



Unicode[®] (no CD)

Objetivos

- Familiarizar-se com o Unicode.
- Discutir a missão do Consórcio Unicode.
- Discutir as bases de projeto do Unicode.
- Entender as três formas de codificação Unicode: UTF-8, UTF-16 e UTF-32.
- Apresentar caracteres e hieróglifos.
- Discutir as vantagens e desvantagens de se usar Unicode.
- Fornecer uma breve visão do *site* do Consórcio Unicode



Sumário

- K.1** **Introdução**
- K.2** **Formatos de transformação Unicode**
- K.3** **Caracteres e hieróglifos**
- K.4** **Vantagens/desvantagens do Unicode**
- K.5** **Site do Consórcio Unicode na Web**
- K.6** **Usando Unicode**
- K.7** **Intervalos de caracteres**

Resumo • Terminologia • Exercícios de auto-revisão • Respostas aos exercícios de auto-revisão • Exercícios

K.1 Introdução

O uso de *codificações* de caracteres (isto é, valores numéricos associados com caracteres) inconsistentes durante o desenvolvimento de produtos globais de *software* causa sérios problemas, pois os computadores processam as informações usando números. Por exemplo, o caractere “a” é convertido para um valor numérico para que o computador possa processar este item de dados. Muitos países e empresas desenvolveram seus próprios sistemas de codificação, que são incompatíveis com os sistemas de codificação de outros países e empresas. Por exemplo, o sistema operacional Microsoft Windows atribui o valor 0xC0 ao caractere “A com acento grave”, enquanto o sistema operacional Apple Macintosh atribui o mesmo valor a um ponto de interrogação de cabeça para baixo. Isto resulta em uma representação enganosa e possível corrupção de dados, pois o dado não é processado como deveria.

Na falta de uma codificação-padrão universal amplamente utilizada, os desenvolvedores globais de *software* precisam *localizar* seus produtos extensivamente antes de sua distribuição. A localização inclui a tradução da linguagem e da adaptação cultural do conteúdo. O processo de localização normalmente inclui modificações substanciais no código-fonte (como conversão de valores numéricos e das hipóteses básicas feitas pelos programadores), resultando em aumento de custos e atrasos na liberação do *software*. Por exemplo, alguns programadores de língua inglesa podem projetar um *software* global assumindo que um caractere isolado pode ser representado em um *byte*. Entretanto, quando estes produtos são localizados para os mercados asiáticos, as hipóteses do programador não são mais válidas, e assim a maior parte do código, ou quase todo o código, precisa ser reescrita. A localização é necessária para cada liberação de uma versão. Quando concluída a localização de um produto de *software* para um mercado particular, uma nova versão, que também necessita ser localizada, está pronta para distribuição. Por conseguinte, é enfadonho e custoso produzir e distribuir produtos globais de *software* em mercados nos quais não existe uma codificação-padrão universal de caracteres.

Em resposta a esta situação, foi criado o *Padrão Unicode*, uma codificação-padrão que facilita a produção e a distribuição de *software*. O Padrão Unicode delinea uma especificação para produzir uma codificação consistente dos *símbolos* e caracteres do mundo. Os produtos de *software* que tratam texto codificado no Padrão Unicode necessitam ser localizados, mas o processo de localização é mais simples, visto que os valores numéricos não necessitam ser convertidos e as hipóteses feitas pelos programadores a respeito da codificação de caracteres são universais. O Padrão Unicode é mantido por uma organização não-lucrativa denominada *Consórcio Unicode*, cujos membros incluem empresas como Apple, IBM, Microsoft, Oracle, Sun Microsystems, Sybase e muitas outras.

Quando o Consórcio concebeu e definiu o Padrão Unicode, desejava um sistema de codificação que fosse *universal*, *eficiente*, *uniforme* e *não-ambíguo*. Um sistema de codificação universal compreende todos os caracteres comumente usados. Um sistema de codificação eficiente permite que os arquivos-texto sejam analisados facilmente. Um sistema de codificação uniforme atribui valores fixos para todos os caracteres. Um sistema de codificação não-ambíguo representa um dado caractere de maneira consistente. Estes quatro termos são citados como a base de projeto do Padrão Unicode.

