



1. Berapa banyak thread yang akan berjalan apabila object Timer dijalankan?

- 1
 2
 3
 4

2. Dari method berikut ini apa maksud dari ttCanvas.repaint(50, 50, 20, 20) ?

```
private void timerRun() {
    count++;
    ttCanvas.update();
    ttCanvas.repaint(50, 50, 20, 20);
}
```

- Menggambar kembali pada class Canvas degan posisi x = 20, y = 20, panjang = 50, dan lebar 50.
 Menggambar kembali pada class Canvas degan posisi x = 50, y = 20, panjang = 50, dan lebar 20.
 Menggambar kembali pada class Canvas degan posisi x = 50, y = 50, panjang = 20, dan lebar 20.
 Menggambar kembali pada class Canvas degan posisi x = 20, y = 50, panjang = 50, dan lebar 20.
 Menggambar kembali pada class Canvas degan posisi x = 50, y = 20, panjang = 20, dan lebar 50.

3. Manakah pernyataan yang benar tentang class Timer?

- Sebuah class yang digunakan untuk menjadwalkan kegiatan akan datang (future task) yang akan dieksekusi sebagai background
 Class ini tidak mendukung real-time proses
 Class ini menjadwalkan task menggunakan method wait() dalam class Object
 VM akan secara otomatis memanggil fungsi timer pada class ini
 E. Fungsi timer dalam class ini dipanggil sesuai dengan request pada aplikasi

4. Apa maksud dari method berikut ini?

```
protected void hideNotify() {
    midletTT.stopTimer();
}
```

- Menjalankan pertama kali method stopTimer pada class midlet.
 Menjalankan pertama kali method stopTimer pada class canvas.
 Menjalankan terakhir method stopTimer pada class midlet.
 Menjalankan terakhir method stopTimer pada class canvas.
 Menghentikan method stopTimer pada akhir program pada class midlet.

5. Method dan parameter yang salah pada class Timer dengan object timer adalah?

- timer.schedule(TimerTask task, long delay);
- timer.schedule(TimerTask task, long delay, long period);
- timer.schedule(TimerTask task, Date firstTime, long period);
- timer.scheduleAtFixedRate(TimerTask task, long delay);
- timer.scheduleAtFixedRate(TimerTask task, long delay, long period);

6. Apakah fungsi dari potongan kode dibawah ini?

```
void stopTimer() {  
    timer.cancel();  
    timerTask.cancel();  
    display.setCurrent(new SuccesCanvas());  
}
```

- Meberhentikan waktu
- Menjalankan waktu.
- Menghapus waktu.
- Menampilkan class succescanvas
- Memberhentikan sementara.

7. Pilih salah satu method yang dimiliki oleh Class TimerTask!

- schedule(TimerTask task, Date firstTime, long period);
- scheduleAtFixedRate(TimerTask task, Date firstTime, long period);
- schedule(TimerTask task, Date time);
- cancel();
- scheduleAtFixedRate(TimerTask task, long delay, long period);

8. Dari pernyataan-pernyataan berikut ini, manakah yang merupakan pernyataan yang benar tentang scheduler dalam TimerTask?

- Setiap scheduler dijalankan dalam waktu tertentu
- Setiap scheduler bisa dijalankan setelah adanya delay tertentu (dalam milliseconds)
- Scheduler bisa dijalankan hanya sekali, atau bisa dijalankan beberapa kali pada interval tertentu
- Scheduler bisa dieksekusi beberapa kali yang secara relatif mengacu pada eksekusi periode yang pertama

9. Mengapa sebuah TimerTask harus didefinisikan diluar MIDlet ?

- Karena TimerTask merupakan sebuah abstract yang tidak bisa didefinisikan/diimplementasikan bersamaan dengan MIDlet
- Karena TimerTask memiliki fungsi yang berbeda dengan MIDlet
- Karena TimerTask merupakan thread yang berbeda dengan MIDlet
- Karena TimerTask merupakan sebuah interface yang tidak dapat diimplementasikan didalam MIDlet

10. Apakah class Timer merupakan sebuah Thread-safe?

- Benar
- Salah

11. Apa yang terjadi jika potongan kode dibawah ini dijalankan?

```
void startTimer() {  
    timer = new Timer();  
    timerTask = new TimerTask() {  
        public void run() {  
            timerRun();  
        }  
    };  
    timer.scheduleAtFixedRate(timerTask, 100, 2000);  
}
```

- menampilkan selang waktu pertama 2000 milisecond dan seanjutnya 100 milisecond.
- menampilkan selang waktu pertama 100 milisecond dan seanjutnya 2000 milisecond. (X)
- menampilkan selang waktu 2000 saja.
- menampilkan selang waktu 100 saja.
- menampilkan delay pertama 2000 x 100 milisecond.

12. Method manakah yang mampu menghitung dengan selang waktu tetapi pada saat awal tampil menggunakan waktu yang lain ?

- schedule(TimerTask task, Date firstTime, long period);
- scheduleAtFixedRate(TimerTask task, Date firstTime, long period);
- schedule(TimerTask task, Date time);
- scheduleAtFixedRate(TimerTask task, long delay, long period);
- schedule(TimerTask task, long delay, long period);

13. Bagaimana untuk menjalankan delay 1 detik (secara konstan) pada class TimerTask dengan object task, dan class Timer dengan object timer?

- timer.schedule(task, 100);
- timer.schedule(task, 100, 1000);
- timer.schedule(task, 1000, 100);

- timer.scheduleAtFixedRate(task, 1000);
- timer.scheduleAtFixedRate(task, 1000, 1000);

14. Manakah potongan kode program yang benar dalam penggunaan method untuk menjalankan waktu?

- ```
private void timerRun() {
 count++;
 ttCanvas.update();
 ttCanvas.repaint(200, 200, 20, 20);
}
```
- ```
protected void startApp() {  
    display = Display.getDisplay(this);  
    ttCanvas = new CounterDownCanvas(this);  
    display.setCurrent(ttCanvas);  
}
```
- ```
void startTimer() {
 timer = new Timer();
 timerTask = new TimerTask() {
 public void run() {
 timerRun();
 }
 };
 timer.scheduleAtFixedRate(timerTask, 1000, 1000);
}
```
- ```
protected void showNotify() {  
    midletTT.startTimer();  
}
```
- ```
protected void update() {
 points--;
 if(points < 1) {
 hideNotify();
 }
}
```

15. Apa maksud dari method dibawah ini?

```
protected void showNotify() {
 midletTT.startTimer();
}
```

- Menjalankan pertama kali method startTimer pada class midlet.
- Menjalankan pertama kali method startTimer pada class canvas.
- Menjalankan terakhir method startTimer pada class midlet.
- Menjalankan terakhir method startTimer pada class canvas.
- Menghentikan method startTimer pada akhir program pada class midlet.

Submit