вернуть по ссылке). Применить созданный метод к расчёту произведения следов единичной матрицы той же размерности, что и матрица *A*. Элементы последней формируются по правилу *A(i, j) = (j + 1) + c \* i*, где *c* – размерность матрицы. Обратить внимание на то, что работа принципиально должна быть выполнена над квадратными матрицами. При наименовании метода использовать префикс «*\_s*».

**Вариант №20.** Составить типизированный метод формирующий строковый адрес ячейки *Microsoft Excel*, расположенной на пересечении *x* столбца и y строки. *x* и *y* – входные целочисленные параметры. При наименовании метода использовать префикс «*\_t*».

**Вариант №21.** Составить нетипизированный (пустой) метод суммирования всех элементов заданного одномерного массива вещественных чисел. Одномерный массив вещественных чисел подаётся в качестве параметра метода и дополняется неинициализированной вещественной переменной одинарной точности (вернуть по ссылке). Сложить между собой элементы трёх различных одномерных массивов вещественных чисел различной размерности. При наименовании метода использовать префикс «*\_u*».

**Вариант №22.** Составить типизированный метод для перевода времени, заданного в секундах одного из целых типов, в астрономическое время строкового типа. При наименовании метода использовать префикс «*\_v*».

**Вариант №23.** Составить нетипизированные (пустые) методы вывода элементов двумерного массива целых неотрицательных чисел спиралью от центра, а также «змейкой». Двумерный массив подаётся в качестве параметра метода, вводятся логические признаки для ожидания нажатия на любую клавишу пользователем и необходимости очистки содержимого консоли перед выводом. При наименовании метода использовать префикс «*\_w*».

**Вариант №24.** Составить типизированный метод «Штрих Шеффера» (И-НЕ). В качестве параметров метода подаются две логические переменные. Составить таблицу соответствия, записанную в матрицу, и вывести её в консоль. При наименовании метода использовать префикс «*\_x*».

**Вариант №25.** Составить нетипизированный (пустой) метод ввода элементов двумерного массива вещественных чисел. Двумерный массив вещественных чисел подаётся в качестве параметра метода (вернуть по ссылке), вводятся логические признаки для очистки консоли после каждого введённого элемента, вывода полученного массива после завершения ввода всех элементов, заполнения массива псевдослучайным образом. При наименовании метода использовать префикс «*\_y*».

**Вариант №26.** Создать типизированный метод перевода строкового значения в целочисленный десятичный эквивалент. В зависимости от системы счисления, поданной на вход, необходимо указывать соответствующий логический признак (флаг). *True – BIN*, *false – HEX*. При наименовании метода использовать префикс «*\_z*».

**Вариант №27.** Составить нетипизированный (пустой) метод для сортировки по возрастанию/убыванию элементов указанной строки/столбца исходного двумерного массива целых чисел. В качестве параметров метода подаётся двумерный массив целых чисел (вернуть по ссылке), логический признак направления сортировки, индекс указанной размерности массива, численный признак размерности сортировки (*1* – строка, 2 – столбец). Выполнить сортировку всех строк и столбцов. Сначала столбцов, затем строк. При наименовании метода использовать префикс «*\_A*».

**Вариант №28.** Составить типизированный метод для перевода времени, заданного строкой, в секунды (значение флага – *true*) и «минуты:секунды» (значение флага – *false*). При наименовании метода использовать префикс «*\_B*».

**Вариант №29.** Составить нетипизированный (пустой) метод для переключения между пунктами меню. Нажатие на «1» соответствует выбору первого пункта, нажатие на «2» – второго пункта, нажатие на «3» – выбору третьего пункта. Подтверждение выбора реализуется по нажатии клавиши «*Enter*». Выполнение программы прекращается по нажатии клавиши «*Esc*». За «1» закрепляется операция вывода значения числа «Пи» с точностью до 8 знаков, за «2» – операция вывода синуса от «Пи / 2», за «3» – операция вывода косинуса от «Пи / 2». При наименовании метода использовать префикс «*\_C*».

**Вариант №30.** Составить типизированный метод поиска экстремального значения в двумерном массиве. Логический признак (флаг), поданный на вход, должен определить алгоритм поиска: *true* – максимальное значение, *false* – минимальное значение. При наименовании метода использовать префикс «*\_D*».

