# Практическая работа №9 «Классы и объекты»

## Часть 1.

1. Реализовать (в отдельном файле) определение нового класса/типа данных (закрытые атрибуты, свойства, конструкторы, инициализация и вывод атрибутов).
2. Для демонстрации работы с объектами написать главную функцию, в которой создаются объекты класса и выводится информация, которая содержится в атрибутах.
3. Написать функцию, реализующую указанное в варианте действие. Рассмотреть два варианта:

1) статическую функцию;

2) метод класса.

В основной функции продемонстрировать работу функции.

1. Используя статическую компоненту класса подсчитать количество созданных в программе объектов.

**Варианты:**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название класса | Атрибуты (поля) | Методы |
| 1 | Time | int hours,  int minutes | Добавление минут (int) к объекту типа Time (учесть, что минут не может быть больше 59).  Результат должен быть типа Time. |
| 2 | Money | int rubles, int kopeks | Добавление копеек (int) к объекту типа Money (учесть, что копеек не может быть больше 99).  Результат должен быть типа Money. |
| 3 | Point | double x, double y (координаты точки) | Вычислить расстояние от одной точки до другой.  Результат должен быть типа double. |
| 4 | Triangle | double a, double b, double c (длины сторон) | Вычислить площадь треугольника. Результат должен быть типа double. |
| 5 | Diapason | double x, double y (начало и конец диапазона) | Определить попадает ли заданное число в диапазон. Результат должен быть типа bool. |
| 6 | Uravnenie | double a, double b, double c  (коэффициенты уравнения) | Вычисление корней квадратного уравнения. Результат должен быть типа double. |
| 7 | Time | int hours,  int minutes | Вычитание переменной типа Time (учесть, что часов и минут не может быть меньше 0). Результат должен быть типа Time. |
| 8 | Money | int rubles, int kopeks | Вычитание переменной типа Money (учесть, что рублей и копеек не может быть меньше 0). Результат должен быть типа Money. |
| 9 | Point | double x, double y (координаты точки) | Вычислить расстояние от точки до начала координат. Результат должен быть типа double. |
| 10 | Triangle | double a, double b, double c (длины сторон треугольника) | Определить существует ли треугольник с такими длинами сторон. Результат должен быть типа bool. |
| 11 | Diapason | double x, double y (начало и конец диапазона) | Определить пересекаются ли заданные диапазоны. Результат должен быть типа bool. |
| 12 | Time | int hours  int minutes | Вычитание минут (int) из объекта типа Time (учесть, что минут не может быть больше 59 и меньше 0).  Результат должен быть типа Time. |
| 13 | Money | int rubles, int kopeks | Вычитание копеек (int) из объекта типа Money (учесть, что рублей и копеек не может быть меньше 0). Результат должен быть типа Money. |
| 14 | Time | int hours  int minutes | Вычитание минут (int) из объекта типа Time (учесть, что часов и минут не может быть меньше 0). Результат должен быть типа Time. |
| 15 | Money | int rubles, int kopeks | Сравнение двух объектов типа Money. Результат должен быть типа bool. |

### Часть 2.

1. Добавить к реализованному классу указанные в варианте перегруженные операции.
2. Написать демонстрационную программу, в которой создаются объекты пользовательских классов и выполняются указанные операции.

