Wielojęzyczne aplikacje

Na podstawie:

- Dr. International, Developing International Software 2nd Edition, Microsoft 2003
- G. Smith-Ferrier, .NET Internationalization: The Developer's Guide to Building Global Windows and Web Applications, Addison Wesley Proffesional, 2006

Ustawienia regionalne *(Locale)*

- Zbiór cech środowiska użytkownika zależnych od języka, kraju, regionu geograficznego lub zwyczajów kulturalnych
- W nomenklaturze informatycznej – zbiór informacji związanych z miejscem geograficznym lub uwarunkowaniami kulturalnymi
  - nazwa i identyfikator używanego języka
  - zbiór znaków używanych w tekstach (skrypt)
  - zwyczaje kulturalne
  - określa: układ klawiatury, format daty, czasu, liczb i waluty
- W Windows XP obsługiwanych jest 135 zestawów ustawień regionalnych, w Windows Vista ponad 200

Przykłady ustawień regionalnych

	Angielski (US)	Polski	Japoński	Arabski (ZEA)
Kraj	Stany Zjednoczone	Polska	Japonia	Zjednoczone Emiraty Arabskie
Język	Angielski	Polski	Japoński	Arabski
Skrypt	Latin	Central Europe	Kana, kanji	Arabic
Kierunek pisania tekstu	Lewa → prawa	Od lewej do prawej	Lewa → prawa lub pionowo odwrotnie	Od prawej do lewej
Strona kodowa	1252	1250	932	1256
Symbol waluty	$	Zł	¥	د.إ.
Długa data	January 27, 2004	27 stycznia 2004	2004 年1月27日	٢٧ يناير, ٢٠٠٤
Krótka data	1/27/04	2004-01-27	04/01/27	04/01/27
Format czasu	1:00 pm	13:00	13:00	١:٠٠ م
Kalendarz	Gregoriański	Gregoriański	Gregoriański (zlokalizowany)	Gregoriański (zlokalizowany)
Papier	U.S. Letter	A4	A4	A4
Symbol dziesiętny	.	,	.	,
Separator listy	,	;	,	;
Grupowanie cyfr	,	(space)	,	,

Wspólna kompilacja

- Zalety:
  - zmniejszenie kosztów i problemów
  - unikanie warunkowych kompilacji
  - jeden kod źródłowy tworzony przez jeden zespół
  - możliwość równoczesnego rozpoczęcia sprzedaży wszystkich wersji językowych
  - uproszczenie procesu uaktualniania oprogramowania
  - ułatwienie pracy klientów

Globalizacja

Proces tworzenia programu w taki sposób, by nie był zależny od jednego języka lub lokalizacji.

Zalecenia

- Używać Unicode
- Nie stosować żadnych założeń związanych z formatem wyświetlanych danych
- Nie ograniczać możliwości wprowadzania tekstu poprzez dostosowanie do jednego sposobu kodowania
- Nie uzależniać pracy programu od obecności wybranej czcionki
- Brać pod uwagę możliwość użycia skomplikowanych skryptów znakowych
- Pozwolić użytkownikowi na wybór języka, w którym będzie prezentowany interfejs
- Zadbać o lokalizowalność
- Zadbać o obsługę innego kierunku wyświetlania tekstów

Unicode

- UTF-8
  - 1 bajt dla znaków US-ASCII (U+0000 - U+007F)
  - 2 bajty dla symboli języków łacińskich, greckiego, cyrylicy, armeńskiego, hebrajskiego, syryjskiego i thaana (U+0080 - U+07FF)
  - 3 bajty dla *Basic Multilingual Plane*
  - 4 bajty dla *UFT-16 surrogate pairs*
- UTF-16
  - 2 bajty na znak
  - *surrogate pairs* – pary znaków UTF-16
- UTF-32
  - 4 bajty na znak

Unicode w Windows

- Windows 95, 98 i Me nie mają wbudowanej obsługi Unicode
  - Microsoft Layer for Unicode – dodatek umożliwiający uruchamianie aplikacji opartych na Unicode
- Windows NT 3.51 – pierwsza wersja z obsługą Unicode
- W Windows NT 4, 2000, XP, 2003 Unicode jest kodowaniem używanym na poziomie systemowym
  - teksty są zapisywane w UTF-16
  - teksty aplikacji nieużywających Unicode są konwertowane

Polskie znaki diakrytyczne

		ą	ć	ę	ł	ń	ó	ś	ź	ż
Unicode	Dec	50309	50311	50329	50562	50564	50099	50587	50618	50620
	Hex	C485	C487	C499	C582	C584	C3B3	C59B	C5BA	C5BC
ISO-8859-2	Dec	177	230	234	179	241	243	182	188	191
	Hex	B1	E6	EA	B3	F1	F3	B6	BC	BF
Windows CP-1250	Dec	185	230	234	179	241	243	156	9C	191
	Hex	B9	E6	EA	B3	F1	F3	159	9F	BF

