**Task-6-5**

**Вычисление площадей фигур.**

**Цели задания:**

1. Изучить паттерн «Strategy» («Стратегия»)
2. Изучить паттерн «Factory» («Фабрика»)
3. Использовать абстрактный класс

**Описание:**

Ах, Стратегия и фабрика… После Singleton-а – это, наверное, самые популярные паттрены, а, следовательно, и слова, которые вы будете использовать в своей работе.

Но в отличие от Singleton-а – это еще и слова, которые вы можете произносить в приличном обществе, *и вас не посчитают извращенцем, с которым опасно вступать в романтические отношения, предполагающие наличие сексуальной компоненты*… Хотя я лично, ничего не имею против Singleton-ов, но это предмет отдельной дискуссии, и отдельного задания. Итак, Стратегия и Фабрика…

В этой задаче вам предлагается переписать **задачу-2-2 «Вычисление площадей фигур»** с использованием правильных ООП-техник, а не при помощи этого мерзкого switch-а, которым вы пользовались ранее:



Если внимательно посмотреть на картинку выше, то мы заметим, что каждая фигура отличается двумя моментами:

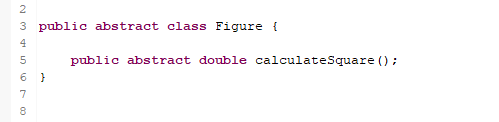
1. Способом ввода, т. е. заданием параметров для расчета.
2. И формулой вычисления площади.

А это значит, что мы можем задать общее поведение и реализовать это поведение для каждой фигуры по-особенному. Это и будет – паттерн «Стратегия». Давайте начнем с вычисления площадей.

Любую стратегию можно реализовать двумя способами

* Либо с использованием интерфейсов (предпочтительный вариант)
* Либо с использованием абстрактных классов

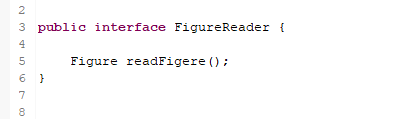
Для стратегии вычисления площади мы выберем абстрактный класс. Создайте базовый абстрактный класс Figure c одним абстрактным методом. Вот так:



Далее, реализуйте столько наследников этого класса, сколько конкретных фигур у вас есть. Готово – стратегия для расчета площадей реализована!

**Внимание!** Класс Figure не отвечает за ввод данных для каждой фигуры! Он отвечает только за расчет площади.

А чтобы реализовать ввод данных для каждой фигуры, мы реализуем вторую стратегию – FigureReader. В данном случае, уже на основе интерфейсов. Создайте вот такой интерфейс:



Видите, интерфейс говорит, что я могу прочитать параметры для фигуры, но он не говорит:

1. Для какой именно фигуры?
2. И как именно прочитать?

С первым пунктом все просто – нужно реализовать классы типа **CircleReader**, **TriangleReader** и т. д… Но что означает вот это:

Как именно прочитать фигуру?

Дело в том, что ваша программа может быть настолько универсальной, что она сможет читать параметры для разных фигур из разных источников данных, например – консоли, файла, удаленного сервера. Таким образом нам недостаточно просто классов **CircleReader** и **TriangleReader**. Нам потребуется вот такая иерархия:

* ConsoleReader
  + CircleConsoleReader
  + TriangleConsoleReader
* FileReader
  + CircleFileReader
  + TriangleFileReader
* RemoteServerReader
  + RemoteServerConsoleReader
  + RemoteServerConsoleReader

Нам потребуется «Стратегия Стратегий», если угодно ☺

И вот такая «Стратегия Стратегий» называется фабрика!