J02 Introductory Java 2

Objects
Classes
Inheritance

ANU – School of Computing – Structured Programming 1110 / 1140 / 6710

Objects

Java is an *object-oriented* language.

Objects combine state and behaviour

- **State**: fields (data)
  - data can be constant (unchanging through the existence of the object), or changing/variable
- **Behaviour**: methods (code)

Classes

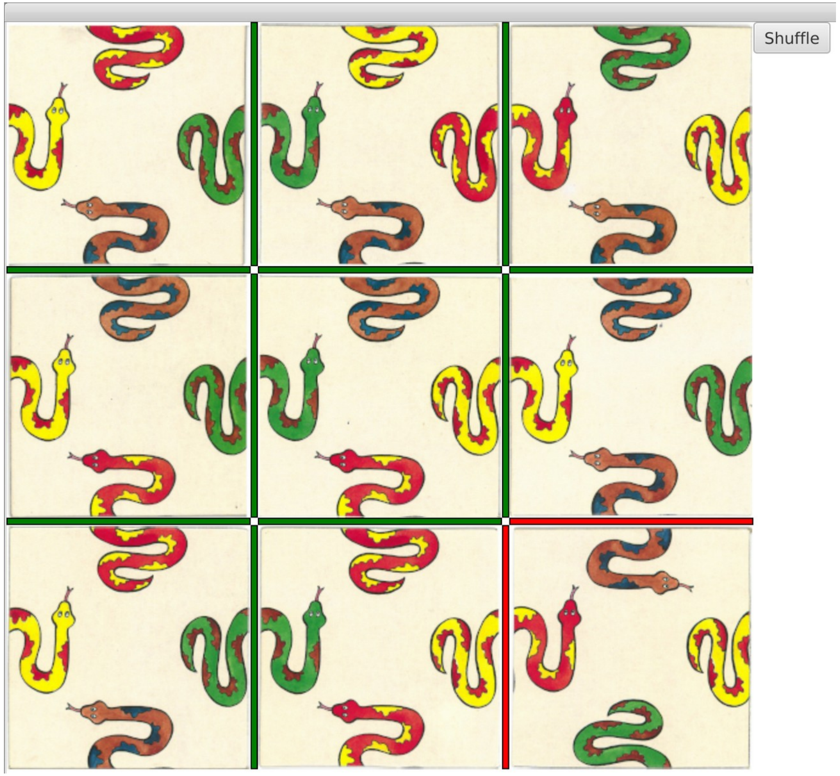
A class is a blueprint or ‘type’ for an object

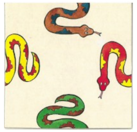
- **Class**: definition used for multiple instances (objects)
  - The class defines what fields and methods each instance of it has; methods are shared by all instances.
  - A class can also have fields, that are shared by all instances.
- **Instance (object)**: one instantiation of a class
  - Each instance has its own concrete values for (non-class) fields.

Methods

- A subroutine/function/procedure
  - Reusable code to perform a specific task
  - Abstraction: modularity, encapsulation
- In Java, almost all code is a method (`main`, if not another).
- Methods may take arguments (parameters).
- Methods may return a value.

The puzzle consists of nine **tile**s, arranged in a 3x3 **grid**. Each tile has one **pattern**, a snake head or tail of a given colour, on each side. The goal of the game is to place and rotate the tiles so that the patterns match at every edge.



- class Tile:
  - Fields: one Pattern for each side, current rotation
  - Methods: rotate
  - Instances:  ...
- class Grid:
  - Fields: tile in each grid position
  - Methods: move (swap) tiles, check if edges match

Inheritance

Classes form a hierarchy

- a sub-class `extends` a super-class
- a child-class `extends` a parent-class

Class `java.lang.Object` is the root (ultimate ancestor) class of all Java Classes