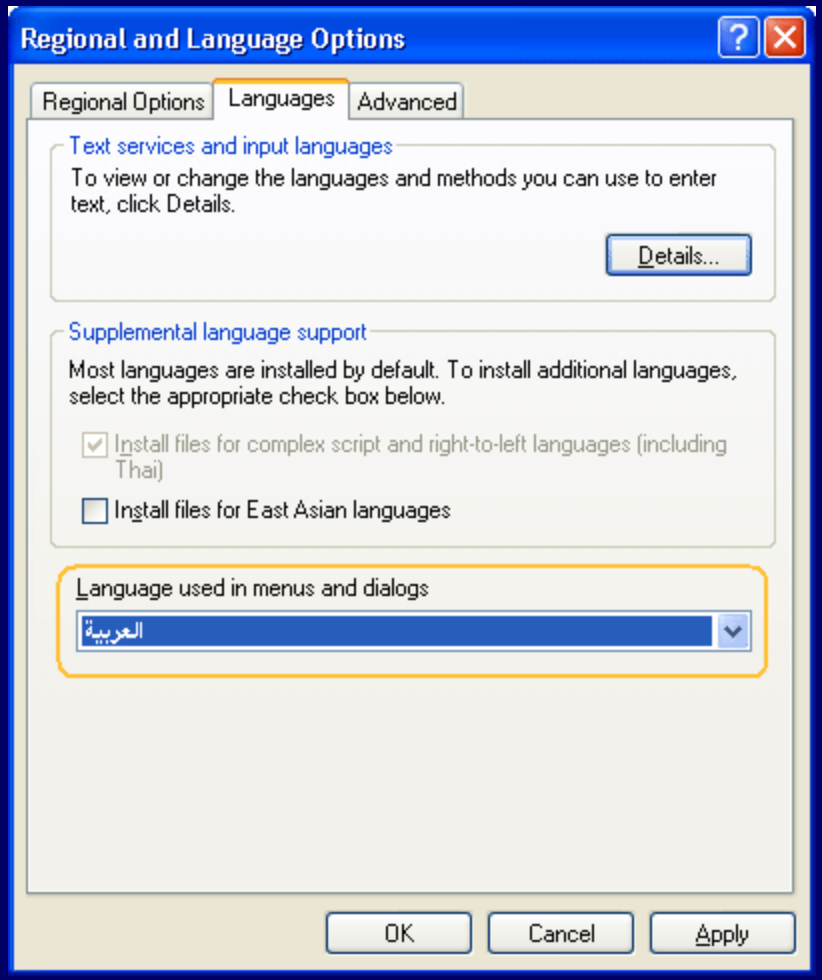
		Ą	Ć	Ę	Ł	Ń	Ó	Ś	Ź	Ż
Unicode	Dec	50308	50310	50328	50561	50563	50067	50586	50617	50619
	Hex	C484	C486	C498	C581	C583	C393	C59A	C5B9	C5BB
ISO-8859-2	Dec	161	198	202	163	209	211	166	172	175
	Hex	A1	C6	CA	A3	D1	D3	A6	AC	AF
Windows CP-1250	Dec	165	198	202	163	209	211	140	143	175
	Hex	A5	C6	CA	A3	D1	D3	8C	8F	AF

Różnice w ustawieniach regionalnych

- Data i kalendarz
- Czas
- Waluta
- Zmiana wielkości liter
- Sortowanie i porównywanie tekstów
- Format liczb
- Adresy
- Rozmiar papieru
- Numery telefonów
- Jednostki miary

Multilingual User Interface (MUI)

- Zbiór plików z zasobami dostosowanymi do różnych lokalizacji, możliwy do zainstalowania w Windows 2000, XP Professional oraz Vista Enterprise i Ultimate
- Daje możliwość zmiany języka interfejsu użytkownika na poziomie systemowym



Lokalizowalność

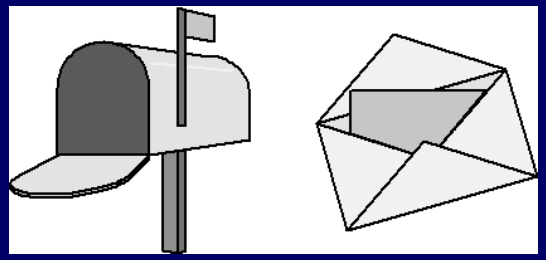
Zaprojektowanie programu w taki sposób, by tworzenie wersji lokalnych było możliwe bez modyfikacji kodu źródłowego.

Wyodrębnienie zasobów do lokalizacji

- Elementy lokalizowane:
  menu, komunikaty, okna dialogowe, teksty, obrazki, dźwięki, elementy pasków narzędzi i paska stanu
- Typy tekstów:
  - elementy interfejsu użytkownika – muszą zostać przetłumaczone
  - nazwy czcionek, folderów, użytkowników itp. – powinny być tłumaczone, ale ostrożnie
  - informacje wykorzystywane podczas śledzenia pracy programu – nie muszą być tłumaczone
  - teksty, od których zależy praca programu (nazwy kluczy w rejestrze, komunikaty wymieniane między modułami itp.) – nie mogą być tłumaczone

Potencjalne problemy

- Zmiana rozmiaru elementów UI
- Dostosowanie zawartości kontrolek
- Obrazki



- Zalecenia dotyczące tłumaczenia:
  - proste zdania i wyrażenia
  - zgodność z zasadami pisowni
  - uwzględnianie uwarunkowań lokalnych i kulturalnych (w tekstach, obrazkach i plikach multimedialnych)
  - stosowanie tekstów łatwych w tłumaczeniu (pod względem technicznym)
  - zapewnienie poprawnego działania systemu pomocy

Lokalizacja

Proces dostosowania programu do lokalnego rynku.