K.2 Formatos de transformação Unicode

Embora o Unicode incorpore o limitado *conjunto de caracteres* (isto é, uma coleção de caracteres) ASCII, ele compreende um conjunto de caracteres mais abrangente. Em ASCII cada caractere é representado por um *byte* contendo 0s e 1s. O *byte* é capaz de armazenar números binários de 0 a 255. A cada caractere é atribuído um número entre 0 e 255, fazendo com que os sistemas baseados em ASCII possam suportar somente 256 caracteres, uma diminuta fração dos caracteres do mundo. O Unicode estende o conjunto de caracteres ASCII, codificando a grande maioria dos caracteres do mundo. O Padrão Unicode codifica todos estes caracteres em um espaço numérico uniforme de 0 a 10FFFF hexadecimal. Uma implementação expressará estes números em um dos diversos formatos de transformação, escolhendo aquele que melhor se ajuste a uma aplicação particular.

Três destes formatos estão em uso, denominados *UTF-8*, *UTF-16* e *UTF-32*, dependendo do tamanho das unidades – em *bits* – que estão sendo usadas. O UTF-8, uma forma de codificação com largura variável, exige de um a quatro bytes para expressar cada caractere Unicode. Os dados em UTF-8 consistem em *bytes* de 8 *bits* (seqüências de um, dois, três ou quatro *bytes*, dependendo dos caracteres que estão sendo codificados) e são bem adequados para os sistemas baseados em ASCII, quando existe uma predominância de caracteres de um *byte* (o ASCII representa caracteres em um *byte*). Atualmente, o UTF-8 é amplamente utilizado em sistemas UNIX e bancos de dados.

A forma de codificação com largura variável UTF-16 expressa caracteres Unicode em unidades de 16 *bits* (isto é, como dois *bytes* adjacentes, ou um inteiro curto em muitas máquinas). A maioria dos caracteres Unicode são expressos em uma única unidade de 16 *bits*. Entretanto, os caracteres com valores acima de FFFF hexadecimal são expressos com um par ordenado de unidades de 16 *bits*, denominados *substitutos*. Os substitutos são inteiros de 16 *bits* no intervalo de D800 até DFFF, que são usados somente com o objetivo de “escapar” para caracteres de numeração mais alta. Aproximadamente um milhão de caracteres podem ser expressos desta maneira. Embora um par substituto exija 32 *bits* para representar caracteres, é eficiente em termos de espaço usar estas unidades de 16 *bits*. Os substitutos são caracteres raros nas implementações atuais. Muitas implementações de tratamento de *strings* são escritas em termos de UTF-16. [Nota: detalhes e exemplos de código para tratamento de UTF-16 são disponibilizados no *site* do Consórcio no endereço www.unicode.org.]

As implementações que exigem uso significativo de caracteres raros ou *scripts* inteiramente codificados acima de FFFF hexadecimal devem usar o UTF-32, uma forma de codificação com largura fixa de 32 *bits* que usualmente exige o dobro de memória do que os caracteres codificados como UTF-16. A maior vantagem da forma de codificação de largura fixa UTF-32 é que ela expressa uniformemente todos os caracteres, e assim se torna fácil processar *arrays* de caracteres.

Existem algumas poucas diretrizes que determinam quando usar uma forma de codificação particular. A melhor forma de codificação a ser usada depende dos protocolos de sistemas de computadores e empresas, e não dos dados em si. Geralmente, a forma de codificação UTF-8 deveria ser usada quando os protocolos de sistemas de computadores e companhias exigem que os dados sejam tratados em unidades de 8 *bits*, particularmente em sistemas legados que estão sendo atualizados, pois ela freqüentemente simplifica as mudanças nos programas existentes. Por esta razão, o UTF-8 se tornou a forma de codificação preferida na Internet. Do mesmo modo, o UTF-16 é a forma de codificação preferida nas aplicações Microsoft Windows. O UTF-32 provavelmente tornar-se-á a mais usada no futuro, à medida que mais caracteres sejam codificados com valores acima de FFFF hexadecimal. Além disso, o UTF-32 exige um tratamento menos sofisticado que o do UTF-16 quando existem pares substitutos.

