1 **import** javax.swing.\*;

2 **import** java.text.DecimalFormat;

3

4 **public** **class** FibonacciDynamic

5 {

6 **static** **long** invocations = 0;

7 **static** **long**[] data = **new** long[101];

8

9 **public** **static** **void** main(String[] args)

10 {

11 DecimalFormat f = **new** DecimalFormat("#,###");

12

13 String s = JOptionPane.showInputDialog("Enter the term number," +

14 " n, to be evaluated:" +

15 " 1<= n <=100");

16 **int** n = Integer.parseInt(s);

17

18 System.out.println("fn = " + f.format(fib(n)) +

19 "\ncalculated making " + f.format(invocations) + 20 " invocations");

21 invocations = 0;

22 System.out.println();

23 System.out.println("fn = " + f.format(fibDynamic(n)) +

24 "\ncalculated making " + f.format(invocations) + 25 " invocations");

26 }

27

28 **public** **static** **long** fib(**int** n)

29 {

30 **long** rp1, rp2;

31 invocations++;

32

33 **if**(n == 1 || n == 2) **//defined base cases**

34 {

35 **return** 1;

36 }

37 **else //general solution**

38 { rp1 = fib(n-1); **//calculate first reduced problem**

39 rp2 = fib(n-2); **//calculate second reduced problem**

40 return rp1 + rp2;

41 }

42 }

43

44 **public** **static** **long** fibDynamic(**int** n)

45 {

46 **long** rp1 = 0;

47 **long** rp2, gs;

49 invocations++;

50

51 **if**(n == 1 || n == 2) **//defined base cases**

52 {

53 data[n] = 1;

54 **return** 1;

55 }

56 **else** **if**(data[n] != 0) **//dynamic programming base case**

57 {

58 **return** data[n];

59 }

60 **else //general solution**

61 {

62 **if**(data[n-1] == 0) **//calculate f1rst reduced problem**

63 {

64 data[n-1] = fibDynamic(n-1);

65 }

66 rp1 = data[n-1];

67 **if**(data[n-2] == 0) **//calculate second reduced problem**

68 {

69 data[n-2] = fibDynamic(n-2);

70 }

71 rp2 = data[n-2];

72 gs = rp1 + rp2;

73 **return** gs;

74 }

75 }

76 }

**Figure 9.20 The application FibonacciDynamic.**