Passing Parameters to a Java Thread

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Lesson

• Understand how Java threads support concurrency
• Learn how our case study app works
• Know alternative ways of giving code to a thread
• Learn how to pass parameters to a Java thread
Passing Parameters to a Java Thread
The run() methods defined in Java Thread & Runnable take no parameters.

This raises the question of how to pass parameters to a Java thread!
Passing Parameters to a Java Thread

- Parameters passed to run() can be supplied via one of two other means
Passing Parameters to a Java Thread

- Parameters passed to `run()` can be supplied via one of two other means, e.g.
- As parameters to a class constructor

```java
public class GCDRunnable extends Random implements Runnable {
    // Class definition...
}
```

• Parameters passed to run() can be supplied via one of two other means, e.g.
• As parameters to a class constructor

```java
class GCDRunnable extends Random implements Runnable {
    private final MainActivity mActivity;
    ...
}
```

*Define field(s) to store parameters passed to a runnable or thread object*
Passing Parameters to a Java Thread

- Parameters passed to run() can be supplied via one of two other means, e.g.
- As parameters to a class constructor

```java
public class GCDRunnable extends Random implements Runnable {
    private final MainActivity mActivity;

    public GCDRunnable(MainActivity mainActivity) {
        mActivity = mainActivity;
    }

    // ...
Parameters passed to run() can be supplied via one of two other means, e.g.:

- As parameters to a class constructor

```java
public class GCDRunnable extends Random implements Runnable {
    private final MainActivity mActivity;

    public GCDRunnable(MainActivity mainActivity) {
        mActivity = mainActivity;
    }

    public void run() {
        final String threadString = " with thread id " + Thread.currentThread();
        mActivity.println("Entering run()" + threadString);
        ...
    }
}
```

Use the field(s) within the thread’s run() hook method to customize its behavior
Passing Parameters to a Java Thread

- Parameters passed to run() can be supplied via one of two other means, e.g.
- As parameters to a class constructor

```java
public class GCDRunnable extends Random implements Runnable {
    private final MainActivity mActivity;

    public GCDRunnable(MainActivity mainActivity) {
        mActivity = mainActivity;
    }

    public void run() {
        final String threadString = " with thread id " + Thread.currentThread();
        mActivity.println("Entering run()" + threadString);
        ...
    }
}
```

```java
public class MainActivity ... { ... 
    public void runRunnable(View v) { ... 
        new Thread(new GCDRunnable(this));
    }
    ...
}
```

*Pass the parameter(s) when the runnable or thread is created*
Passing Parameters to a Java Thread

- Parameters passed to run() can be supplied via one of two other means, e.g.
  - As parameters to a class constructor
  - As parameters to “setter” methods

```java
public class GCDThread extends Thread {
```

Passing Parameters to a Java Thread

- Parameters passed to run() can be supplied via one of two other means, e.g.
  - As parameters to a class constructor
  - As parameters to “setter” methods

```java
public class GCDThread extends Thread {
    private MainActivity mActivity;
    private Random mRandom;
    ...
}
```

Define field(s) to store parameters passed to a runnable or thread object
Parameters passed to run() can be supplied via one of two other means, e.g.

- As parameters to a class constructor
- As parameters to "setter" methods

```java
public class GCDThread extends Thread {
    private MainActivity mActivity; private Random mRandom;

    public GCDThread setActivity(MainActivity activity) {
        mActivity = activity; return this;
    }

    public GCDThread setRandom(Random random) {
        mRandom = random; return this;
    }
    ...

Define setter methods that update field(s)
```
Passing Parameters to a Java Thread

- Parameters passed to run() can be supplied via one of two other means, e.g.
  - As parameters to a class constructor
  - As parameters to “setter” methods

```java
public class GCDThread extends Thread {
    private MainActivity mActivity; private Random mRandom;

    public GCDThread setActivity(MainActivity activity) {
        mActivity = activity; return this;
    }

    public GCDThread setRandom(Random random) {
        mRandom = random; return this;
    }
    ...
```

See [en.wikipedia.org/wiki/Fluent_interface](en.wikipedia.org/wiki/Fluent_interface)

Note use of “fluent interfaces,” which enables “chaining” of method calls
Passing Parameters to a Java Thread

- Parameters passed to run() can be supplied via one of two other means, e.g.
  - As parameters to a class constructor
  - As parameters to “setter” methods

```java
public class GCDThread extends Thread {
    private MainActivity mActivity; private Random mRandom;

    public GCDThread setActivity(MainActivity activity) {
        mActivity = activity; return this; }

    public GCDThread setRandom(Random random) {
        mRandom = random; return this; }

    ...

    public void run() {
        mActivity.println("Entering run()" + threadString);
        ...
        int number1 = mRandom.nextInt();
        int number2 = mRandom.nextInt(); ...
    }
```

Use the fields within the thread’s run() hook method to customize its behavior
Passing Parameters to a Java Thread

- Parameters passed to run() can be supplied via one of two other means, e.g.
  - As parameters to a class constructor
  - As parameters to “setter” methods

```java
public class GCDThread extends Thread {
    ...
}

public class MainActivity ...
    public void runThread(View v) {
        Thread thread =
            new GCDThread()
                .setActivity(this)
                .setRandom(new Random());
    ...
}
```

*Use the fluent interface to pass parameter(s) when the runnable or thread is created*
End of Passing Parameters to a Java Thread