Требования к программе, общие для всех вариантов

Определить класс **Person**, который имеет

* закрытое поле типа string, в котором хранится имя;
* закрытое поле типа string, в котором хранится фамилия;
* закрытое поле типа System.DateTime для даты рождения.

В классе **Person** определить конструкторы:

* конструктор c тремя параметрами типа string, string, DateTime для инициализации всех полей класса;
* конструктор без параметров, инициализирующий все поля класса некоторыми значениями по умолчанию.

В классе **Person** определить свойства c методами get и set:

* свойство типа string для доступа к полю с именем;
* свойство типа string для доступа к полю с фамилией;
* свойство типа DateTime для доступа к полю с датой рождения;
* свойство типа int c методами get и set для получения информации(get) и изменения (set) года рождения в закрытом поле типа DateTime, в котором хранится дата рождения.

В классе **Person** определить

* перегруженную(override) версию виртуального метода string ToString() для формирования строки со значениями всех полей класса;
* виртуальный метод string ToShortString(), который возвращает строку, содержащую только имя и фамилию.

Cравнить время, необходимое для выполнения операций с элементами одномерного, двумерного прямоугольного и двумерного ступенчатого массивов с одинаковым числом элементов.

Для этого в методе Main() создать

* одномерный массив;
* двумерный прямоугольный массив;
* двумерный ступенчатый массив.

Тип элементов массивов зависит от варианта лабораторной работы. Массивы должны иметь одинаковое число элементов. Если число строк в двумерном прямоугольном массиве равно nrow, а число столбцов ncolumn, то одномерный массив должен содержать nrow\*ncolumn элементов, в двумерном ступенчатом массиве общее число элементов также должно быть равно nrow\*ncolumn.

Значения nrow и ncolumn вводятся в процессе работы приложения в виде одной строки с разделителями. В приглашении, которое получает пользователь, должна быть информация о том, какие символы можно использовать как разделители, число разделителей должно быть больше 1. С помощью метода Split класса System.String приложение разбирает введенную пользователем текстовую строку с информацией о числе строк и числе столбцов двумерного массива и присваивает значения переменным, которые содержат значения nrow и ncolumn. В первой лабораторной работе не требуется обрабатывать ошибки ввода, предполагается, что пользователь правильно ввел данные.

Приложение распределяет память для всех массивов и инициализирует элементы массивов. Для инициализации элементов можно использовать конструктор без параметров.

Для всех элементов массивов выполняется одна и та же операция, например, присваивается значение одному из свойств, определенных для элементов массива. В лабораторной работе требуется сравнить время выполнения этой операции для одномерного, двумерного прямоугольного и двумерного ступенчатого массивов с одинаковым числом элементов.

Для измерения времени выполнения операций можно использовать свойство Environment.TickCount. Cтатическое свойство TickCount класса Environment имеет тип int, использует информацию системного таймера и содержит время в миллисекундах, которое прошло с момента перезагрузки компьютера.

Чтобы получить время выполнения некоторого блока кода, необходимо вызвать Environment.TickCount непосредственно перед блоком и сразу же после последнего оператора блока и взять разность значений.

В блоке кода, для которого измеряется время, не должно быть операций распределения памяти для массивов, инициализации элементов массивов и операций вывода данных на консоль. Блоки кода должны содержать только операции с элементами массива.

Вычисленные значения времени выполнения операций для трех типов массивов, а также число строк nrow и столбцов ncolumn выводятся на консоль. Вывод должен быть подписан, т.е. вывод должен содержать информацию о том, к какому типу массива отвечает выведенное значение.

ВАРИАНТ 1

Определить тип **Education** - перечисление(enum)со значениями Specialist, Вachelor, SecondEducation.

Определить класс **Exam**, который имеет три открытых автореализуемых свойства, доступных для чтения и записи:

* свойство типа string, в котором хранится название предмета;
* свойство типа int, в котором хранится оценка;
* свойство типа System.DateTime для даты экзамена.

В классе **Exam** определить:

* конструктор с параметрами типа string, int и DateTime для инициализации всех свойств класса;
* конструктор без параметров, инициализирующий все свойства класса некоторыми значениями по умолчанию;
* перегруженную(override) версию виртуального метода string ToString() для формирования строки со значениями всех свойств класса.

