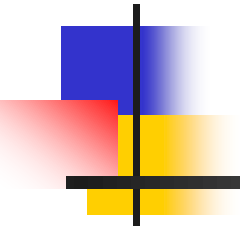
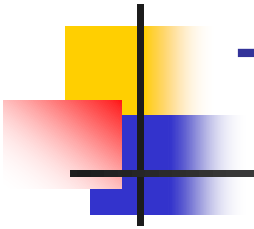
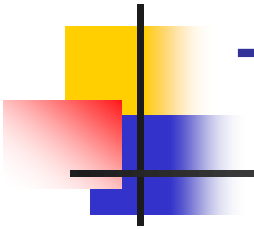
7. Timer & TimerTask





Timer Class

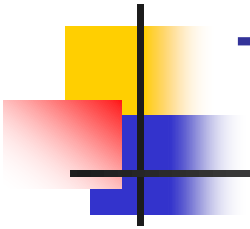
- The java.util.Timer class provides facility for threads to schedule tasks for future execution in a background thread
- This class is thread-safe i.e. multiple threads can share a single Timer object without the need for external synchronization



Timer Class

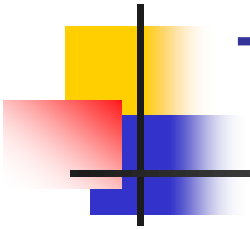
- This class schedules tasks for one-time execution, or for repeated execution at regular intervals.

- All constructors start a timer thread.



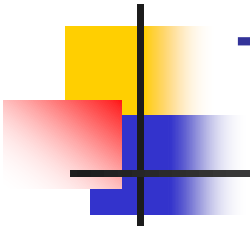
Timer Class

- **public void schedule(TimerTask task, long delay, long period)**
  - Schedules the specified task for repeated fixed-delay execution, beginning after the specified delay (which may be zero)
  - Subsequent executions take place at approximately regular intervals separated by the specified period
  - Times are specified in milliseconds (1/1000s of a second)



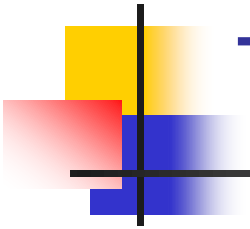
Timer Class

- **public void scheduleAtFixedRate (TimerTask task, long delay, long period)**
  - Schedules the specified task for repeated fixed-delay execution, beginning after the specified delay (which may be zero)
  - Subsequent executions take place at approximately regular intervals separated by the specified period
  - Times are specified in milliseconds (1/1000s of a second)



Timer Class

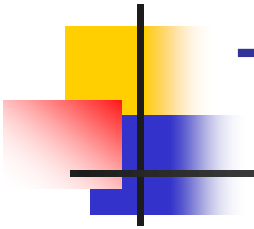
- **public void cancel()**
  - Terminates this timer, discarding any currently scheduled tasks.
  - Does not interfere with a currently executing task (if it exists).
  - Once a timer has been terminated, its execution thread terminates gracefully, and no more tasks may be scheduled on it.



TimerTask class

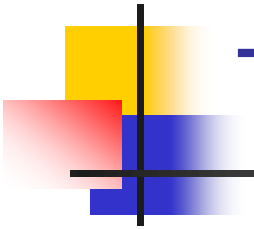
- The java.util.TimerTask class represents a task that can be schdeuled for one-time or repeated execution by a Timer.

- protected TimerTask() - This constructor creates a new timer task.



TimerTask class

- TimerTask is an abstract class you must extend and provide a
  - public void run() method
- TimerTask provides an (implemented) public boolean cancel() method
  - Returns false if there were no scheduled executions to cancel



Timer class

- In fixed rate, it doesn't matter how long the previous execution took, the next execution will happen when it was scheduled. With fixed-delay, the next execution will happen X time after the previous finished, even if it was late.