Lokalizacja

- Tłumaczenie
  - wszystkie teksty, menu, okna dialogowe, przyciski, kreatory, pomoc, drukowana dokumentacja, opakowanie, etykiety na CD itp.
  - pliki multimedialne (synchronizacja mówionego tekstu)
- Dostosowanie do ustawień lokalnych
  - waluta, adresy, liczby, daty
  - sortowanie
  - czcionki
  - orientacja tekstu od prawej do lewej
- Zlokalizowana wersja powinna wyglądać i działać tak, jakby została stworzona w docelowym miejscu

Elementy podlegające lokalizacji

- Teksty
- Układ
- Grafika i multimedia
- Skróty klawiszowe
- Czcionki
- Zbiory znaków (wprowadzanie i wyświetlanie)
- Instalatory, opakowania, etykiety

Testowanie

Krzysztof Mossakowski
Wydział Matematyki i Nauk Informacyjnych Politechniki Warszawskiej
http://www.mini.pw.edu.pl/~mossakow

Wdrożenia zlokalizowanych aplikacji

- Proces instalacji w środowiskach z innymi ustawieniami językowymi
- Proces instalacji w systemie z zainstalowanym Multilingual User Interface
- Funkcjonalność aplikacji w środowisku z innymi ustawieniami regionalnymi
- Proces odinstalowania aplikacji

Obsługa tekstów

- Wprowadzanie tekstów w różnych językach
- Operacja na schowku
- Wyświetlanie i drukowania
- Niezależność od czcionek
- Teksty w interfejsie użytkownika
- Poruszanie kursorem (szerokości znaków, orientacja tekstu)
- Obsługa znaków zapisanych na dwóch bajtach
- Obsługa używanego w systemie sposobu kodowania
- Rozmiary buforów dla tekstów
- Niezależne od języka przechowywanie danych

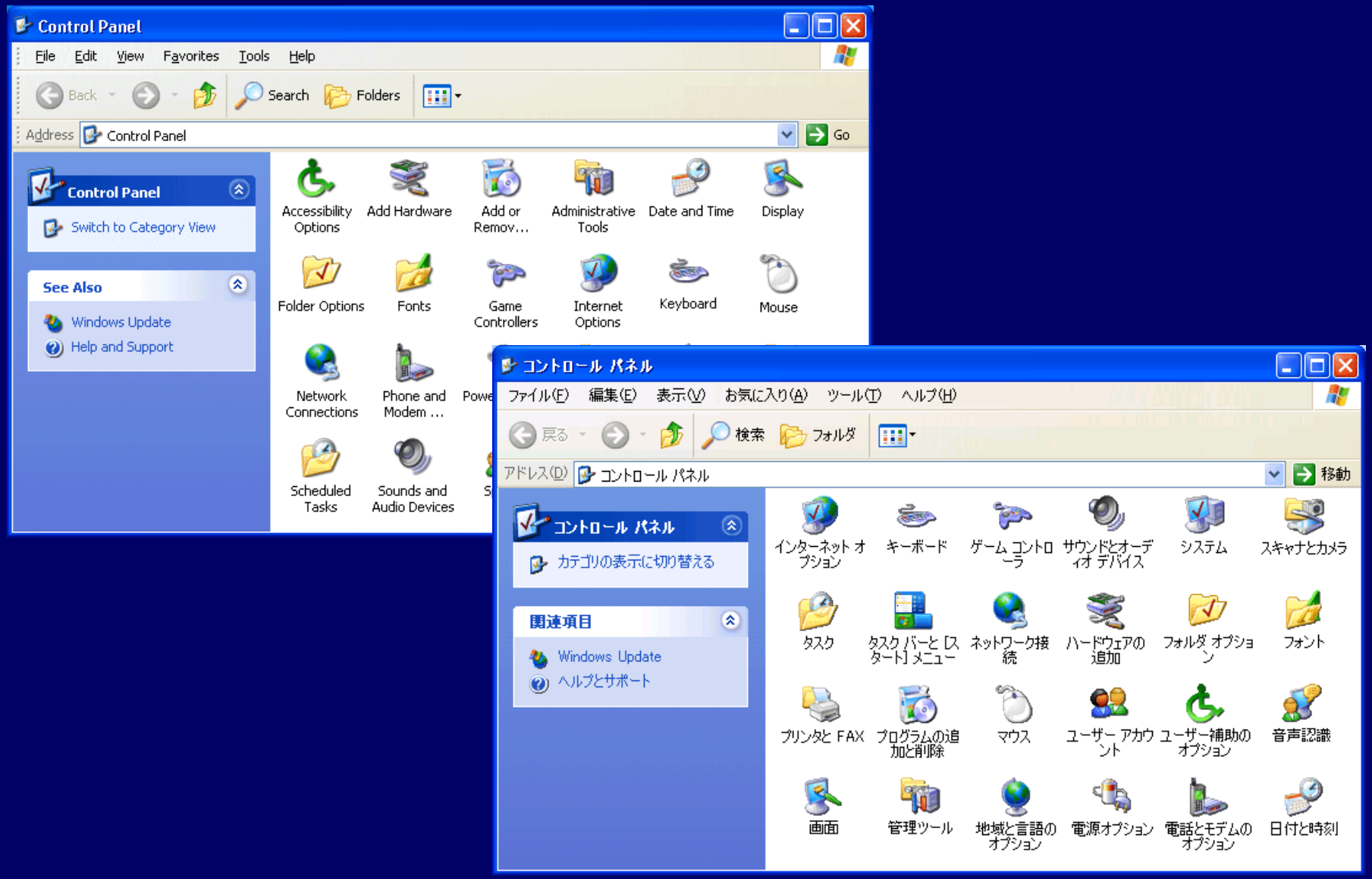
Wpływ ustawień regionalnych

- Użycie wewnętrznej reprezentacji danych niezależnej od ustawień regionalnych
  - odpowiednio formatować przed wyświetleniem
- Zachowanie charakterystycznych cech ustawień regionalnych
  - A.M./P.M., nazwy miesięcy, symbol waluty itp.
- Niezależność od separatorów
- Rozmiar papieru i kopert
- Niezależność od systemu metrycznego