**Варианты:**

|  |  |  |
| --- | --- | --- |
| № | Название класса | Методы |
| 1 | Time | Унарные операции:  ++ добавление минуты к объекту типа Time (учесть, что минут не может быть больше 59).  -- вычитание минуты из объекта типа Time (учесть, что часов и минут не может быть меньше 0).  Операции приведения типа:  int (явная) – результатом является количество часов (минуты отбрасываются);  bool (неявная) – результатом является true, если часы и минуты не равны нулю и false в противном случае.  Бинарные операции:  + Time t, целое число (минуты) (лево- и право- сторонние операции). Результат должен быть типа Time.  – Time t, целое число (минуты) (лево- и право- сторонние операции). Результат должен быть типа Time. |
| 2 | Money | Унарные операции:  ++ добавление копейки к объекту типа Money (учесть, что копеек не может быть больше 99).  -- вычитание копейки из объекта типа Money (учесть, что рублей и копеек не может быть меньше 0).  Операции приведения типа:  int (явная) – результатом является количество рублей (копейки отбрасываются);  double (неявная) – результатом является копейки, рубли отбрасываются, результат <1.  Бинарные операции:  + Money m, целое число (лево- и право- сторонние операции), учесть, что копеек не может быть больше 99. Результат должен быть типа Money.  – Money m, целое число (лево- и право- сторонние операции), учесть, что рублей и копеек не может быть меньше 0. Результат должен быть типа Money. |
| 3 | Point | Унарные операции:  ++ увеличить координаты x на 1,  -- уменьшение координаты х на 1.  Операции приведения типа:  int (явная) – результатом является целая часть координаты х;  double (неявная) – результатом является координата y.  Бинарные операции:  + Point p – вычисляется расстояние до точки p. Результатом должно быть вещественное число.  + целое число (лево- и право- сторонние операции). Увеличивается координата х, результатом является объект типа Point. |
| 4 | Triangle | Унарные операции:  ++ увеличивает стороны треугольника на 1;  -- уменьшает стороны треугольника на 1;  Операции приведения типа:  double (явная) – результатом является площадь треугольника, если треугольник существует и отрицательное число в противном случае;  bool (неявная) – результатом является true, если треугольник с такими длинами сторон существует и false в противном случае.  Бинарные операции:  <= Triangle t1, Triangle t2 – сравнивает площади треугольников;  >= Triangle t1, Triangle t2 - сравнивает площади треугольников. |
| 5 | Diapason | Унарные операции:  ! - вычислить длину диапазона, результат должен быть типа double;  ++ увеличить координаты диапазона на 1.  Операции приведения типа:  int (явная) – результатом является целая часть координаты х;  double (неявная) – результатом является координата y.  Бинарные операции:  + int d, целое число (лево- и право-сторонняя операция, координаты увеличиваются на число d);  < Diapason d, целое число – результат равен true, если целое число попадает в заданный диапазон и false – в противном случае. |
| 6 | Uravnenie | Унарные операции:  ++ увеличивает коэффициенты уравнения на 1;  -- уменьшает коэффициенты уравнения на 1.  Операции приведения типа:  double (неявная) – результатом является один из корней уравнения, если корни существуют и 0 в противном случае.  bool (явная) – результатом является true, если корни существуют и false в противном случае;  Бинарные операции  == Uravnenie t1, Uravnenie t2 - уравнения равны, если равны их коэффициенты;  != Uravnenie t1, Uravnenie t2 - треугольники не равны, если не равны их коэффициенты. |
| 7 | Time | Унарные операции:  ++ добавление минуты к объекту типа Time.  -- вычитание минуты из объекта типа Time (учесть, что минут не может быть меньше 0).  Операции приведения типа:  int (неявная) – результатом является количество минут (время переводится в минуты);  bool (явная) – результатом является true, если часы и минуты не равны нулю и false в противном случае.  Бинарные операции:  < Time t1, Time t2 – время переводится в минуты, результатом является true, если количество минут в левом операнде меньше, чем количество минут в правом операнде и false – в противном случае.  > Time t1, Time t2 - время переводится в минуты, результатом является true, если количество минут в левом операнде больше, чем количество минут в правом операнде и false – в противном случае. |
| 8 | Money | Унарные операции:  ++ добавление копейки к объекту типа Money (учесть, что копеек не может быть больше 99).  -- вычитание копейки из объекта типа Money (учесть, что копеек и рублей не может быть меньше 0).  Операции приведения типа:  int (неявная) – результатом является количество рублей (копейки отбрасываются);  double (явная) – результатом является копейки, рубли отбрасываются, результат <1.  Бинарные операции:  - Money m, целое число (лево- и право- сторонние операции), результат должен быть типа Money.  - Money m, Money m, результат должен быть типа Money.  Результат не может быть отрицательным. |
| 9 | Point | Унарные операции:  -- уменьшить координаты x и y на 1;  - поменять координаты х и у местами.  Операции приведения типа:  int (неявная) – результатом является целая часть координаты х;  double (явная) – результатом является координата y.  Бинарные операции:  – Point p, целое число (левосторонняя операция, уменьшается координата х);  – целое число, Point p (правосторонняя операция, уменьшается координата y);  – Point p – вычисляется расстояние до точки p, результатом должно быть вещественное число. |
| 10 | Triangle | Унарные операции:  – вычисляется площадь треугольника, результатом должно быть вещественное число.  Операции приведения типа:  double (неявная) – результатом является периметр треугольника ;  bool (явная) – результатом является true, если треугольник с такими длинами сторон существует и false в противном случае.  Бинарные операции:  < - сравниваются площади треугольников;  > - сравниваются площади треугольников. |
| 11 | Diapason | Унарные операции:  ! вычислить длину диапазона, результатом должно быть вещественное число.  Операции приведения типа:  int (неявная) – результатом является целая часть координаты х;  double (явная) – результатом является координата y.  Бинарные операции:  - Diapason целое число (левосторонняя операция, уменьшается координата х);  - целое число Diapason (правосторонняя операция, уменьшается координата y);  < Diapason d – результат равен true, если левый и правый диапазоны пересекаются и false – в противном случае. |
| 12 | Time | Унарные операции:  - обнуление часов и минут в объекте типа Time.  -- вычитание минут из объекта типа Time (учесть, что минут и часов не может быть меньше 0).  Операции приведения типа:  int (неявная) – результатом является количество часов (минуты отбрасываются);  bool (явная) – результатом является true, если часы и минуты не равны нулю и false в противном случае.  Бинарные операции:  == Time t1, Time t2 результатом является true, если значение часов и минут левого операнда равно соответствующим значениям правого операнда, и false – в противном случае.  != Time t1, Time t2 результатом является true, если значение часов и минут левого операнда не равно соответствующим значениям правого операнда, и false – в противном случае. |
| 13 | Money | Унарные операции:  -- вычитание копейки из объекта типа Money (учесть, что рублей и копеек не может быть меньше 0).  ++ добавление копейки к объекту типа Money (учесть, что копеек не может быть больше 99).  Операции приведения типа:  int (явная) результатом является количество рублей (копейки отбрасываются);  bool (неявная) результатом является true, если денежная сумма не равна 0.  Бинарные операции:  - Money m, целое число (лево- и право- сторонние операции). Увеличиваются копейки, необходимо учесть, что копеек не может быть больше 99.  - Money m1, Money m2 вычитание денежных сумм, учесть, что результат не может быть меньше 0. |
| 14 | Time | Унарные операции:  -- вычитание минуты из объекта типа Time (учесть, что часов и минут не может быть меньше 0).  Операции приведения типа:  int (явная) – результатом является количество часов (минуты отбрасываются);  bool (неявная) – результатом является true, если часы и не равны нулю и false в противном случае.  Бинарные операции:  + Time t, целое число (лево- и право- сторонние операции) – добавление минут к временному интервалу.  + Time t1, Time t2 – сложение временных интервалов. |
| 15 | Money | Унарные операции:  -- вычитание копейки из объекта типа Money (учесть, что рублей и копеек не может быть меньше 0).  ++ добавление копейки к объекту типа Money (учесть, что копеек не может быть больше 99).  Операции приведения типа:  int (неявная) результатом является количество рублей (копейки отбрасываются);  double (явная) результатом является копейки, рубли отбрасываются, результат <1.  Бинарные операции:  - Money m, целое число (лево- и право- сторонние операции) вычитание копеек из денежной суммы, учесть, что результат не может быть меньше 0. |