Определить класс **Student**, который имеет

* закрытое поле типа Person, в котором хранятся данные студента;
* закрытое поле типа Education для информации о форме обучения;
* закрытое поле типа int для номера группы;
* закрытое поле типа Exam [] для информации об экзаменах, которые сдал студент.

В классе **Student** определить конструкторы:

* конструктор c параметрами типа Person, Education, int для инициализации соответствующих полей класса;
* конструктор без параметров, инициализирующий поля класса значениями по умолчанию.

В классе **Student** определить свойства c методами get и set:

* свойство типа Person для доступа к полю с данными студента;
* свойство типа Education для доступа к полю с формой обучения;
* свойство типа int для доступа к полю с номером группы;
* свойство типа Exam [] для доступа к полю со списком экзаменов.

В классе **Student** определить

свойство типа double ( только с методом get), в котором вычисляется средний балл как среднее значение оценок в списке сданных экзаменов;

индексатор булевского типа (только с методом get) с одним параметром типа Education; значение индексатора равно true, если значение поля с формой обучения студента совпадает со значением индекса, и false в противном случае;

метод void AddExams ( params Exam [] ) для добавления элементов в список экзаменов;

перегруженную версию виртуального метода string ToString() для формирования строки со значениями всех полей класса, включая список экзаменов;

виртуальный метод string ToShortString(), который формирует строку со значениями всех полей класса без списка экзаменов, но со значением среднего балла.

В методе **Main()**

1. Создать один объект типа Student, преобразовать данные в текстовый вид с помощью метода ToShortString() и вывести данные.
2. Вывести значения индексатора для значений индекса Education.Specialist, Education.Bachelor и Education.SecondEducation.
3. Присвоить значения всем определенным в типе Student свойствам, преобразовать данные в текстовый вид с помощью метода ToString() и вывести данные.
4. C помощью метода AddExams( params Exam[] ) добавить элементы в список экзаменов и вывести данные объекта Student, используя метод ToString().
5. Сравнить время выполнения операций с элементами одномерного, двумерного прямоугольного и двумерного ступенчатого массивов с одинаковым числом элементов типа Exam.

ВАРИАНТ 2

Определить тип **Frequency -** перечисление(enum)со значениями Weekly, Monthly, Yearly.

Определить класс **Article**, который имеет три открытых автореализуемых свойства, доступных для чтения и записи:

* свойство типа Person, в котором хранятся данные автора статьи;
* свойство типа string для названия статьи;
* свойство типа double для рейтинга статьи.

В классе **Article** определить:

* конструктор c параметрами типа Person, string, double для инициализации всех свойств класса;
* конструктор без параметров, инициализирующий все свойства класса некоторыми значениями по умолчанию;
* перегруженную(override) версию виртуального метода string ToString() для формирования строки со значениями всех свойств класса.

Определить класс **Magazine**, который имеет

* закрытое поле типа string c названием журнала;
* закрытое поле типа Frequency с информацией о периодичности выхода журнала;
* закрытое поле типа DateTime c датой выхода журнала;
* закрытое поле типа int с тиражом журнала;
* закрытое поле типа Article[] со списком статей в журнале.

В классе **Magazine** определить конструкторы:

* конструктор с параметрами типа string, Frequency, DateTime, int для инициализации соответствующих полей класса;
* конструктор без параметров, инициализирующий поля класса значениями по умолчанию.

В классе **Magazine** определить свойства c методами get и set:

* свойство типа string для доступа к полю с названием журнала;
* свойство типа Frequency для доступа к полю с информацией о периодичности выхода журнала;
* свойство типа DateTime для доступа к полю c датой выхода журнала;
* свойство типа int для доступа к полю с тиражом журнала;
* свойство типа Article[] для доступа к полю со списком статей.

В классе **Magazine** определить

свойство типа double ( только с методом get), в котором вычисляется среднее значение рейтинга в списке статей;

индексатор булевского типа (только с методом get) с одним параметром типа Frequency; значение индексатора равно true, если значение поля типа Frequency совпадает со значением индекса, и false в противном случае;

метод void AddArticles (params Article[]) для добавления элементов в список статей в журнале;

перегруженную версию виртуального метода string ToString() для формирования строки со значениями всех полей класса, включая список статей;

виртуальный метод string ToShortString(), который формирует строку со значениями всех полей класса без списка статей, но со значением среднего рейтинга статей.