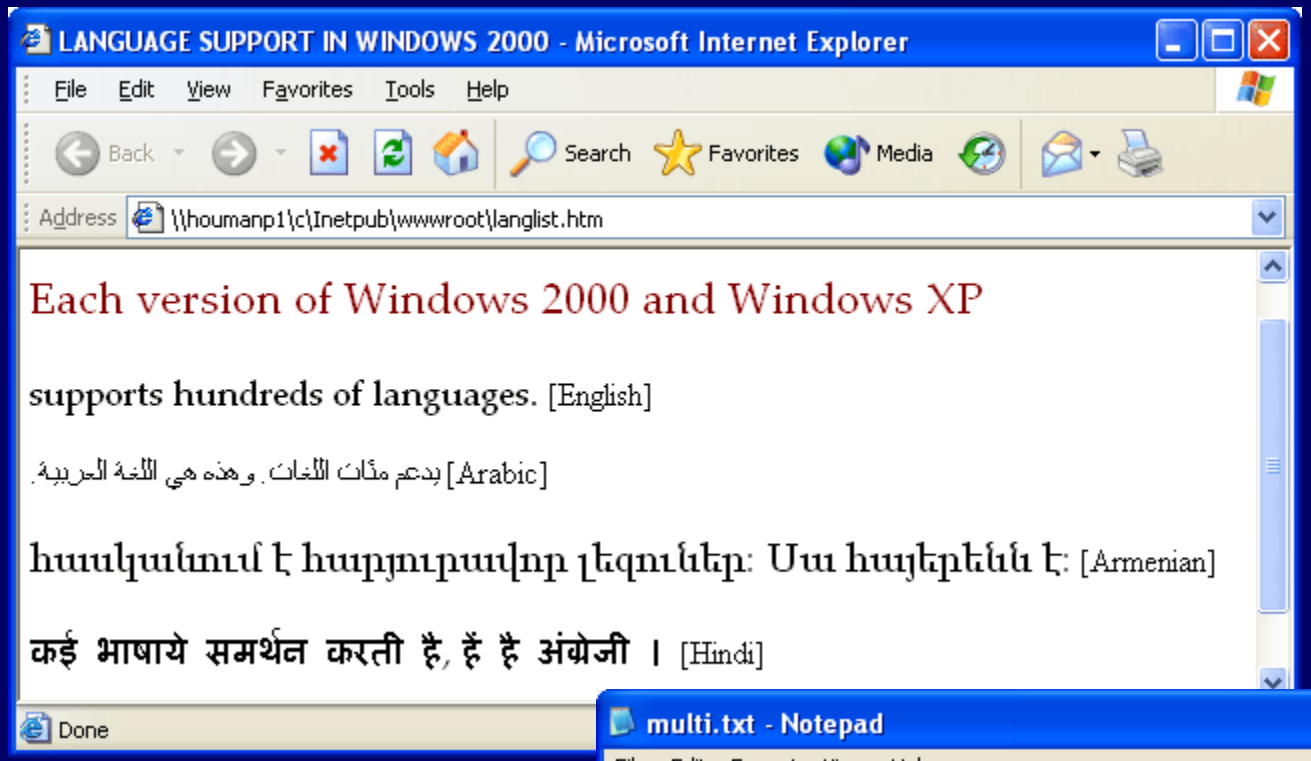
Lokalizowalność

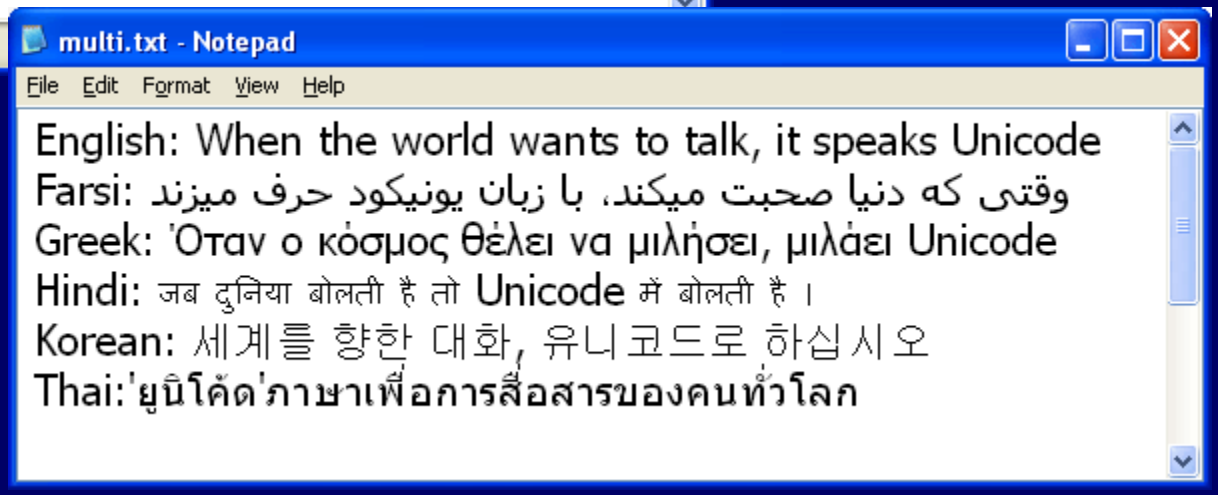
- Wyodrębnienie lokalizowanych zasobów
- Niezależne od języka użycie zasobów
- Unikanie zależności pomiędzy tekstami
- Nazwy plików niezależne od języka
- Niezależność od rozmiaru czcionki i rozdzielczości
- Poprawne działanie elementów niezależnych od ustawień
- Dostosowanie wielkości elementów interfejsu
- Poprawne działanie w innej orientacji *(Mirroring)*
- Lokalizowalność zasobów innych niż teksty
- Dynamiczne tworzenie tekstów
- Pobieranie tekstów systemowych (nazwa użytkownika, foldery itp.)