A Fig. K.1 mostra as diferentes maneiras nas quais as três formas de codificação tratam a codificação de caracteres.

K.3 Caracteres e hieróglifos

O Padrão Unicode consiste em *caracteres*, componentes escritos (isto é, alfabetos, números, sinais de pontuação, acentos, etc.) que podem ser representados por valores numéricos. Entre os exemplos de caracteres incluem-se: U+0041 LETRA MAIÚSCULA LATINA A. Na primeira representação de caractere, U+yyyy é um *valor de código*, no qual U+ se refere a valores de código Unicode, para se diferenciar de outros valores hexadecimais. O yyyy representa um número hexadecimal de quatro dígitos de um caractere codificado. Os valores de códigos são combinações de *bits* que representam caracteres codificados. Os caracteres são representados com *hieróglifos*, diversas formas, fontes e tamanhos para exibir caracteres. Não existem valores de código para hieróglifos no Padrão Unicode. Na Fig. K.2, há exemplos de hieróglifos.

O Padrão Unicode abrange alfabetos, ideogramas, silabários, sinais de pontuação, *diacríticos*, operadores matemáticos, etc. que fazem parte das linguagens e *scripts* usados no mundo. O diacrítico é uma marca especial adicionada a um caractere para diferenciá-lo de outra letra ou para indicar uma pronúncia (por exemplo, em Espanhol o til “~” sobre o caractere “n”). Atualmente, o Unicode fornece 94.140 valores de códigos para representação de caracteres, com mais de 880.000 valores de códigos reservados para expansão futura.

Caracteres	UTF-8	UTF-16	UTF-32
LETRA LATINA MAIÚSCULA A	0x41	0x0041	0x00000041
LETRA GREGA MAIÚSCULA ALPHA	0xCD 0x91	0x0391	0x00000391
CJK IDEOGRAMA UNIFICADO – 4E95	0xE4 0xBA 0x95	0x4E95	0x00004E95
ANTIGA LETRA EM ITÁLICO A	0xF0 0x80 0x83 0x80	0xDC00 0xDF00	0x00010300

Fig. K.1 Correlação entre as três formas de codificação.



Fig. K.2 Vários hieróglifos do caractere A.

K.4 Vantagens/desvantagens do Unicode

O Padrão Unicode possui muitas vantagens significativas que favorecem seu uso. Uma é o seu impacto no desempenho da economia internacional. O Unicode padroniza os caracteres para os sistemas mundiais de escrita em um modelo uniforme que incentiva a transferência e o compartilhamento de dados. Os programas desenvolvidos com tal esquema mantêm sua precisão pois cada caractere possui uma definição única (isto é, *a* é sempre U+0061, *%* é sempre U+0025). Isto permite que as empresas gerenciem as altas demandas dos mercados internacionais através do processamento de diferentes sistemas de escrita ao mesmo tempo. Além disso, todos os caracteres podem ser gerenciados de maneira idêntica, evitando, assim, qualquer confusão causada por diferentes arquiteturas do código de caracteres. Além do mais, gerenciar dados de uma maneira consistente elimina a corrupção dos dados, pois eles podem ser ordenados, pesquisados e tratados com um processo consistente.

Outra vantagem do Padrão Unicode é a *portabilidade* (isto é, *software* que pode ser executado em computadores incompatíveis ou com sistemas operacionais incompatíveis). A maioria dos sistemas operacionais, bancos de dados, linguagens de programação e navegadores da Web atualmente suportam ou planejam suportar o Unicode.

Uma desvantagem do Padrão Unicode é a quantidade de memória exigida pelo UTF-16 e pelo UTF-32. Os conjuntos de caracteres ASCII possuem comprimento de 8 *bits*, de modo que exigem menos memória que o conjunto *default* de caracteres Unicode de 16 *bits*. Entretanto, o *double-byte character set* (DBCS) e o *multi-byte character set* (MBCS), que codificam os caracteres asiáticos (ideogramas) exigem 2 a 4 *bytes*, respectivamente. Em tais casos, as formas de codificação UTF-16 ou UTF-32 podem ser usadas com poucas restrições quanto à memória ou ao desempenho.

