Глава 3
КЛАССЫ

Класс представляет описание совокупности объектов с общими атрибутами, методами, отношениями и семантикой.

Классы – основной элемент абстракции языка Java, основное назначение которого, кроме реализации назначенного ему контракта, это сокрытие реализации. Классы всегда взаимодействуют друг с другом и объединяются в пакеты. Из пакетов создаются модули, которые взаимодействуют друг с другом только через ограниченное количество методов и классов, не имея никакого представления о процессах, происходящих внутри других модулей.

Имя класса в пакете должно быть уникальным. Физически пакет представляет собой каталог, в который помещаются программные файлы, содержащие реализацию классов.

Классы позволяют провести декомпозицию поведения сложной системы до множества элементарных взаимодействий связанных объектов. Класс определяет структуру и/или поведение некоторого элемента предметной области, для которой разрабатывается программная модель.

Определение простейшего класса без наследования имеет вид:

```
class ИмяКласса {
    {}//логические блоки
    // дружественные данные и методы
    private// закрытые данные и методы
    protected// защищенные данные и методы
    public// открытые данные и методы
}
```

Переменные класса и константы

Классы инкапсулируют переменные и методы – члены класса. Переменные класса объявляются в нем следующим образом:

```
спецификатор тип имя;
```

В языке Java могут использоваться статические переменные класса, объявленные один раз для всего класса со спецификатором **static** и одинаковые для всех экземпляров (объектов) класса, или переменные экземпляра класса, создаваемые для каждого объекта класса. Поля класса объявляются со спецификаторами доступа **public**, **private**, **protected** или по умолчанию без спецификатора. Кроме данных – членов класса, в методах класса используются локальные переменные и параметры методов. В отличие от переменных класса, инкапсулируемых нулевыми элементами, переменные методов не инициализируются по умолчанию.

Переменные со спецификатором **final** являются константами. Спецификатор **final** можно использовать для переменной, объявленной в методе, а также для параметра метода.

В следующем примере рассматриваются объявление и инициализация значений полей класса и локальных переменных метода, а также использование параметров метода:

```
/* пример # 1 : типы атрибутов и переменных: Second.java */
package chapt03;
import java.util.*;

class Second {
     private int x; // переменная экземпляра класса
     private int y = 71; // переменная экземпляра класса
     public final int CURRENT_YEAR = 2007; // константа
     protected static int bonus; // переменная класса
     static String version = "Java SE 6"; // переменная класса
     protected Calendar now;

     public int method(int z) { // параметр метода
          z++;
          int a; // локальная переменная метода
          //a++; // ошибка компиляции, значение не задано
          a = 4; //инициализация
          a++;
          now = Calendar.getInstance();//инициализация
          return a + x + y + z;
     }
}
```

В рассмотренном примере в качестве переменных экземпляра класса, переменных класса и локальных переменных метода использованы данные базовых типов, не являющиеся ссылками на объекты (кроме **String**). Данные могут быть ссылками, назначить которым реальные объекты можно с помощью оператора **new**.

Ограничение доступа

Язык Java предоставляет несколько уровней защиты, обеспечивающих возможность настройки области видимости данных и методов. Из-за наличия пакетов Java работает с четырьмя категориями видимости между элементами классов:

- по умолчанию – дружественные члены класса доступны классам, находящимся в том же пакете;
- **private** – члены класса доступны только членам данного класса;
- **protected** – члены класса доступны классам, находящимся в том же пакете, и подклассам – в других пакетах;
- **public** – члены класса доступны для всех классов в этом и других пакетах.