Przykłady


Control Panel
File Edit View Favorites Tools Help
Back Search Folders
Address Control Panel
Go
Control Panel
Switch to Category View
See Also
Windows Update
Help and Support
Accessibility Options
Add Hardware
Add or Remov...
Administrative Tools
Date and Time
Display
Folder Options
Fonts
Game Controllers
Internet Options
Keyboard
Mouse
Network Connections
Phone and Modem ...
Scheduled Tasks
Sounds and Audio Devices
コントロール パネル
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
戻る 検索 フォルダ
アドレス(D) コントロール パネル
移動
コントロール パネル
カテゴリの表示に切り替える
関連項目
Windows Update
ヘルプとサポート
インターネット オプション
キーボード
ゲーム コントローラ
サウンドとオーディオ デバイス
システム
スキャナとカメラ
タスク
タスク バーと [スタート] メニュー
ネットワーク接続
ハードウェアの追加
フォルダ オプション
フォント
プリンタと FAX
プログラムの追加と削除
マウス
ユーザー アカウント
ユーザー補助のオプション
音声認識
画面
管理ツール
地域と言語のオプション
電源オプション
電話とモデムのオプション
日付と時刻

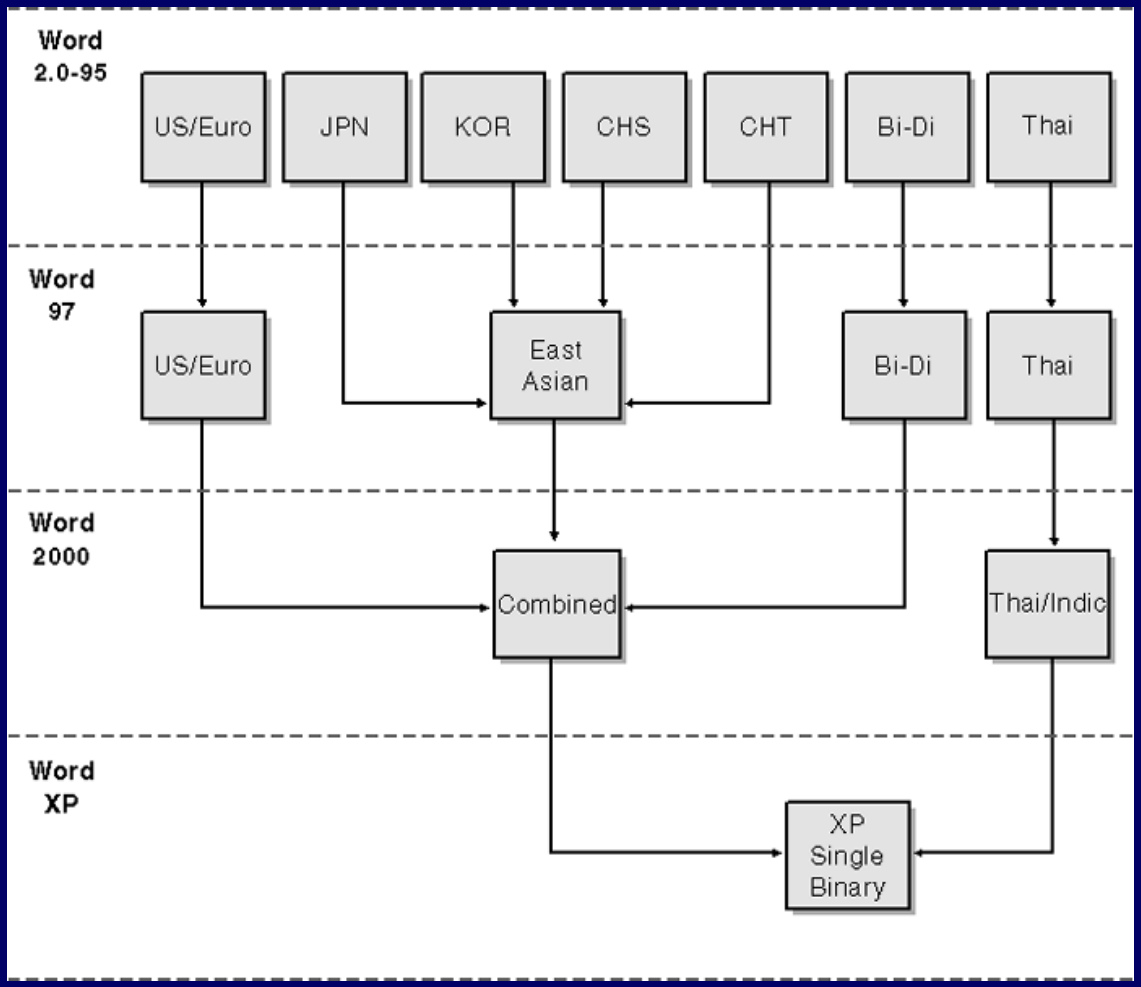

المفضلة
ملف تحرير عرض المفضلة أدوات تعليمات
الخلف بحث مجلدات
عنوان C:\Documents and Settings\Administrator\Favorites انتقال
المجلدات
سطح المكتب
المستندات
جهاز الكمبيوتر
(A:)قرص مرن ٣,٥
(C:)القرص المحلي
D
Documents and Settings
Administrator
Cookies
My Documents
الصور
الموسيقى
المفضلة
Links
سطح المكتب
Links
MSN.com
اختصار إنترنت
ك.ب 1
دليل محطة الراديو
اختصار إنترنت
ك.ب 1
Windows Media Player
ص ١١:٥٢
EN
المفضلة
ابدأ


LANGUAGE SUPPORT IN WINDOWS 2000 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address \\houmanp1\c\Inetpub\wwwroot\langlist.htm
Each version of Windows 2000 and Windows XP
supports hundreds of languages. [English]
يدعم مئات اللغات. وهذه هي اللغة العربية. [Arabic]
հասկանում է հարյուրավոր լեզուներ: Սա հայերենն է: [Armenian]
कई भाषाये समर्थन करती है, हैं है अंग्रेजी । [Hindi]
Done


multi.txt - Notepad
File Edit Format View Help
English: When the world wants to talk, it speaks Unicode
Farsi: وقتی که دنیا صحبت میکند، با زبان یونیکود حرف میزند
Greek: Όταν ο κόσμος θέλει να μιλήσει, μιλάει Unicode
Hindi: जब दुनिया बोलती है तो Unicode में बोलती है ।
Korean: 세계를 향한 대화, 유니코드로 하십시오
Thai:'ยูนิโค้ด'ภาษาเพื่อการสื่อสารของคนทั่วโลก


Hello سلام

- Historia tworzenia wersji językowych w programie Microsoft Word



.NET Framework

Języki w .NET Framework

- .NET Framework od wersji 2.0 jest dostępny w 24 językach, także w polskim