Outra desvantagem de Unicode é que embora, ele inclua mais caracteres que qualquer outro conjunto de caracteres em uso, ele ainda não permite codificar todos os caracteres de escrita do mundo.

Outra desvantagem do Padrão Unicode é que o UTF-8 e o UTF-16 são formas de codificação com largura variável, de modo que os caracteres ocupam quantidades diferentes de memória.

K.5 Site do Consórcio Unicode na Web

Se você quiser aprender mais sobre o Padrão Unicode, visite o endereço www.unicode.org. Este *site* fornece uma profusão de informações sobre o Padrão Unicode que dá uma compreensão clara sobre o Unicode. Atualmente, a página principal é organizada em várias seções – *New to Unicode*, *General Information*, *The Consortium*, *The Unicode Standard*, *Work in Progress* e *For Members*.

A seção *New to Unicode* consiste em duas subseções: **What is Unicode** e **How to Use this Site**. A primeira subseção fornece uma introdução técnica ao Unicode, descrevendo seus principais princípios, interpretações e atribuições de caracteres, processamento de texto e conformidade com Unicode. A leitura desta subseção é recomendada a todos os novatos em Unicode. Além disso, esta seção fornece uma lista de *links* relacionados ao Unicode, que fornecem ao leitor informações adicionais. A subseção **How to Use this Site** contém informações sobre como usar e como navegar neste *site*, bem como *hyperlinks* para recursos adicionais.

A seção *General Information* contém seis subseções: **Where is my Character**, **Display Problems**, **Useful Resources**, **Enabled Products**, **Mail Lists** e **Conferences**. As principais áreas cobertas nesta seção incluem um *link* para as tabelas de código Unicode (uma lista completa de valores de códigos) reunidas pelo Consórcio Unicode, além de um guia detalhado sobre como localizar um caractere codificado na tabela de código. Além disso, a seção contém conselhos sobre como configurar diferentes sistemas operacionais e navegadores da Web de forma que os caracteres Unicode possam ser visualizados adequadamente. Além do mais, a partir desta seção, o usuário pode navegar para outros *sites* que fornecem informações sobre vários tópicos, como fontes, padrões linguísticos e outros padrões como o *Armenian Standards Page* e o *Chinese GB 18030 Encoding Standard*.

A seção *Consortium* consiste em cinco subseções: **Who we are**, **Our Members**, **How to Join**, **Press Info** e **Contact Us**. Esta seção fornece a lista dos atuais participantes do Consórcio Unicode, bem como informações para se tornar um sócio. Os privilégios para cada tipo de sócio – *full*, *associate*, *specialist* e *individual* – e as taxas estabelecidas para cada membro estão relacionadas aqui.

A seção *The Unicode Standard* consiste em nove subseções: **Start Here**, **Latest Version**, **Technical Reports**, **Code Charts**, **Unicode Data**, **Update & Errata**, **Unicode Policies**, **Glossary** e **Technical FAQ**. Esta seção descreve as atualizações aplicadas à última versão do Padrão Unicode e categoriza todas as codificações definidas. O usuário pode aprender como a última versão foi modificada para abranger mais características e recursos. Por exemplo, uma melhoria da Versão 3.1 é que ela contém codificação de caracteres adicionais. Além disso, se os usuários não estão familiarizados com os termos do vocabulário usado pelo Consórcio Unicode, eles podem navegar pela subseção **Glossary**.

A seção *Work in Progress* consiste em três subseções: **Calendar of Meetings**, **Proposed Characters** e **Submitting Proposals**. Esta seção apresenta ao usuário um catálogo de caracteres recentemente incluídos no esquema Padrão Unicode e aqueles caracteres que estão sendo considerados para inclusão. Se os usuários concluem que um caractere foi esquecido, eles podem enviar uma proposta escrita para a inclusão de tal caractere. A subseção **Submitting Proposals** contém diretrizes precisas sobre como proceder ao enviar propostas escritas.

A seção *For Members* consiste em duas subseções: **Member Resources** e **Working Documents**. Estas subseções são protegidas por senha para que os somente membros do consórcio possam acessar os *links*.

