# Лабораторная работа №6

## Создание класса

### Вариант 1

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

### Задание к работе

1. Изучить главы 8, 9, 10 учебника «Visual C# 2010: полный курс».
2. По заданному описанию класса в соответствии с вариантом создать класс на языке C#.
3. Создать класс Point, разработав следующие элементы класса:
   1. Поля:
      * int x, y.
   2. Конструкторы, позволяющие создать экземпляр класса:
      * с нулевыми координатами;
      * с заданными координатами.
   3. Методы, позволяющие:
      * вывести координаты точки на экран;
      * рассчитать расстояние от начала координат до точки;
      * переместить точку на плоскости на вектор (a, b).
   4. Свойства позволяющие:
      * получить-установить координаты точки (доступное для чтения и записи);
      * умножить координаты точки на скаляр (доступное только для записи).

# Решение

## Описание класса

Для хранения точки в классе Point используется структура coordinates типа Coords, которая состоит из двух полей: x и y.

Для инициализации экземпляра класса используются два конструктора: первый, когда нужно инициализировать точку нулевыми координатами, второй — заданными.

### Краткое описание методов и свойств класса

1. Print() — метод для вывода координат точки.
2. Vector() — метод, возвращающий расстояние от начала координат до точки.
3. Transition(int a, int b) — метод, перемещающий точку на плоскости на вектор (a, b).
4. Coordinates — свойство, устанавливающее или отдающее координат точки.
5. Multiplication — свойство, доступное только для записи, умножающее координаты на скаляр.

### Результат работы программы

|  |
| --- |
| Сообщение на экране |
| Координаты после инициализации: (0; 0) Координаты после инициализации значениями 5 и 10: (5; 10) Расстояние от 0,0 до заданной точки: 11.18 Координаты точки после перемещения на вектор (3, 7): (8; 17) Координаты точки после задания x = 1, y = 2: (1; 2) И после их умножения на 10: (10; 20) |

### Текст программы

|  |  |
| --- | --- |
| 001 | **using** System; |
| 002 |  |
| 003 | **namespace** OOP\_06 |
| 004 | { |
| 005 | **class** Program |
| 006 | { |
| 007 | **static void** Main(**string**[] args) |
| 008 | { |
| 009 | Point point = **new** Point(); |
| 010 | Console.Write(*"Координаты после инициализации: "*); |
| 011 | point.Print(); |
| 012 |  |
| 013 | point = **new** Point(5, 10); |
| 014 | Console.Write(*"Координаты после инициализации значениями  5 и 10: "*); |
| 015 | point.Print(); |
| 016 |  |
| 017 | Console.WriteLine(*"Расстояние от 0,0 до заданной точки:   {0:N2}"*, point.Vector()); |
| 018 |  |
| 019 | Console.Write(*"Координаты точки после перемещения на вектор  (3, 7):"*); |
| 020 | point.Transition(3, 7); |
| 021 | point.Print(); |
| 022 |  |
| 023 | Console.Write(*"Координаты точки после задания x = 1,  y = 2: "*); |
| 024 | Point.Coords tmp; |
| 025 | tmp.x = 1; |
| 026 | tmp.y = 2; |
| 027 | point.Coordinates = tmp; |
| 028 | point.Print(); |
| 029 |  |
| 030 | Console.Write(*"И после их умножения на 10: "*); |
| 031 | point.Multiplication = 10; |
| 032 | point.Print(); |
| 033 |  |
| 034 | Console.ReadKey(); |
| 035 | } |
| 036 | } |
| 037 |  |
| 038 | **public class** Point |
| 039 | { |
| 040 | /// <summary> |
| 041 | /// Структура для хранения координат точки |
| 042 | /// </summary> |
| 043 | **public struct** Coords |
| 044 | { |
| 045 | **public int** x; |
| 046 | **public int** y; |
| 047 | } |
| 048 |  |
| 049 | /// <summary> |
| 050 | /// Поле для выполнения операций с элементом класса — точкой |
| 051 | /// </summary> |
| 052 | **private** Coords coordinates; |
| 053 |  |
| 054 | /// <summary> |
| 055 | /// Конструктор класса, задающий нулевые значения |
| 056 | /// </summary> |
| 057 | **public** Point() |
| 058 | { |
| 059 | coordinates.x = 0; |
| 060 | coordinates.y = 0; |
| 061 | } |
| 062 |  |
| 063 | /// <summary> |
| 064 | /// Конструктор класса, с заданными координатами |
| 065 | /// </summary> |
| 066 | /// <param name="a">x</param> |
| 067 | /// <param name="b">y</param> |
| 068 | **public** Point(**int** a, **int** b) |
| 069 | { |
| 070 | coordinates.x = a; |
| 071 | coordinates.y = b; |
| 072 | } |
| 073 |  |
| 074 | /// <summary> |
| 075 | /// Выводит координаты точки на экран |
| 076 | /// </summary> |
| 077 | **public void** Print() |
| 078 | { |
| 079 | Console.WriteLine(*"({0}; {1})"*, coordinates.x, coordinates.y); |
| 080 | } |
| 081 |  |
| 082 | /// <summary> |
| 083 | /// Находит расстояние от начала координат до точки |
| 084 | /// </summary> |
| 085 | /// <returns>Длину вектора от 0,0 до x,y</returns> |
| 086 | **public** **double** Vector() |
| 087 | { |
| 088 | **return** Math.Sqrt(coordinates.x \* coordinates.x +   coordinates.y \* coordinates.y); |
| 089 | } |
| 090 |  |
| 091 | /// <summary> |
| 092 | /// Перемещает точку на плоскости на вектор (a, b) |
| 093 | /// </summary> |
| 094 | /// <param name="a">a</param> |
| 095 | /// <param name="b">b</param> |
| 096 | **public** **void** Transition(**int** a, **int** b) |
| 097 | { |
| 098 | coordinates.x += a; |
| 099 | coordinates.y += b; |
| 100 | } |
| 101 |  |
| 102 | /// <summary> |
| 103 | /// Позволяет задать или получить координаты |
| 104 | /// </summary> |
| 105 | **public** Coords Coordinates |
| 106 | { |
| 107 | **get** { **return** coordinates; } |
| 108 | **set** { coordinates = **value**; } |
| 109 | } |
| 110 |  |
| 111 | /// <summary> |
| 112 | /// Позволяет умножить координаты на скаляр (только для записи) |
| 113 | /// </summary> |
| 114 | **public** **int** Multiplication |
| 115 | { |
| 116 | **set** |
| 117 | { |
| 118 | coordinates.x \*= **value**; |
| 119 | coordinates.y \*= **value**; |
| 120 | } |
| 121 | } |
| 122 | } |
| 123 | } |