

# Контрольная работа. Модуль 2

## Вариант 1

1. Описать класс Point со следующими полями и методами:
  - a. final double x – x координата точки (final означает «константа»)
  - b. final double y – y координата точки
  - c. Point(double x, double y) – конструктор
  - d. double distanceTo(Point other) – евклидово расстояние от текущей точки до другой, вычисляется по формуле  $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
2. Описать абстрактный класс Shape, который представляет собой абстрактную фигуру и определяет следующий метод:
  - a. abstract boolean contains(Point point) – проверяет, лежит ли точка внутри фигуры
3. Описать класс Circle как наследника класса Shape, который описывает круг, задаваемый точкой центра и радиусом:
  - a. Circle(Point center, double radius) – конструктор.
4. Описать класс Composite, как наследник класса Shape, который описывает фигуру, составленную объединением других фигур. Такая фигура будет содержать список фигур, из которых она состоит (List<Shape>). Для того, чтобы узнать лежит ли точка внутри составной фигуры, нужно проверить, лежит ли эта точка хотя бы в одной из фигур, из которых состоит составная. Описать метод:
  - a. void add(Shape shape) – добавляет фигуру к текущей составной фигуре.
5. Описать класс Program, который будет содержать метод main, в котором будет построена фигура следующего вида:
  - a. Составная фигура из:
    - i. Круга (0,0) ~ 10 (центр — точка 0,0; радиус — 10)
    - ii. Круга (5,3) ~ 5
    - iii. Составной фигуры из:
      1. Круга (10,2) ~ 2
      2. Круга (100,30) ~ 15
6. Пользователю должно быть предложено ввести координаты точки, после чего ему должно быть выведено сообщение лежит ли эта точка внутри данной фигуры или нет.

## Вариант 2

1. Опишите абстрактный класс `Trader`, который будет описывать продавцов и обладать следующими методами:
  - a. `abstract double profit();` – вычисляет прибыль данного продавца
  - b. `abstract void printInfo( int indent );` – выводит на экран информацию о продавце, начиная каждую строку вывода с отступа в заданное количество пробелов.
2. Опишите класс `Dealer` как наследника класса `Trader`, у которого есть определенный стабильный доход, методы:
  - a. `Dealer(double profit)` – конструктор
  - b. `printInfo( int indent )` – выводит на экран строку вида "`<indent>Dealer: $XX.XX per month`"
3. Опишите класс `Manager` как наследник класса `Trader`, который имеет список подчиненных продавцов (`List<Trader>`) и фиксированный процент, который он имеет с прибыли своих подчиненных. Реализуйте методы:
  - a. `Manager(double percent)` – конструктор
  - b. `void add( Trader trader )` – добавляет подчиненного продавца менеджеру
  - c. `profit()` – возвращает процент от суммы прибылей подчиненных продавцов
  - d. `printInfo( int indent )` – выводит на экран строку вида "`<indent>Manager: XX% from:`", потом выводит информацию о всех подчиненных, с отступом, увеличенным на 4 пробела, каждый в отдельной строке.
4. Опишите класс `Market`, в котором реализуйте метод `main`, который будет формировать структуру продавцов, приведенную ниже, выводить на экран информацию о продавцах первого уровня (которые не являются чьими-то подчиненными) с 0 отступом, а так же их доход. Продавцы верхнего уровня должны храниться в списке. Структура продавцов:
  - a. Dealer: \$20 per month
  - b. Dealer: \$50 per month
  - c. Manager: 20% from:
    - i. Dealer: \$100 per month
    - ii. Manager: 10% from:
      1. Dealer: \$30
      2. Dealer: \$50
      3. Dealer: \$80
    - iii. Dealer: \$85
  - d. Manager: 10% from:
    - i. Dealer: \$100
    - ii. Dealer: \$30