

6. Наследование и полиморфизм

6.1. Наследование и интерфейсы

6.1-0 Описать интерфейс `ICalculation` для преобразований чисел типа `double`. Он должен содержать метод `Perform` (выполнить преобразование), принимающий число и возвращающий результат преобразования.

Описать два класса, реализующих этот интерфейс: `Add` и `Multiply`. Первый класс увеличивает число на некоторую величину, а второй умножает на заданный коэффициент (величины определяются конструкторами).

Написать функцию `Calculate`, принимающую число и два преобразования. Функция должна последовательно применять преобразования к числу и возвращать результат.

Написать программу, использующую эту функцию.

Пример вызова функции:

```
var x = Calculate(1, new Add(2), new Multiply(3));
```

Построить UML-диаграмму.

6.1-1 Описать интерфейс `IShape` для геометрических фигур. Интерфейс должен содержать методы: `Perimeter` и `Area`, возвращающие периметр и площадь соответственно.

Описать классы `Triangle` (треугольник) и `Disk` (круг), реализующие этот интерфейс. Параметры фигур должны задаваться при создании экземпляра.

Написать функцию, принимающую фигуру и выводящую на экран её название, параметры, периметр и площадь.

Написать программу, использующую эту функцию.

Построить UML-диаграмму.

6.1-2 Описать абстрактный класс `Person` (человек), соответствующий человеку. Экземпляры этого класса должны хранить информацию о фамилии (`Surname`), имени (`Name`) и отчестве (`Patronymic`). Класс должен содержать абстрактный метод `Income` без параметров, возвращающий годовой доход.

Описать классы `Student` (студент) и `Employee` (сотрудник), являющиеся потомками класса `Person`. Класс `Student` содержит величину ежемесячной стипендии, а класс `Employee` — величину месячного оклада и процентную ставку налога.

Написать программу, использующую эти классы.

Построить UML-диаграмму.

6.1-3 Описать интерфейс `ISolid` для геометрических тел. Интерфейс должен содержать методы: `Volume` и `SurfaceArea`, возвращающие объём и площадь поверхности соответственно.

Описать классы `Cube` (куб) и `Cylinder` (цилиндр), реализующие этот интерфейс. Параметры тел должны задаваться при создании экземпляра.

Написать функцию, принимающую тело и выводящую на экран её название, параметры, объём и площадь поверхности.

Написать программу, использующую эту функцию.

Построить UML-диаграмму.

6.1-4 Описать интерфейс `IPrinter`, соответствующий способам вывода действительных чисел на экран. Он должен содержать метод `Print`, принимающий число и выводящий его на экран в соответствии с конкретной реализацией.

Описать классы `PrecisionPrinter` и `LinePrinter`, реализующие этот интерфейс. Первый класс выводит число с указанной точностью, а второй после каждого числа рисует линию указанной длины из знаков «-». Необходимые параметры (точность и длина линии) передаются через конструктор.

Написать функцию `PrintNumbers`, выводящую числа начиная с 0 с указанным шагом при помощи класса, передаваемого через аргументы. Также через аргументы должно передаваться количество выводимых чисел и шаг.

Написать программу, использующую эту функцию.

Пример вызова функции:

```
PrintNumbers(5, 0.2, new LinePrinter(10));
```

Построить UML-диаграмму.

6.1-5 Описать интерфейс `IPolynomial`, соответствующий многочленам. Он должен содержать методы `IsAbove` и `IsBelow`, проверяющие, находится ли указанная точка на плоскости строго выше или ниже графика многочлена соответственно.

Описать классы `Linear` (линейный) и `Quadratic` (квадратный), реализующие этот интерфейс. Написать функцию `IsBetween`, принимающую два многочлена и координаты точки и проверяющую, находится ли точка выше первого и ниже второго многочлена.

Написать программу, использующую эту функцию.

Построить UML-диаграмму.

6.1-6 Описать интерфейс `IVector`, соответствующий векторам. Он должен содержать методы:

- `Size` (размер), возвращающий количество координат вектора;
- `Get` (получить), возвращающий координату вектора по номеру.

Описать классы `Vector2` (двумерный вектор), `Vector3` (трёхмерный вектор), реализующие этот интерфейс. Написать функцию, находящую скалярное произведение двух векторов, если это возможно. (Произведение определено только для векторов одного размера.)

Написать программу, использующую эту функцию.

Построить UML-диаграмму.

6.1-7 Реализовать иерархию классов: Point (точка) — MassivePoint (точка с массой) — MassiveBall (шар с массой). Точка определяется координатами в пространстве, шар имеет дополнительную характеристику — радиус. Все параметры должны задаваться при создании объектов.

Написать функции:

- Distance (расстояние), определяющую расстояние между произвольными объектами;
- Attraction (притяжение), находящую силу взаимного притяжения между объектами, имеющими массу.