- Pakiet redystrybucyjny .NET Framework jest pojedynczym pakietem obsługującym wszystkie języki
  - angielska wersja .NET Framework jest zawsze instalowana
- Tłumaczenia tekstów używanych przez .NET Framework są dostępne w *.NET Framework Language Packs*
  - w jednym systemie może być zainstalowanych wiele rozszerzeń języków
  - aplikacje napisane w .NET będą używały zasobów z rozszerzenia najbliższego ustawieniu właściwości System.Threading.Thread.CurrentThread.CurrentUICulture

Kodowania znaków .NET Framework

- .NET Framework używa UTF-16
  - w niektórych przypadkach w wewnętrznych operacjach używany jest UTF-8

- Użycie przestrzeni nazw **`System.Text`** do zmiany sposobu kodowania tekstów
  - UTF-16 – klasa **`UnicodeEncoding`**
  - UTF-8 – klasa **`UTF8Encoding`**
  - ASCII – klasa **`ASCIIEncoding`**
  - kodowania Windows/ISO – klasa **`Encoding`**

Ustawienia regionalne *(Locales, Cultures)*

- Ustawienie regionalne składa się z języka i opcjonalnie z regionu
- Jest reprezentowane przez `System.Globalization.CultureInfo` class

```
//CultureInfo cultureInfo = new CultureInfo("en");
CultureInfo cultureInfo = new CultureInfo("en-GB");

string shortDatePattern =
    cultureInfo.DateTimeFormat.ShortDatePattern;

string currencySymbol =
    cultureInfo.NumberFormat.CurrencySymbol;
```

CurrentCulture i CurrentUICulture

- **`Thread.CurrentCulture`**
  - Reprezentuje domyślne ustawienie regionalne dla wszystkich klas z przestrzeni nazw **`System.Globalization`**
  - Ma wpływ na wszystkie aspekty zależne od ustawień regionalnych (np. formatowanie daty, czasu, liczb, kwot, parsowanie, sortowanie itp.)
  - Domyślnie jest zgodne z ustawieniem funkcji Win32 **`GetUserDefaultLCID`**
- **`Thread.CurrentUICulture`**
  - Reprezentuje ustawienie regionalne używane przez metody klasy **`ResourceManager`**
  - Ma wpływ na ustawianie zasobów wykorzystywanych w interfejsie użytkownika, np. tekstów i obrazków
  - Domyślnie zgodne z funkcją **`GetUserDefaultUILanguage`**
- Dla każdego wątku właściwości **`CurrentCulture`** i **`CurrentUICulture`** muszą być ręcznie ustawione
  - Wątki nie dziedziczą tych ustawień

