

Programmazione J2ME

Lezione 1

Le MIDlet



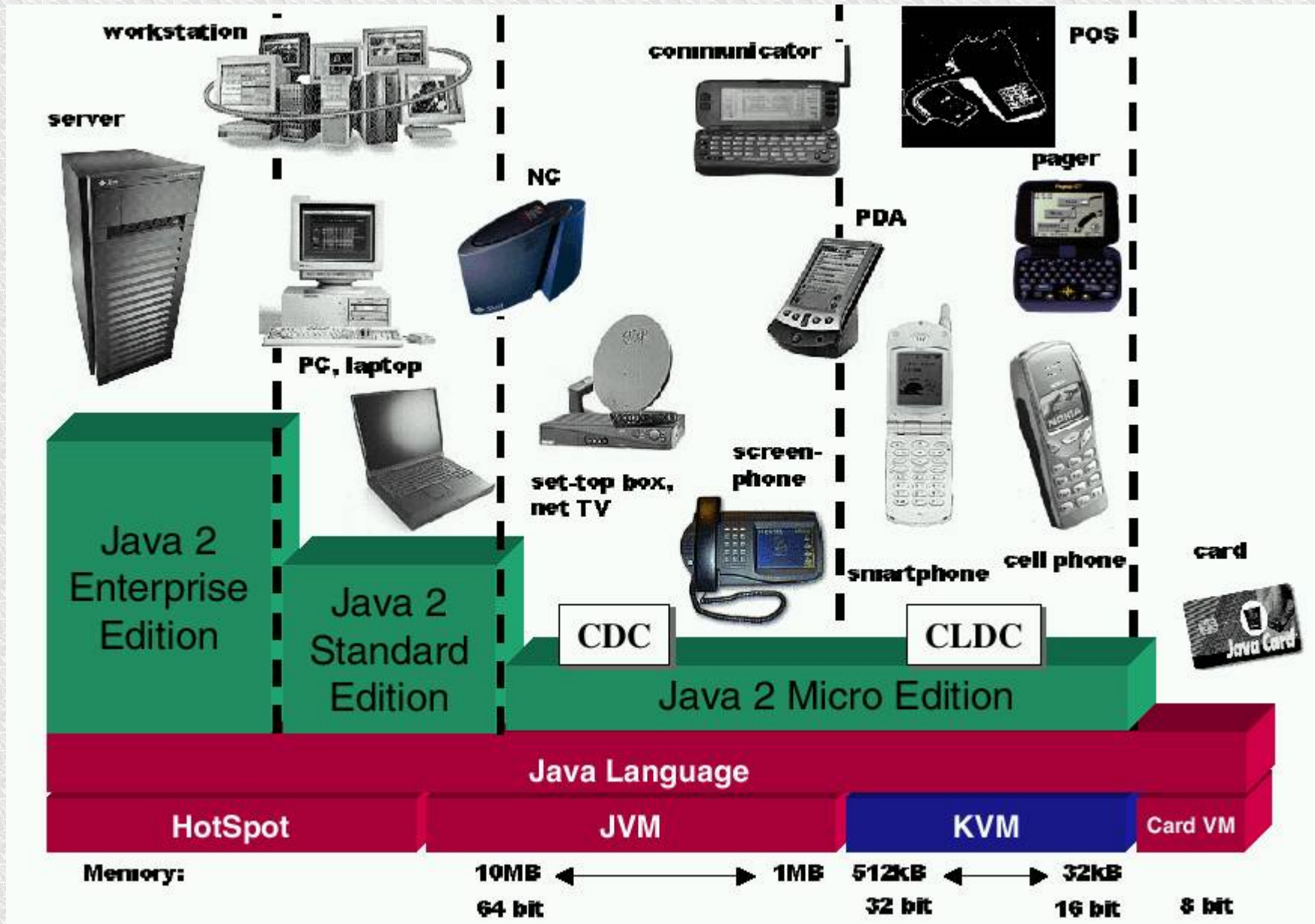
Cosa impareremo oggi?