При вычислениях пренебречь возможными взаимными пересечениями объектов.

Написать программу, использующую эти функции.

Построить UML-диаграмму.

6.1-8 Описать интерфейс ISequence, соответствующий числовым последовательностям. Он должен содержать метод GetElement, возвращающий элемент последовательности по его номеру.

Описать классы ArithmeticProgression (арифметическая прогрессия) и GeometricProgression (геометрическая прогрессия), реализующие этот интерфейс. Параметры прогрессий (первый элемент и разность или знаменатель) должны задаваться при создании экземпляра.

Написать функцию Sum которая возвращают сумму указанного количества элементов последовательности начиная с первого.

Написать программу, использующую эту функцию.

Пример вызова функции:

```
var s = Sum(new ArithmeticProgression(3, 5), 10);
```

Построить UML-диаграмму.

6.1-9 Описать абстрактный класс Viewer, соответствующий зрителям кино-театра. Он должен содержать:

- поле visits, хранящее количество посещений;
- метод Visit (посетить), увеличивающий число посещений на 1;
- абстрактный метод Cost (стоимость), получающий цену билета и возвращающий его стоимость с учётом скидок.

Изначальное количество посещений задаётся при помощи конструктора.

Описать классы RegularViewer (постоянный посетитель) и StudentViewer (студент), являющиеся потомками класса Viewer. Постоянный посетитель за каждые

10 посещений получает дополнительную скидку в 1 %, но не более, чем 20 %. Для студентов на каждое третье посещение даётся скидка в 50 %.

Написать функцию `TotalCost`, принимающую экземпляр класса `Viewer`, цену билета и количество сеансов, и возвращающую общую стоимость билетов с учётом скидок. Эта функция должна моделировать посещения кинотеатра.

Написать программу, использующую эту функцию.

Пример вызова функции:

```
var viewer = new StudentViewer(5);  
var total = new TotalCost(viewer, 35.00M, 20);
```

Этот вызов вычисляет стоимость двадцати посещений сеансов, стоящих 35 рублей, для студента, который уже посетил пять сеансов. После вызова количество посещений у объекта `viewer` будет равно 25.

Построить UML-диаграмму.

6.2. Сравнение экземпляров

6.2-0 Описать класс `Triangle`, соответствующий треугольникам. Определить в нем операции и методы сравнения, сравнивающие треугольники по площади.

6.2-1 Описать класс `Time`, соответствующий времени суток (часы и минуты). Определить в нем операции и методы сравнения. Более ранние моменты времени считать меньшими, чем более поздние.

6.2-2 Описать класс `Progression`, соответствующий геометрическим прогрессиям. Определить в нём операции и методы сравнения, сравнивающие прогрессии по значению суммы бесконечного числа их элементов.

6.2-3 Описать класс `Box`, соответствующий параллелепипедам. Определить операции и методы сравнения, позволяющие проверить, можно ли вложить один параллелепипед в другой (в этом случае вложенный считать меньшим, чем объёмлющий). Стенки считать бесконечно тонкими.

6.2-4 Описать класс `RGBColor`, соответствующий цвету в модели RGB. В этой модели каждый цвет задаётся тремя числами R , G и B , соответствующими интенсивностям красной, зелёной и синей компонент соответственно. Интенсивности — действительные числа из отрезка $[0; 1]$.

Определить в классе операции и методы сравнения, сравнивающие цвета по яркости. Яркость вычисляется по формуле:

$$Y = 0,299R + 0,587G + 0,114B.$$

6.2-5 Описать класс `Deposit`, соответствующий банковскому вкладу. Вклад определятся тремя величинами: начальной суммой, сроком (в годах) и процентной ставкой. Каждый год сумма на счету увеличивается на величину процентной ставки.

Определить в классе операции и методы сравнения, сравнивающие вклады по величине чистой прибыли за срок действия вклада.

6.2-6 Описать класс `Interval`, соответствующий отрезкам числовой прямой. Определить в классе операции и методы сравнения, сравнивающие отрезки следующим образом. Из двух пересекающихся отрезков больше тот, у которого доля общей части больше. Непересекающиеся отрезки считаются разными.

6.2-7 Описать класс `Rate`, соответствующий пакетам услуг связи. Каждый пакет услуг определяется стоимостью, количеством минут в пакете и количеством бесплатных минут.

Определить в классе операции и методы сравнения, сравнивающие пакеты по экономичности. Более экономичным является тот пакет услуг, в котором стоимость одной минуты ниже.

6.2-8 Описать класс `Point`, соответствующий точкам в пространстве. Определить в классе операции и методы сравнения. Считать меньшей ту точку, которая ближе к началу координат.

6.2-9 Описать класс `Line` отрезков на плоскости. Определить в нем операции и методы сравнения, сравнивающие отрезки по длине.