CultureInfo

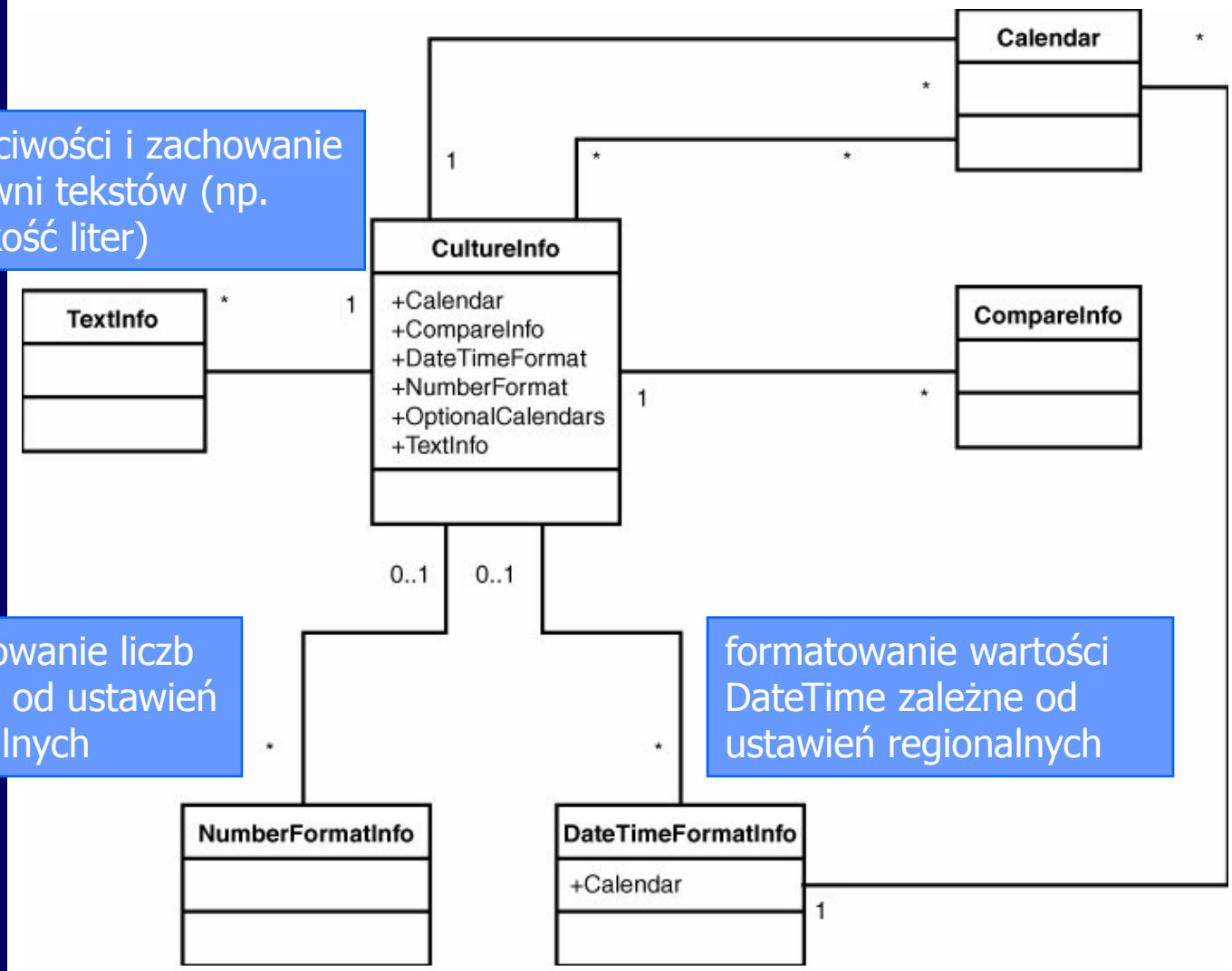
podstawowe operacje na datach

właściwości i zachowanie pisowni tekstów (np. wielkość liter)

porównywanie tekstów zależne od ustawień regionalnych

formatowanie liczb zależne od ustawień regionalnych

formatowanie wartości DateTime zależne od ustawień regionalnych



Porównywanie tekstów

- Dostępne możliwości porównywania tekstów:
  - Operator `==` lub `String.Equals`
    - niezależne od ustawień regionalnych
    - zależne od wielkości liter
  - `String.CompareTo`, `String.Compare`, `String.CompareOrdinal`
    - identyczność stwierdzana na podstawie porównania niezależnego od ustawień regionalnych
    - porządek stwierdzany na podstawie ustawień regionalnych

Zasady tworzenia graficznego interfejsu użytkownika

Interfejs użytkownika

- Interfejs użytkownika jest częścią komputera i jego oprogramowania, którą ludzie mogą widzieć, słyszeć, dotykać, mówić do niej lub w inny sposób odbierać oraz nią kierować
- Wejście – sposób informowania komputera o potrzebach użytkownika
  - klawiatura, mysz, trackball, palec na ekranie dotykowym, głos
- Wyjście – sposób przekazywania użytkownikowi wyników obliczeń oraz informowania o potrzebach
  - ekran, głos, dźwięk

Graficzny interfejs użytkownika

- Podstawowym sposobem interakcji użytkownika z komputerem jest urządzenie wskazujące
  - urządzenie mające zastąpić ludzkie ręce
- Interfejs WIMP: *windows, icons, menus, pointers*

Zalety systemów graficznych

- Symbole mogą być szybciej rozpoznawane niż tekst
- Szybsze uczenie, łatwiejsze zapamiętywanie
- Bardziej naturalne
- Wykorzystują sygnały wizualne i przestrzenne
- Zwiększają poczucie kontroli nad komputerem
- Zmniejszają obawy i nieśmiałość użytkowników
- Są przyjemniejsze w użyciu
- Symbole graficzne mogą zajmować mniej miejsca niż tekst
- Zmniejszają potrzebę użycia słów
- Zmniejszają potrzebę pisania na klawiaturze