- Cosa è J2ME
- I profili MIDP ed i livelli di configurazione
- Il sistema di sicurezza di J2ME
- Le MIDlet
- I package J2ME
- Distribuzione delle applicazioni

Introduzione (1)

- J2ME (Java2 Mobile Edition) deriva da J2SE (Java2 Standard Edition)
- È ottimizzato e “leggero” per girare su cellulari, smartphone e PDA di ultima generazione
- Si affianca a J2EE (Java2 Enterprise Edition) dedicato invece alle applicazioni server

Introduzione (2)



Introduzione (3)

- Un dispositivo abilitato J2ME consente di scaricare un'applicazione *MIDlet* di pochi kbyte (alcuni terminali permettono ad esempio applicazioni di max 10KB ciascuna e 50 applicazioni totali sul terminale)
- Tali applicazioni vengono installate sul device e possono essere utilizzate anche off-line
- La modalità di installazione e configurazione di J2ME e delle *MIDlet* varia da terminale a terminale

Introduzione (4)

- Oltre alle limitazioni di memoria e di potenza di elaborazione, J2ME deve adattarsi anche a display particolarmente ridotti (ad esempio un piccolo telefono cellulare può disporre di 12.288 pixel, 96×128 , un PDA di 20.000)
- Ultimamente l'evoluzione tecnologica in questo senso tende ad abbattere queste limitazioni con display sempre più ampi e con sempre più colori

Introduzione (5)

- Il tutto si basa su una Java Virtual Machine, presente nei terminali, che interpreta ed esegue l'applicazione *MIDlet*
- J2ME comprende due livelli di configurazione (CDC e CLDC) ed inoltre il livello “Profile” MIDP
- Inoltre esistono attualmente molte librerie aggiuntive ed alcune proposte di librerie per estendere la JVM in diversi contesti

I profili MIDP (1)

- L'idea della Sun è quella di partire da una base comune a tutti i dispositivi
- Essa contiene un set di API che fornisce le funzionalità base per un qualunque device a limitata configurazione
- Saranno poi i produttori di particolari categorie ad aggiungere funzionalità al CLDC, sotto forma di librerie, generando così quello che è un “profilo”

I profili MIDP (2)

- Il primo profilo è MIDP (Mobile Information Device Profile) per terminali wireless in grado di instaurare connessioni HTTP
- Esistono 2 versioni MIDP: 1.0 e 2.0
- MIDP 2.0 è un'estensione delle librerie MIDP 1.0 e contiene classi per la gestione di audio, video, fotocamera, connessioni seriali, bluetooth, ecc.

I profili MIDP (3)

- Alcuni compiti di MIDP sono:
 - gestione del ciclo di vita delle applicazioni (caricamento, esecuzione, distruzione)
 - gestione della GUI (dispositivi di input/output)
 - salvataggio persistente dei dati
 - networking (implementazione dei protocolli)

I profili MIDP (4)

- Connected Device Configuration (CDC)
 - per dispositivi “things that you plug into a wall”, cioè collegati in rete, possibilmente *always on*, ad alta banda e con buona potenza di calcolo e memoria (TV set-top box, PDA di fascia alta)
 - 512 kB minimo di memoria per l'applicazione Java (codice) e 256 kB minimo di memoria runtime, allocata dall'applicazione
 - La JVM è quasi pari a quelle presenti nei PC con un sottoinsieme ampio delle API di J2SE (esempio PersonalJava)

I profili MIDP (5)

- Connected Limited Device Configuration (CLDC)
 - Per dispositivi “things that you hold in your hand”, caratterizzati da connettività wireless, ridotta banda, accesso discontinuo e bassa potenza di calcolo e memoria
 - 128 kB di memoria per Java (codice) e 32 kB di memoria runtime, interfaccia utente ridotta, bassi consumi, alimentazione a batteria
 - Tutti i cellulari (J2ME-enabled) attualmente in commercio hanno una configurazione CLDC