В методе **Main()**

1. Создать один объект типа Magazine, преобразовать данные в текстовый вид с помощью метода ToShortString() и вывести данные.
2. Вывести значения индексатора для значений индекса Frequency.Weekly, Frequency.Monthly и Frequency.Yearly.
3. Присвоить значения всем определенным в типе Magazine свойствам, преобразовать данные в текстовый вид с помощью метода ToString() и вывести данные.
4. C помощью метода AddArticles( params Article[] ) добавить элементы в список статей и вывести данные объекта Magazine, используя метод ToString().
5. Сравнить время выполнения операций с элементами одномерного, двумерного прямоугольного и двумерного ступенчатого массивов с одинаковым числом элементов типа Article.

ВАРИАНТ 3

Определить тип **TimeFrame** - перечисление(enum)со значениями Year, TwoYears, Long.

Определить класс **Paper**, который имеет три открытых автореализуемых свойства, доступных для чтения и записи:

* свойство типа string, в котором хранится название публикации;
* свойство типа Person для автора публикации;
* свойство типа DateTime c датой публикации.

В классе **Paper** определить

* конструктор c параметрами типа string, Person, DateTime для инициализации всех свойств класса;
* конструктор без параметров, инициализирующий все свойства класса некоторыми значениями по умолчанию;
* перегруженную(override) версию виртуального метода string ToString() для формирования строки со значениями всех полей класса.

Определить класс **ResearchTeam**, который имеет

* закрытое поле типа string c названием темы исследований;
* закрытое поле типа string с названием организации;
* закрытое поле типа int – регистрационный номер;
* закрытое поле типа TimeFrame для информации о продолжительности исследований;
* закрытое поле типа Paper[], в котором хранится список публикаций.

В классе **ResearchTeam** определить конструкторы:

* конструктор c параметрами типа string, string, int, TimeFrame для инициализации соответствующих полей класса;
* конструктор без параметров, инициализирующий поля класса значениями по умолчанию.

В классе **ResearchTeam** определить свойства c методами get и set:

* свойство типа string для доступа к полю с названием темы исследований;
* свойство типа string для доступа к полю с названием организации;
* свойство типа int для доступа к полю с номером регистрации;
* свойство типа TimeFrame для доступа к полю с продолжительностью исследований;
* свойство типа Paper[] для доступа к полю со списком публикаций по теме исследований.

В классе **ResearchTeam** определить

свойство типа Paper ( только с методом get), которое возвращает ссылку на публикацию с самой поздней датой выхода; если список публикаций пустой, свойство возвращает значение null;

индексатор булевского типа (только с методом get) с одним параметром типа TimeFrame; значение индексатора равно true, если значение поля с информацией о продолжительности исследований совпадает со значением индекса, и false в противном случае;

метод void AddPapers ( params Paper[] ) для добавления элементов в список публикаций;

перегруженную версию виртуального метода string ToString() для формирования строки со значениями всех полей класса, включая список публикаций;

виртуальный метод string ToShortString(), который формирует строку со значениями всех полей класса без списка публикаций.

В методе **Main()**

1. Создать один объект типа ResearchTeam, преобразовать данные в текстовый вид с помощью метода ToShortString() и вывести данные.
2. Вывести значения индексатора для значений индекса TimeFrame.Year, TimeFrame.TwoYears, TimeFrame.Long.
3. Присвоить значения всем определенным в типе ResearchTeam свойствам, преобразовать данные в текстовый вид с помощью метода ToString() и вывести данные.
4. С помощью метода AddPapers (params Paper []) добавить элементы в список публикаций и вывести данные объекта ResearchTeam.
5. Вывести значение свойства, которое возвращает ссылку на публикацию с самой поздней датой выхода;
6. Сравнить время выполнения операций с элементами одномерного, двумерного прямоугольного и двумерного ступенчатого массивов с одинаковым числом элементов типа Paper.