

Pobranie i instalacja środowisk

Aby móc sprawdzić przykłady zaprezentowane w serwisie, a także testować później z własnymi pomysłami, będziemy potrzebowali kilku niezbędnych rzeczy.

Są to przede wszystkim:

1. JDK, czyli środowisko, zawierające zarówno wszystkie standardowe pakiety składające się na J2SE, jak i standardową maszynę Javy.

Najnowsze JDK znaleźć można tutaj:

<http://java.sun.com/j2se/downloads.html>

2. Środowisko uruchomieniowe, czyli tzw. emulator, potrzebny nam będzie do wizualizacji tego, co napiszemy, ale i jako maszyna wirtualna, czyli KVM.

Można pobrać emulatory telefonów rzeczywistych ze stron ich producentów, ale my korzystać będziemy ze wzorcowej implementacji, jaką proponuje Sun. "J2ME Wireless Toolkit" proponuje oprócz emulacji skompilowanego kodu, również łatwą jego kompilację i preweryfikację oraz łatwe tworzenie archiwów JAR. Screeny prezentowane w serwisie robione są właśnie z tego programu.

Najświeższą wersję Toolkita pobrać można z poniższej lokalizacji:

<http://java.sun.com/products/j2mewtoolkit>

3. Do tego potrzebny nam będzie jakikolwiek edytor tekstowy. Wygodniej oczywiście, jeśli będzie kolorował tekst. Najlepsze są proste edytory. Jeśli ktoś chciałby skorzystać ze zintegrowanego środowiska klasy IDE, zachęcam do poniższych:

1. [Sun ONE Studio 4 update 1, Mobile Edition](#) - darmowe IDE Sun'a, skierowane do developerów Javy ME.

2. [Borland JBuilder i MobileSet](#) - do ściągnięcia 30-dniowa wersja próbna.

Dokumentacja. Może nie jest wymagana przy pracy, tak jak JDK czy emulator, ale w praktyce nie można bez niej pracować. Najwygodniej jest oczywiście pracować z dokumentacją wygenerowaną przez javadoc.

API poszczególnych technologii, możemy znaleźć w odpowiedniej części JCP (<http://jcp.org>) lub bezpośrednio ze stron domowych Javy: <http://java.sun.com/api>.

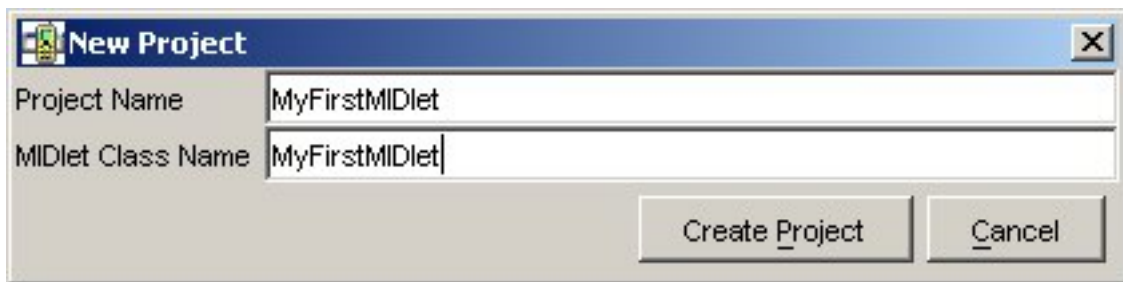
Instalujemy więc JDK i koniecznie J2ME Wireless Toolkit, bo to na nim prezentować będziemy przykłady.

Będąc tak przygotowanym do pracy jesteśmy już gotowi do przejścia do kolejnego etapu...