## Часть 3

1. Реализовать класс-коллекцию (в отдельном файле), полем которого является одномерный массив (**не использовать стандартные коллекции C#!)** из элементов заданного в варианте типа. Например, для класса Fraction нужно создать класс FractionArray следующим образом:

class FractionArray

{

Fraction[] arr;//одномерный массив элементов типа Fraction

. . . .

}

В классе реализовать

* конструктор без параметров,
* конструктор с параметрами, заполняющий элементы случайными значениями,
* конструктор с параметрами, позволяющий заполнить массив элементами, заданными пользователем с клавиатуры,
* метод для просмотра элементов массива.

1. Реализовать индексатор для доступа к элементам коллекции. Предусмотреть проверку при выходе индекса за пределы массива.
2. Написать демонстрационную программу, позволяющую создать массив разными способами и распечатать элементы массива. Подсчитать количество созданных объектов.
3. Написать функцию **в классе Program** для выполнения указанного в варианте задания (**использовать индексатор** и, если необходимо, перегрузить нужные для выполнения задачи операции).

**Варианты**

|  |  |  |
| --- | --- | --- |
| № | Название класса |  |
| 1 | Time | Среднее арифметическое. |
| 2 | Money | Минимальное значение. |
| 3 | Point | Самую удаленную от центра координат точку. |
| 4 | Triangle | Номер элемента с минимальной площадью. |
| 5 | Diapason | Максимальное значение. |
| 6 | Uravnenie | Уравнение с самым большим по абсолютному значению корнем. |
| 7 | Time | Максимальное значение. |
| 8 | Money | Максимальное значение |
| 9 | Point | Самую приближенную к центру координат точку |
| 10 | Triangle | Треугольник с минимальной площадью |
| 11 | Diapason | Максимальное значение |
| 12 | Time | Минимальное значение |
| 13 | Money | Среднее арифметическое |
| 14 | Time | Номер максимального элемента |
| 15 | Money | Номер минимального элемента |

**Методические указания:**

1. Операции ввода данных и вывода результата (пользовательский интерфейс) реализовать в отдельном классе.
2. При реализации конструктора использовать свойства для инициализации полей класса.
3. Для подсчета количества объектов использовать статическую переменную.
4. В индексаторе использовать исключения (throw new ArgumentException()).

Содержание отчета:

1. Постановка задачи (общая и конкретного варианта)
2. Диаграмма классов для каждой части работы (из VS или Visio)
3. Код программы
4. Код unit-тестов (или тесты в Excel)
5. Анализ покрытия кода тестами (из VS или в Excel)