

Nombre:	
Código:	Fecha: 6 de Abril de 2019

Unidad 1: Construcción de la interfaz gráfica

OE1.1. Utilizar una arquitectura de tres capas para el desarrollo de un programa de computador, repartiendo de manera adecuada las responsabilidades entre la interfaz de usuario, el control de la interfaz y el modelo. El estudiante deberá poder explicar la importancia de mantener separadas las clases de estos tres dominios.

OE1.2. Construir las clases que implementan una interfaz de usuario.

Unidad 3: Concurrencia, Dibujo Básico en 2D y Persistencia

OE3.1 Desarrollar un programa que maneje concurrencia, de manera que sea posible que ejecute más de una parte del programa de manera simultánea, utilizando hilos de ejecución (threads).

OE3.2 Construir interfaces de usuario que incluyan gráficas en 2 dimensiones como una alternativa en la presentación de información al usuario.

OE3.4 Manipular archivos de texto y utilizarlos para implementar requerimientos del cliente relacionados con persistencia.

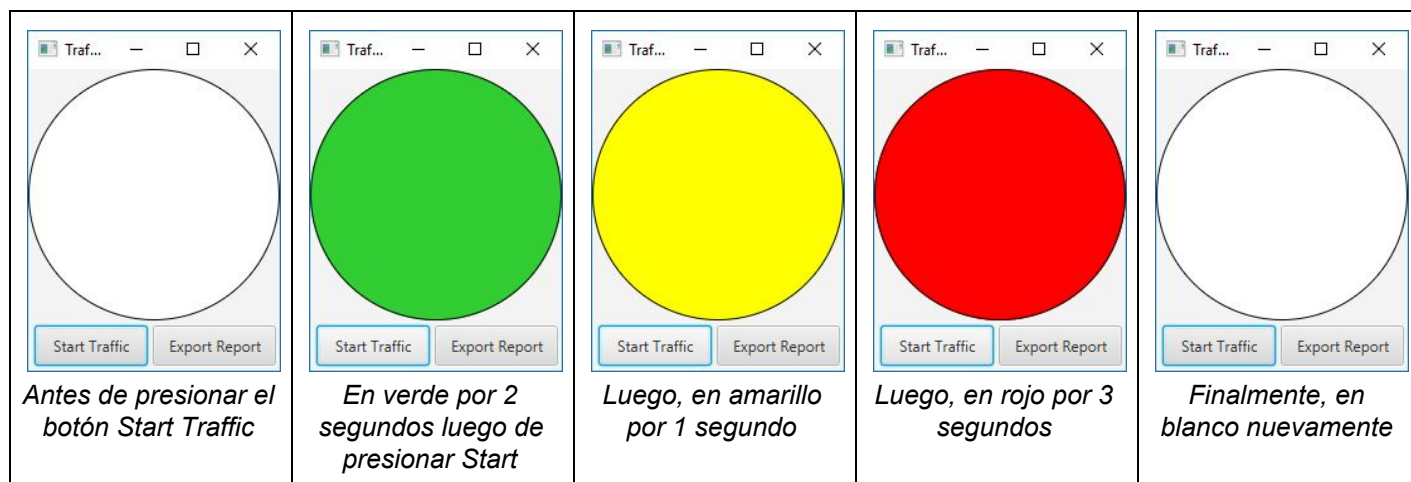
Enunciado

Ante la alta tasa de infracciones de tránsito, el gobierno municipal contrató un estudio socio psicológico que permitiera encontrar las razones de tantas violaciones al código de tránsito por parte de las personas. Uno de los hallazgos del estudio es que a los niños no se les da a conocer conceptos de control de tráfico elementales como señales de tránsito, semáforos, policías de tránsito, etc y luego cuando son adultos, presentan una resistencia a la autoridad y a las normas.