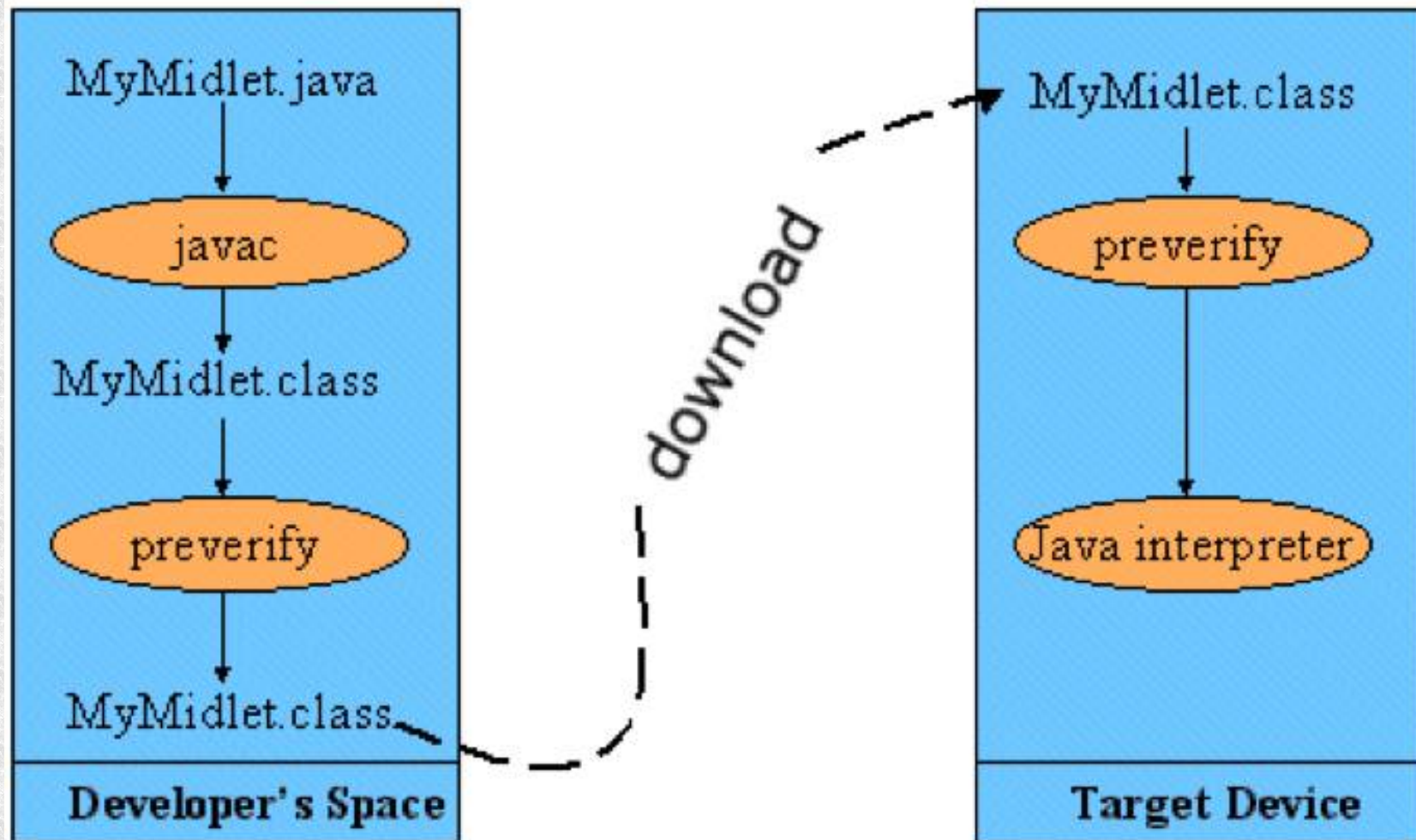
Sicurezza (1)

- La configurazione CLDC definisce un modello conosciuto come “sandbox” che vieta l’accesso a tutto quello che è fuori dalla “scatola”
- Il sistema è esattamente identico (nella logica) a quello delle sandbox delle applet

Sicurezza (2)

- Un ultimo aspetto è quello della verifica di integrità dei file di classe
- Pre-verifica (in compilazione)
 - si esegue un programma per l'inserimento di attributi aggiuntivi nel file di classe
- Verifica (sull'apparecchio)
 - caricata una classe pre-verificata, il verificatore interno ne percorre tutte le istruzioni e se non trova “inconsuetudini” la firma, permettendone quindi l'installazione

Sicurezza (3)



La prima MIDlet (1)

- La scrittura di una *MIDlet* è abbastanza semplice ed intuitiva
- Si deve:
 - estendere la classe `javax.microedition.midlet.MIDlet`
 - implementare i metodi *startApp*, *pauseApp* e *destroyApp*

La prima MIDlet (2)

- *void startApp()*
 - segnala alla *MIDlet* che è stata posta nello stato di attività
- *void pauseApp()*
 - segnala alla *MIDlet* il suo inserimento nello stato di pausa
- *void destroyApp(boolean unconditional)*
 - impone alla *MIDlet* di terminare le proprie attività ed entrare nello stato *destroyed*

La prima MIDlet (4)

Esempi

HelloJ2ME_a1

HelloJ2ME_a2

HelloJ2ME_a3

I package J2ME (1)

- `javax.microedition.lcdui`
 - è il cosiddetto *User Interface Package*, contiene le classi per creare le interfacce utente sul display
- `javax.microedition.rms`
 - utilizzato per la memorizzazione persistente, fornisce i meccanismi che permettono di memorizzare e recuperare dati in modo persistente

I package J2ME (2)

- `javax.microedition.midlet`
 - definisce le interfacce ed il ciclo di vita delle *MIDlet*
- `javax.microedition.io`
 - detto anche il *Networking Package* include i supporti per la connessione alla rete da un CLDC
 - contiene molte interfacce ma una sola classe *Connector* utilizzata per creare gli oggetti *Connection*

I package J2ME (3)