Wady systemów graficznych

- Większa złożoność definiowania interfejsu
- Zmiany w systemie oznaczają konieczność nauki
- Brak opartych na doświadczeniu zaleceń dot. tworzenia interfejsu
- Niekonsekwencje w technologii i nazewnictwie
- Nie zawsze w pełni znany użytkownikom
- Niewielka liczba sprawdzonych symboli graficznych
- Niewydajne dla użytkowników biegle piszących na klawiaturze
- Niewydajne dla bardzo zaawansowanych użytkowników
- Strata czasu poprzez mnogość możliwości
- Ograniczenia sprzętowe

Ogólne zasady tworzenia dobrego graficznego interfejsu użytkownika

Zachowanie i wygląd kontrolek

- Użytkownik musi być w stanie przewidzieć zachowanie kontrolki na podstawie jej wyglądu
- Zachowanie wszystkich kontrolek w aplikacji musi być spójne
- Jeśli aplikacja potrzebuje nowej kontrolki, której zachowanie będzie inne niż standardowych, należy nadać jej charakterystyczny wygląd

Spójność na poziomie środowiska

- Użytkownik musi być w stanie przewidzieć zachowanie aplikacji na podstawie własnego doświadczenia z innymi aplikacji uruchamianymi w tym środowisku
  - ma to zastosowanie do: kontrolek, zachowania myszy, klawiszy skrótu, pozycji menu oraz symboli graficznych
- Należy utrzymywać zgodność zachowania aplikacji z wzorcami środowiska
  - zgodność ze środowiskiem ma pierwszeństwo przed zgodnością wersji aplikacji dla różnych systemów

Ostrzeżenia i informacje o błędach

- W dobrym interfejsie rzadko zachodzi konieczność ostrzegania użytkownika i informowania o błędach
  - wyjątki: problemy sprzętowe (np. błąd dostępu do dysku, utrata połączenia sieciowego), ostrzeżenia przed wykonaniem nieodwracalnej i potencjalnie niebezpiecznej akcji)
- Należy tak projektować interfejs, żeby uniemożliwiać wprowadzanie niepoprawnych danych lub przynajmniej sugerować ich poprawny format
  - wykorzystać możliwości kontrolek (lista, maski w polach tekstowych itp.)
- Uniemożliwić wykonanie akcji w niepoprawnej kolejności

Interakcja z użytkownikiem

- Powiadomienia o stanie aplikacji
  - użytkownik zawsze powinien wiedzieć jaki jest stan programu, zaawansowanie obliczeń, ew. znać szacunkową czasochłonność obliczeń
  - należy tak projektować interfejs, żeby początkujący użytkownik zawsze był w stanie patrząc na aplikację domyśleć się jakie operacje zostały wykonane
- Interakcja na poziomie kontrolek
  - kontrolki muszą swoim wyglądem powiadamiać o wykonanej akcji (np. wciskanie przycisku, zaznaczanie pola wyboru itp.)

Poznawanie interfejsu poprzez próby

- Dobry interfejs użytkownika zachęca do poznawania go przez samodzielne prób
  - samodzielne wykonanie zadania zwiększa zadowolenie użytkownika
- Zawsze należy dać możliwość cofnięcia i powtórnego wykonania akcji bez obawy o poprawność danych

Intuicyjność interfejsu

- Celem jest stworzenie interfejsu, który nie potrzebuje wyjaśnienia
  - dobre aplikacje mają znakomite systemy pomocy
  - bardzo dobre aplikacje rzadko zmuszają początkujących użytkowników do korzystania z pomocy

Dźwięki, kolory, animacje

- Dźwięki, kolory i animacje nie zawsze są mile widziane przez użytkowników
  - powinny być używane tylko w przypadku, gdy zwiększają szansę użytkownika na poprawne wykonanie zadania
- Należy dostosować się do standardów środowiska
  - wykorzystać standardowe kolory i dźwięki, wykorzystywać animacje w podobnych sytuacjach, jak w innych aplikacjach

Ustawienia użytkownika

- Interfejs powinien pomagać użytkownikowi w dostosowaniu aplikacji do swoich potrzeb
- Aplikacje, które mogą być używane przez wielu użytkowników lub mogą być używane na wielu komputerach powinny spełniać standardy określane przez środowisko
  - sprzętowe możliwości obrazu – wielkość ekranu, rozdzielczość, liczba kolorów
  - upośledzenia użytkownika
- Dopasowanie interfejsu do potrzeb użytkownika
  - czcionki, kolory, dźwięki, położenie i wygląd pasków narzędziowych, pozycje menu
  - bardzo pomocne są predefiniowane schematy

Zachowanie „modalne”

- Modalne zachowanie zmusza użytkownika do wykonywania zadań w określonej kolejności
  - zmniejsza elastyczność interfejsu i powoduje zaburzenia w pracy użytkownika
- Użycie kreatorów ułatwia wykonanie skomplikowanych zadań, ale zmniejsza elastyczność interfejsu

Niedostrzeganie interfejsu użytkownika

- Bardzo dobry interfejs jest rzadko dostrzegany przez użytkowników
  - uwaga użytkowników jest skupiona na zadaniu, a nie na elementach interfejsu użytkownika, których wykorzystanie jest konieczne do wykonania tego zadania

Literatura

- Wilbert O. Galitz, The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques, Second Edition, 2002
- Microsoft Windows User Experience, 1999
- Everett N. McKay, Developing user interfaces for Microsoft Windows, 1999
- Larry E. Wood, User Interface Design: Bridging the Gap from User Requirements to Design, 1997
- Leslie Cortes, Designing a Graphical User Interface, 1997