**(Opcional) Estudo de caso de GUIs e imagens gráficas: cores e formas**

**Preenchidas :** Embora você possa criar muitos designs interessantes apenas com linhas e formas básicas, a classe Graphics fornece várias ou-

tras capacidades. Os próximos dois recursos que introduzimos são cores e formas preenchidas. Acrescentar cor enriquece os desenhos que um usuário vê na tela do computador. As formas podem ser preenchidas com cores sólidas.

**public :** É o menos restritivo de todos.

**private :**  É o mais restritivo.

**protected :** Atributos e métodos definidos como protected são acessíveis pelos métodos da própria classe e pelas classes derivadas.

Dois editores amplamente utilizados nos sistemas Linux são vi e emacs. O Windows fornece o Bloco de Notas. Já o OS X for-

nece o TextEdit. Também há muitos editores freeware e shareware disponíveis on-line, incluindo Notepad++ (notepad-plus-plus.

org), EditPlus (www.editplus.com), TextPad (www.textpad.com) e jEdit (www.jedit.org).

Ambientes de desenvolvimento integrado (IDEs) fornecem ferramentas que suportam o processo de desenvolvimento de soft-

ware, como editores e depuradores para localizar erros lógicos (que fazem programas serem executados incorretamente) e outros.

Há muitos IDEs Java populares, incluindo:

• Eclipse (www.eclipse.org)

• NetBeans (www.netbeans.org)

• IntelliJ IDEA (www.jetbrains.com)

No site dos autores (Seção Antes de começar, nas páginas iniciais do livro) estão os vídeos Dive Into®, que mostram como execu-

tar os aplicativos Java desta obra e como desenvolver novos aplicativos Java com o Eclipse, NetBeans e IntelliJ IDEA.

Fase 2: compilando um programa Java em bytecodes

Na Fase 2, utilize o comando javac (o compilador Java) para compilar um programa (Figura 1.7). Por exemplo, a fim de

compilar um programa chamado Welcome.java, você digitaria

javac Welcome.java

na janela de comando do seu sistema (isto é, o Prompt do MS-DOS, no Windows, ou o aplicativo Terminal, no Mac OS X) ou em um

shell Linux (também chamado Terminal em algumas versões do Linux). Se o programa compilar, o compilador produz um arquivo

.class chamado Welcome.class que contém a versão compilada. IDEs tipicamente fornecem um item de menu, como Build ou

Make, que chama o comando javac para você. Se o compilador detectar erros, você precisa voltar para a Fase 1 e corrigi-los. No

Capítulo 2, discutiremos com detalhes os tipos de erro que o compilador pode detectar.

O compilador Java converte o código-fonte Java em bytecodes que representam as tarefas a serem executadas na fase de exe-

cução (Fase 5). O Java Virtual Machine (JVM) — uma parte do JDK e a base da plataforma Java — executa bytecodes. A máquina

virtual (virtual machine — VM) é um aplicativo de software que simula um computador, mas oculta o sistema operacional e o

hardware subjacentes dos programas que interagem com ela. Se a mesma máquina virtual é implementada em muitas plataformas

de computador, os aplicativos escritos para ela podem ser utilizados em todas essas plataformas. A JVM é uma das máquinas virtuais

mais utilizadas. O .NET da Microsoft utiliza uma arquitetura de máquina virtual semelhante.

Diferentemente das instruções em linguagem de máquina, que são dependentes de plataforma (isto é, de hardware específico

de computador), instruções bytecode são independentes de plataforma. Portanto, os bytecodes do Java são portáveis — sem re-

compilar o código-fonte, as mesmas instruções em bytecodes podem ser executadas em qualquer plataforma contendo uma JVM que

entende a versão do Java na qual os bytecodes foram compilados. A JVM é invocada pelo comando java. Por exemplo, para executar

um aplicativo Java chamado Welcome, você digitaria

**Exercício de Array**

package Arrays;

public class Main {

public static void main(String[] args) {

int [] arrInt = {1,2,3} ;

int [] arrInt2 = new int [3];

int [] arrInt3 = new int[]{1,2,3};

int [][] dias = new int[3][] ;

dias [0] = new int [2];

dias[1] = new int[]{1,2,3} ;

dias[2] = new int [4];

for (int [] arr : dias) {

for (int sum : arr) {

System.out.println(sum);

}

}

}

}