K.6 Usando Unicode

Várias linguagens de programação (por exemplo, C, Java, JavaScript, Perl, Visual Basic, etc.) fornecem algum nível de suporte ao Padrão Unicode. A Fig. K.3 mostra um programa Java que imprime o texto “Welcome to Unicode!” em oito línguas diferentes: inglês, russo, francês, alemão, japonês, português, espanhol e chinês tradicional. [Nota: o *site* do Consórcio Unicode na Web contém um *link* para tabelas de código que listam os valores de código do Unicode 16 bits].

```

1 // Fig. K.3: Unicode.java
2 // Demonstrando como usar Unicode em programas Java.
3
4 // Pacotes do núcleo de Java

```

Fig. K.3 Programa Java que usa a codificação Unicode (parte 1 de 3).

```

5  import java.awt.*;
6
7  // Pacotes de extensão de Java
8  import javax.swing.*;
9
10 public class Unicode extends JFrame {
11     private JLabel english, chinese, cyrillic, french, german,
12         japanese, portuguese, spanish;
13
14     // construtor Unicode
15     public Unicode()
16     {
17         super( "Demonstrating Unicode" );
18
19         // obtém painel de conteúdo e configura seu leiaute
20         Container container = getContentPane();
21         container.setLayout( new GridLayout( 8, 1 ) );
22
23         // construtor JLabel com um string como argumento
24         english = new JLabel( "\u0057\u0065\u0063\u0066" +
25             "\u0064\u0065\u0020\u0074\u0066\u0020Unicode\u0021" );
26         english.setToolTipText( "This is English" );
27         container.add( english );
28
29         chinese = new JLabel( "\u6B22\u8FCE\u4F7F\u7528\u0020" +
30             "\u0020Unicode\u0021" );
31         chinese.setToolTipText( "This is Traditional Chinese" );
32         container.add( chinese );
33
34         cyrillic = new JLabel( "\u0414\u043E\u0431\u0440\u043E" +
35             "\u0020\u043F\u043E\u0436\u0430\u043B\u043E\u0432" +
36             "\u0430\u0442\u044A\u0020\u0432\u0020Unicode\u0021" );
37         cyrillic.setToolTipText( "This is Russian" );
38         container.add( cyrillic );
39
40         french = new JLabel( "\u0042\u0069\u0065\u006E\u0076" +
41             "\u0065\u006E\u0075\u0065\u0020\u0061\u0075\u0020" +
42             "Unicode\u0021" );
43         french.setToolTipText( "This is French" );
44         container.add( french );
45
46         german = new JLabel( "\u0057\u0069\u0063\u006B\u0066" +
47             "\u0064\u0065\u0065\u0020\u007A\u0075\u0020" +
48             "Unicode\u0021" );
49         german.setToolTipText( "This is German" );
50         container.add( german );
51
52         japanese = new JLabel( "Unicode\u3078\u3087\u3045\u3053" +
53             "\u305D\u0021" );
54         japanese.setToolTipText( "This is Japanese" );
55         container.add( hiragana );
56
57         portuguese = new JLabel( "\u0053\u00E9\u006A\u0061\u0020" +
58             "\u0042\u0065\u0064\u0076\u0069\u006E\u0064" +
59             "\u0066\u0020Unicode\u0021" );
60         portuguese.setToolTipText( "This is Portuguese" );
61         container.add( portuguese );
62
63         spanish = new JLabel( "\u0042\u0069\u0065\u006E\u0076" +
64             "\u0065\u006E\u0069\u0064\u0061\u0020\u0061\u0020" +

```

Fig. K.3 Programa Java que usa a codificação Unicode (parte 2 de 3).

```

65         "Unicode\u0021" );
66         spanish.setToolTipText( "This is Spanish" );
67         container.add( spanish );
68
69     } // fim do construtor Unicode
70
71     // executa o aplicativo
72     public static void main( String args[] )
73     {
74         Unicode application = new Unicode();
75         application.setDefaultCloseOperation(
76             JFrame.EXIT_ON_CLOSE );
77         application.pack();
78         application.setVisible( true );
79
80     } // fim do método main
81
82 } // fim da classe Unicode

```

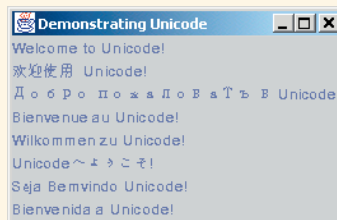


Fig. K.3 Programa Java que usa a codificação Unicode (parte 3 de 3).