Tworzenie nowego projektu

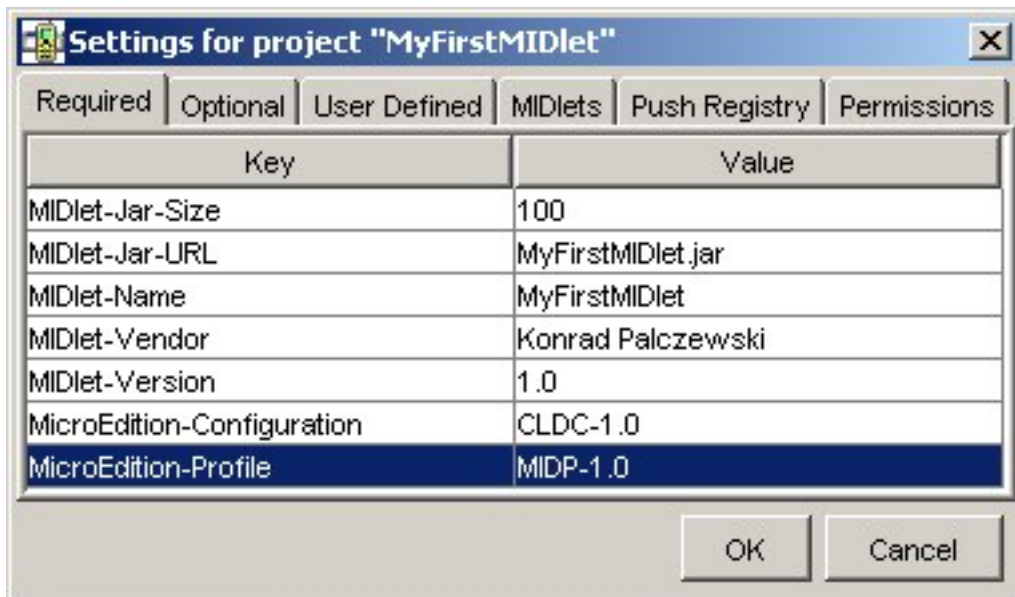
Uruchamiamy więc nasz Toolkit, i tworzymy nowy projekt. Jak się domyślamy, służy do tego przycisk po lewej oznaczony "New Project...". Ja tworząc przykłady dla tego artykułu budowałem projekt zaprezentowany w artykule "Pierwszy MIDlet".

I tak po wciśnięciu przycisku "New Project...", pojawia się okno, w którym wpisać musimy nazwę naszego projektu i - co ważniejsze - nazwę klasy głównej naszego MIDletu:



Tworzenie nowego projektu...

Po potwierdzeniu wprowadzonej nazwy, pojawi się okno, w którym wypełnić możemy - omówione w artykule "Archiwa JAR i deskryptory JAD" - konieczne, opcjonalne, oraz własne atrybuty prezentowane w plikach JAD i `manifest.mf`, oraz dodać MIDlety należące do jednego *MIDlet Suit*:



... wypełnianie jego deskryptorów.

I teraz całą hierarchię katalogów dla naszego projektu mamy stworzoną. Jeśli więc `%WTK20%` będzie lokalizacją, gdzie zainstalowany jest Wireless Toolkit, to:

1. w katalogu "`%WTK20%\apps\MyFirstMIDlet\src`" umieszczać będziemy stworzony przez nas kod,
2. w katalogu "`%WTK20%\apps\MyFirstMIDlet\res`" umieszczać będziemy zasoby, jak np. ikony czy pliki tekstowe, z których chcemy skorzystać w trakcie działania aplikacji,
3. w katalogu "`%WTK20%\apps\MyFirstMIDlet\lib`" umieszczać będziemy biblioteki, które chcemy dołączyć do projektu (jako pliki JAR lub ZIP).

W przypadku naszego bardzo prostego pierwszego MIDletu, w którym brakuje wykorzystania ikon, czy innych zasobów dołączonych do pliku JAR, jedyny plik z kodem źródłowym `MyFirstMIDlet.java` wylądował w katalogu `%WTK20%\apps\MyFirstMIDlet\src`.

Zabieramy się więc do pracy. Zapraszam do kolejnych artykułów...