- `java.io`
 - si occupa di gestire l'input e l'output da un generico stream di dati
- `java.lang`
 - include tutte le classi tipiche del linguaggio Java incluse nella J2SE
 - troviamo tutte le classi per i tipi di dato (Boolean, Byte, Character, Integer, Long, Short, String, StringBuffer) oltre alle classi base di sistema e per i Thread (Math, Object, Runtime, System, Thread, Throwable)

I package J2ME (4)

- `java.util`
 - include tutte le classi di utility incluse nella J2SE
 - in questo package troviamo le classi di utilità che un programmatore Java è abituato ad utilizzare (Calendar, Date, Hashtable, Random, Stack, Timer, TimerTask, TimeZone , Vector)

I package J2ME (5)

- `javax.microedition.lcdui.game` (MIDP 2.0)
 - fornisce una serie di classi che aiutano lo sviluppo di videogame
- `javax.microedition.media` (MIDP 2.0),
`javax.microedition.media.control` (MIDP 2.0)
 - forniscono le classi necessarie per lo sviluppo di applicazioni multimediali (audio e video)
- `javax.microedition.pki` (MIDP 2.0)
 - fornisce le classi per la certificazione di connessioni sicure

I package J2ME (6)

- Altre (non tutte) librerie opzionali proposte in MIDP 2.0
 - WMA 1.0: per gestire l'invio e la ricezione degli SMS, possibilità di “risvegliare” una MIDlet
 - WMA 2.0: aggiornamento con inclusione al supporto MMS
 - Java API for Bluetooth: per connessioni bluetooth
 - Mobile3D Graphics: librerie per gestire il 3D

Distribuzione delle applicazioni (1)

- Una volta terminato lo sviluppo di un'applicazione, per poterla distribuire è necessario creare un file che contenga tutte le classi e le risorse necessarie (immagini, audio, ecc.)
- Il file che si andrà a creare è un file JAR
- Questo file JAR, oltre alle classi Java ed alle risorse, ha al suo interno un particolare file, Manifest.MF, che si trova nella cartella META-INF dell'archivio

Distribuzione delle applicazioni (2)

- Nel Manifest si trovano degli attributi molto importanti per la definizione della *MIDlet*
 - MIDlet-Name: nome della *MIDlet* che apparirà nell'elenco delle applicazioni
 - MIDlet-Version: numero di versione della *MIDlet*
 - MIDlet-Vendor: autore della *MIDlet*
 - MicroEdition-Profile: MIDP-1.0/MIDP-2.0
 - MicroEdition-Configuration: CLDC-1.1/CDC
 - MIDlet-Icon: l'icona
 - MIDlet-Info-URL: URL per le info supplementari

JAR & JAD

- Oltre al JAR è necessario (almeno in MIDP 1.0) fornire un file JAD contenente alcuni parametri necessari al dispositivo per localizzare ed installare la *MIDlet*
 - MIDlet-Name: nome della *MIDlet* che verrà visualizzato
 - MIDlet-Version: numero di versione della *MIDlet*
 - MIDlet-Vendor: autore della *MIDlet*
 - MIDlet-Jar-URL: URL del file JAR (assoluto)
 - MIDlet-Jar-Size: dimensione del file JAR in byte

Suggerimenti di design (1)

- Lo sviluppo di applicazioni J2ME richiede l'utilizzo di particolari strategie di progettazione
- È molto importante progettare un'applicazione J2ME prima di passare alla stesura del codice
- Le limitate risorse dei dispositivi J2ME potrebbero rendere disastrose eventuali revisioni del progetto dovute a scarsa attenzione in fase progettuale

Suggerimenti di design (2)

- Mantenere il progetto semplice:
 - rimuovere caratteristiche non necessarie (fornendole eventualmente tramite applicazioni secondarie)
- “Piccolo è meglio”:
 - applicazioni piccole usano meno memoria e richiedono installazioni più brevi
- Minimizzare l'uso della memoria run-time:
 - ad esempio usando (quando possibile) tipi scalari al posto di oggetti

Suggerimenti di design (3)

- Non fare troppo affidamento sul garbage collector:
 - gestire la memoria nel modo più efficiente possibile (riutilizzo di oggetti, forzatura dei puntatori a null quando gli oggetti non servono più, ecc.)
- Usare variabili locali:
 - l'accesso alle variabili locali è più veloce dell'accesso alle variabili di istanza

Suggerimenti di design (4)

- Evitare la concatenazione di stringhe:
 - la concatenazione di stringhe decrementa le performance ed incrementa l'uso della memoria
- Usare i *Thread* ed evitare la sincronizzazione:
 - qualsiasi operazione che richiede più di 1/10 di secondo dovrebbe essere svolta da un *Thread* separato
- Utilizzare fortemente lo schema MVC:
 - MVC separa la logica applicativa dall'aspetto estetico e dai controlli utente