O programa `Unicode.java` usa *seqüências de escape* para representar caracteres. Uma seqüência de escape usa a forma `\uxxxx`, onde `xxxx` representa o valor do código em quatro dígitos hexadecimais. As linhas 24 e 25 contêm as séries de seqüências de escape necessárias para imprimir “Welcome to Unicode!” em inglês. A primeira seqüência de escape (`\u0057`) equiivale ao caractere “W”, a segunda seqüência de escape (`\u0065`) equiivale ao caractere “e”, e assim por diante. A seqüência de escape `\u0020` (linha 25) é a codificação para o caractere *espaço*. As seqüências de escape `\u0074` e `\u0066` equivalem à palavra “to”. Observe que “Unicode” não é codificado pois é uma marca registrada e não possui equivalente em muitas linguagens. A linha 25 também contém a seqüência de escape `\u0021` para o ponto de exclamação (!).

As linhas 29 a 65 contêm a seqüência de escape para as outras sete línguas. Os caracteres para inglês, francês, alemão, português e espanhol estão localizados no bloco **Basic Latin**, os caracteres japoneses estão localizados no bloco **Hiragana**, os caracteres russos estão localizados no bloco **Cyrillic** e os caracteres chineses tradicionais estão localizados no bloco **CJK Unified Ideographs**.

[Nota: para exibir adequadamente a saída do `Unicode.java`, copie o arquivo `font.properties.zh` para os arquivos `font.properties` (localizados nos diretórios `C:\ProgramFiles\JavaSoft\JRE\1.3.1\lib` e `C:\jdk1.3.1\jre\lib`). Salve o conteúdo de `font.properties` antes de sobrescrevê-lo com o conteúdo de `font.properties.zh`.

K.7 Intervalos de caracteres

O Padrão Unicode atribui valores de códigos, no intervalo de 0000 (**Basic Latin**) a E007F (**Tags**), aos caracteres escritos do mundo. Atualmente, existem valores de códigos para 94.140 caracteres. Para simplificar a pesquisa de um caractere e seu valor de código associado, o Padrão Unicode geralmente agrupa os valores de códigos por *script* e função (isto é, os caracteres latinos são agrupados em um bloco, os operadores matemáticos são agrupados em outro bloco, etc.). Como regra, o *script* é um único sistema de escrita que é usado para várias línguas (por exem-

plo, o *script* Latin é usado para inglês, francês, espanhol, etc.). A página **Code Charts** no *site* do Consórcio Unicode relaciona todos os blocos definidos e seus respectivos valores de códigos. A Fig. K.4 relaciona alguns blocos (*scripts*) do *site* e seus intervalos de valores de códigos.

Script	Intervalo de valores de códigos
Arabic	U+0600 a U+06FF
Basic Latin	U+0000 a U+007F
Bengali (Índia)	U+0980 a U+09FF
Cherokee (América)	U+13A0 a U+13FF
CJK Unified Ideographs (Leste Asiático)	U+4E00 a U+9FAF
Cyrillic (Rússia e Leste Europeu)	U+0400 a U+04FF
Ethiopic	U+1200 a U+137F
Greek	U+0370 a U+03FF
Hangul Jamo (Coreia)	U+1100 a U+11FF
Hebrew	U+0590 a U+05FF
Hiragana (Japão)	U+3040 a U+309F
Khmer (Camboja)	U+1780 a U+17FF
Lao (Laos)	U+0E80 a U+0EFF
Mongolian	U+1800 a U+18AF
Myanmar	U+1000 a U+109F
Ogham (Irlanda)	U+1680 a U+169F
Runic (Alemanha e Escandinávia)	U+16A0 a U+16FF
Sinhala (Sri Lanka)	U+0D80 a U+0DFF
Telugu (Índia)	U+0C00 a U+0C7F
Thai	U+0E00 a U+0E7F

Fig. K.4 Alguns intervalos de caracteres.