**Вариант №31.** Составить нетипизированный (пустой) метод для вывода элементов двумерного массива символов змейкой, начиная с позиции обусловленной наличием логического признака (флага). Для печати из правого нижнего угла влево использовать значение *true*, для печати из левого верхнего угла – значение *false*. При наименовании метода использовать префикс «*\_E*».

**Вариант №32.** Составить типизированный метод расчёта факториала числа *M*. *M* подаётся в качестве параметра метода. Применить созданный метод для расчёта суммы ряда *1 – (x^2 / 2!) + (x^4 / 4!) – (x^6 / 6!)* …, состоящего из *N* элементов. При наименовании метода использовать префикс «*\_F*».

**Вариант №33.** Составить нетипизированный (пустой) метод расчёта расстояния до точки, заданной своими координатами *x*, *y*, *z*. Вернуть результат в заранее не инициализированную переменную вещественного типа. При наименовании метода использовать префикс «*\_G*».

**Вариант №34.** Составить типизированный метод определения следа матрицы. Матрица подаётся в качестве параметра метода. Применить созданный метод к расчёту произведения следов единичной матрицы той же размерности, что и матрица, элементы которой формируются по правилу: *A(i, j) = (j + 1) + c \* i*, где *c* – размерность матрицы. Обратить внимание на то, что работа принципиально должна быть выполнена над квадратными матрицами. При наименовании метода использовать префикс «*\_H*».

**Вариант №35.** Создать нетипизированный (пустой) метод перевода строкового значения в целочисленный десятичный эквивалент (вернуть по ссылке). Содержимое метода должно распознать значение двоичной или шестнадцатеричной системы счисления и применить к нему определённый алгоритм преобразования. Пользователь не участвует в процессе прямого указания исходной системы счисления. При наименовании метода использовать префикс «*\_I*».

1. Пример корректного подбора тестовых примеров в нужном количестве представлен в разделе 7 Сборника задач [Сафронов, А. И. Получение первичных профессиональных умений и навыков научно-исследовательской деятельности: Сборник задач для проведения аудиторных занятий по учебной практике / А. И. Сафронов, Н. Н. Зольникова, В. Г. Новиков. – Москва: РУТ. – 2019. – 91 с. – *EDN SXMWOD*]. Скачать можно на сайте Национальной Электронной Библиотеки (НЭБ) *elibrary.ru*: [*https://www.elibrary.ru/item.asp?id=46307421*](https://www.elibrary.ru/item.asp?id=46307421). [↑](#footnote-ref-1)
2. То же, что и сноска 1. [↑](#footnote-ref-2)
3. Пункт 6 на странице 391 [Сафронов, А. И. Способы проектирования эргономичных графических пользовательских интерфейсов для интеллектуальных транспортных систем / А. И. Сафронов // Интеллектуальные транспортные системы: Материалы III Международной научно-практической конференции, Москва, 30 мая 2024 года. – Москва: РУТ. – 2024. – С. 385-394. – *DOI 10.30932/9785002446094-2024-385-394*. – *EDN TGVERU*]. Скачать можно на сайте Национальной Электронной Библиотеки (НЭБ) *elibrary.ru*: [*https://www.elibrary.ru/item.asp?id=68533620*](https://www.elibrary.ru/item.asp?id=68533620). [↑](#footnote-ref-3)
4. Пример оформления отчётной документации [Сафронов, А. И. Составление отчётной документации по решённым задачам алгоритмизации и программирования: Учебно-методическое пособие для проведения аудиторных занятий по учебной практике / А. И. Сафронов, Н. Н. Зольникова, В. Г. Новиков. – Москва: РУТ. – 2018. – 83 с. – *EDN WNAIFP*]. Скачать можно на сайте Национальной Электронной Библиотеки (НЭБ) *elibrary.ru*: [*https://www.elibrary.ru/item.asp?id=46271697*](https://www.elibrary.ru/item.asp?id=46271697). [↑](#footnote-ref-4)
5. Правила изложены в разделе 8 того же учебно-методического пособия, что указано по сноске 4. [↑](#footnote-ref-5)
6. То же, что и сноска 1. [↑](#footnote-ref-6)
7. То же, что и сноска 1. [↑](#footnote-ref-7)
8. То же, что и сноска 3. [↑](#footnote-ref-8)