Primera Parte

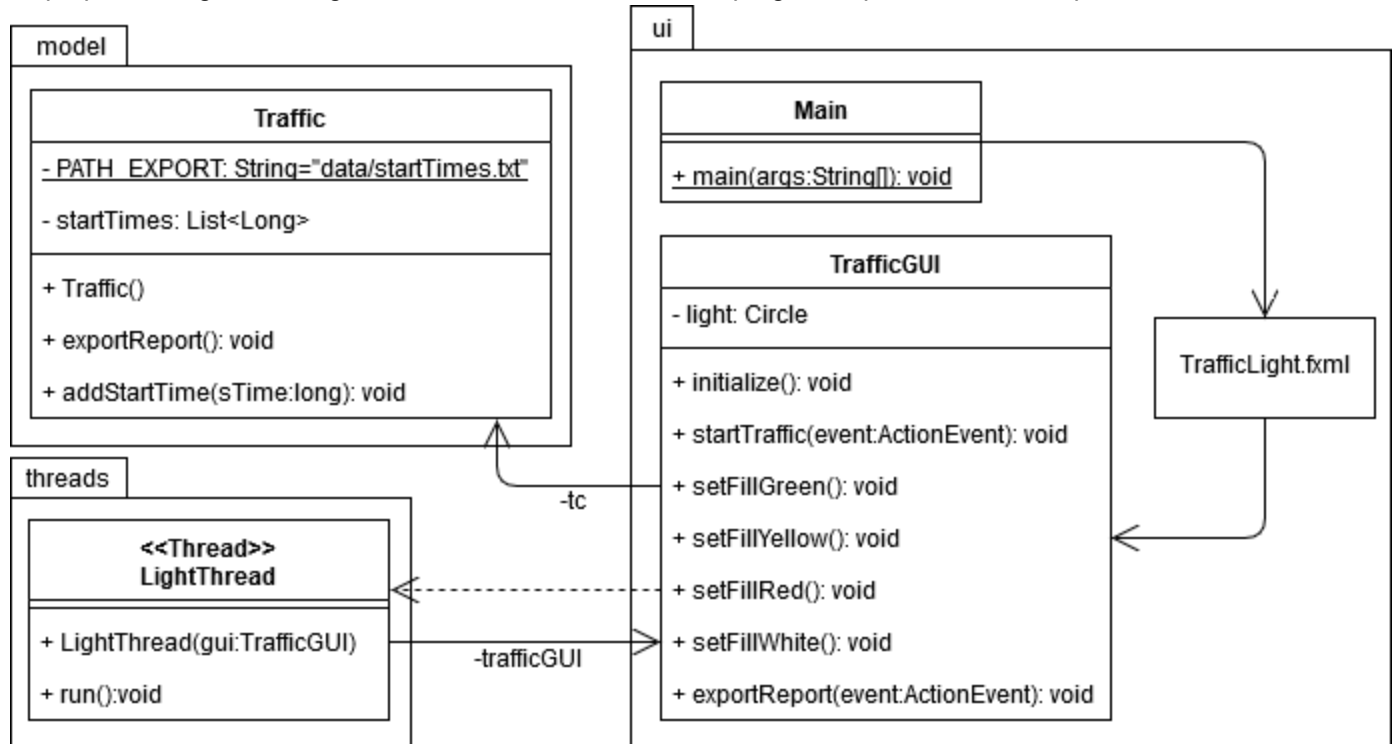
Por lo anterior, usted ha sido contratado para que desarrolle una programa que permita familiarizar a los niños pequeños con el funcionamiento de un semáforo. En esta primera versión, el programa debe solamente tener un círculo de 200x200 pixeles en el centro de la pantalla simulando una luz de semáforo. Esta luz estará inicialmente en blanco, pero luego de que el niño presione el botón *Start Traffic*, la luz cambiará a verde por 2 segundos, luego a amarillo por 1 segundo, luego a rojo por 3 segundos y finalmente a blanco para quedarse de esa manera. Si el botón es presionado nuevamente, la misma acción debe llevarse a cabo. Esto aplica para todas las oportunidades en que el botón sea presionado.

A continuación se muestra los estados del programa en cuestión:



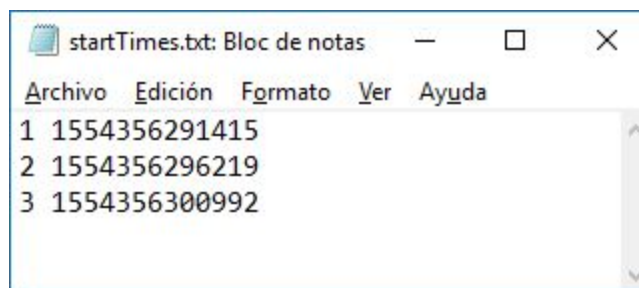
	<i>Traffic</i>			
--	----------------	--	--	--

Se propone el siguiente diagrama de clases como diseño del programa que usted debe implementar.



Segunda Parte

Cada vez que el niño haga clic sobre botón de inicio de tráfico (*Start Traffic*), el programa debe guardar internamente el tiempo actual en milisegundos. En Java, puede utilizar la función `System.currentTimeMillis()` que retorna un `long` con ese valor. Luego, el niño u otro usuario de la aplicación (generalmente el tutor del niño), puede hacer clic en el botón *Export Report* y el programa guardará en un archivo de texto los tiempos en los cuales el niño presionó el botón de inicio de tráfico (*Start Traffic*). El formato de salida del archivo es: una línea por cada tiempo, y cada línea tiene un consecutivo (que inicia en 1) seguido del tiempo. Un ejemplo de salida de este archivo se muestra a continuación:



Funcionalidades

1. [15%] Interfaz gráfica con JavaFX. Ventana, atributos, botones y acciones.
2. [20%] Forma básica en 2D. Círculo, propiedades y actualización de la interfaz.

3. [35%] Hilos. Declaración de la clase, implementación del método run, creación del hilo.
4. [30%] Archivos de texto. Generación de la cadena del reporte e instrucciones para escribirla en un archivo de texto.