Resumo

- Antes do Unicode, os desenvolvedores de *software* eram incomodados pelos uso inconsistente da codificação de caracteres (isto é, valores numéricos para caracteres). A maioria dos países e organizações tinham seus próprios sistemas de codificação, os quais eram incompatíveis. Um bom exemplo de sistemas de codificações individuais eram as plataformas Windows e Macintosh.
- Os computadores processam os dados através da conversão de caracteres para valores numéricos. Por exemplo, o caractere “a” é convertido para um valor numérico de modo que um computador possa processar este dado.
- A localização global de *software* exige modificações substanciais no código-fonte, resultando em aumento de custos e atrasos na liberação de produtos.
- A localização é necessária a cada liberação de uma versão. Quando concluída a localização de um produto de *software* para um mercado particular, uma nova versão, que também necessita ser localizada, está pronta para distribuição. Por conseguinte, é enfadonho e custoso produzir e distribuir produtos globais de *software* em mercados nos quais não existe uma codificação-padrão universal de caracteres.
- O Consórcio Unicode desenvolveu o Padrão Unicode em resposta a sérios problemas criados por várias codificações de caracteres e o uso destas codificações.
- O Padrão Unicode facilita a produção e a distribuição de *software* localizado. Ele define uma especificação para a codificação consistente de caracteres e símbolos do mundo.

- Os produtos de *software* que tratam texto codificado no Padrão Unicode necessitam ser localizados, mas o processo de localização é mais simples e mais eficiente, visto que os valores numéricos não necessitam ser convertidos.
- O Padrão Unicode foi projetado para ser universal, eficiente, uniforme e não-ambíguo.
- Um sistema de codificação universal compreende todos os caracteres comumente usados. Um sistema de codificação eficiente permite que os arquivos-texto sejam analisados facilmente. Um sistema de codificação uniforme atribui valores fixos para todos os caracteres. Um sistema de codificação não-ambíguo representa um dado caractere de maneira consistente.
- O Unicode estende o conjunto limitado de caracteres ASCII para incluir todos os principais caracteres do mundo.
- O Unicode usa três formatos de transformação Unicode: UTF-8, UTF-16 e UTF-32, e cada um pode ser apropriado para uso em diferentes contextos.
- Os dados em UTF-8 consistem em *bytes* de 8 *bits* (seqüências de um, dois, três ou quatro *bytes*, dependendo dos caracteres que estão sendo codificados) e é bem adequado para os sistemas baseados em ASCII quando existe uma predominância de caracteres de um *byte* (ASCII representa caracteres em um *byte*).
- O UTF-8 é uma forma de codificação com largura variável que é mais compacta para texto que envolve a maioria dos caracteres latinos e pontuação ASCII.
- O UTF-16 é a forma *default* de codificação do Padrão Unicode. É uma forma de codificação com largura variável que usa unidades de código de 16 *bits* em vez de *bytes*. A maioria dos caracteres são representados por uma única unidade de 16 *bits*, mas alguns caracteres exigem pares substitutos.
- Sem pares substitutos, a codificação UTF-16 pode somente abranger 65.000 caracteres, mas com pares substitutos isto se expande para incluir mais de um milhão de caracteres.
- O UTF-32 é uma forma de codificação em 32 *bits*. A principal vantagem da forma de codificação com largura fixa é que ela expressa uniformemente todos os caracteres, de modo que se torna fácil processá-los em *arrays* e assim por diante.
- O Padrão Unicode consiste em caracteres. O caractere é qualquer componente escrito que pode ser representado por um valor numérico.
- Os caracteres são representados com hieróglifos, que são diversas formas, fontes e tamanhos para exibir caracteres.
- Os valores de códigos são combinações de *bits* que representam caracteres codificados. A notação de Unicode para um valor de código é U+yyyy, no qual U+ se refere a valores do código Unicode, para se diferenciar de outros valores hexadecimais. O yyyy representa um número hexadecimal de quatro dígitos.
- Atualmente, o Unicode fornece valores de códigos para representação de 94.140 caracteres.
- Uma vantagem do Padrão Unicode é seu impacto no desempenho geral da economia mundial. As aplicações que atendem a um padrão de codificação podem ser facilmente processadas por computadores.
- Outra vantagem do Padrão Unicode é sua portabilidade. As aplicações escritas em Unicode podem ser facilmente transferidas para diferentes sistemas operacionais, bancos de dados, navegadores da Web, etc. Muitas empresas atualmente suportam ou planejam suportar o Unicode.
- Para obter mais informações sobre o Padrão Unicode e o Consórcio Unicode, visite o endereço www.unicode.org. Ele contém um *link* para as tabelas de códigos, que contêm os valores de códigos de 16 *bits* de caracteres atualmente codificados.
- Várias linguagens de programação fornecem algum nível de suporte ao Padrão Unicode.
- Nos programas Java, a seqüência de escape `\uyyyy` representa um caractere, onde yyyy é um valor de código em quatro dígitos hexadecimais. A seqüência de escape `\u0020` é a codificação universal para o caractere *espaço*.

Terminologia

ASCII

base de projeto do Unicode

bloco

caractere

codificar

conjunto de caracteres

Consórcio Unicode

diacrítico

double-byte character set (DBCS)

eficiente (base de projeto do Unicode)

hieróglifo

localização

multi-byte character set (MBCS)

não-ambíguo (base de projeto do Unicode)

notação hexadecimal

Padrão Unicode

portabilidade

script

seqüência de escape

seqüência de escape `\uyyyy`

símbolo

substituto

uniforme (base de projeto do Unicode)
universal (base de projeto do Unicode)
UTF-16

UTF-32
UTF-8
valor de código

Exercícios de auto-revisão

- K.1** Preencha as lacunas em cada uma das seguintes frases:
- Os desenvolvedores globais de *software* precisam _____ seus produtos para um mercado específico antes da distribuição.
 - O Padrão Unicode é um padrão _____ que facilita a produção e distribuição uniformes de produtos de *software*.
 - As quatro bases de projeto que constituem o Padrão Unicode são: _____, _____, _____ e _____.
 - O _____ é o menor componente de escrita que pode ser representado com um valor numérico.
 - Os *software* que pode ser executado em diferentes sistemas operacionais é conhecido como _____.
- K.2** Diga se cada uma das afirmativas abaixo é *verdadeira* ou *falsa*. Se for *falsa*, explique por quê.
- O Padrão Unicode abrange todos os caracteres do mundo.
 - Um valor de código Unicode é representado por U+yyyy, onde yyyy representa um número em notação binária.
 - O diacrítico é um caractere com uma marca especial que enfatiza uma pronúncia.
 - O Unicode é portátil.
 - Ao se projetar os programas em Java, a sequência de escape é indicada por `/uyyyy`.

Respostas aos exercícios de auto-revisão

- K.1** a) localizar. b) de codificação. c) universal, eficiente, uniforme, não-ambíguo. d) caractere. e) portátil.
- K.2** a) Falsa. Ele compreende a maioria dos caracteres mundiais. b) Falsa. O yyyy representa um número hexadecimal. c) Falsa. O diacrítico é uma marca especial adicionada a um caractere para diferenciá-lo de outra letra ou para indicar uma pronúncia. d) Verdadeira. e) Falsa. A sequência de escape é indicada por `\uyyyy`.

Exercícios

- K.3** Navegue no *site* do Consórcio Unicode na Web (www.unicode.org) e anote os valores de código hexadecimal para os caracteres a seguir. Em que bloco eles estão localizados?
- Letra latina “Z”.
 - Letra latina “n” com o “til” (~).
 - Letra grega delta.
 - Operador matemático “menor que ou igual a”.
 - Sinal de pontuação “abre aspas” (“).
- K.4** Descreva as bases de projeto do Padrão Unicode.
- K.5** Defina os seguintes termos:
- valor de código.
 - substituto.
 - Padrão Unicode.
- K.6** Defina os seguintes termos:
- UTF-8.
 - UTF-16.
 - UTF-32.
- K.7** Descreva um cenário em que é ótimo armazenar seus dados no formato UTF-16.
- K.8** Usando os valores de códigos do Padrão Unicode, escreva um programa Java que imprime seu nome e sobrenome. O programa deve imprimir o nome em letras maiúsculas e em letras minúsculas. Se você conhece outras línguas, imprima também seu nome e sobrenome nestas línguas.