**4**

**3**

public class Encoder

{

class LiteralEncoder

{

class Encoder2

{

...

}

};

class LenEncoder

{

...

};

class LenPriceTableEncoder extends LenEncoder

{

...

};

class Optimal

{

...

};

...

int GetOptimum(int position) throws IOException

{

...

while (true)

{

...

if (newLen >= startLen){

...

for (int lenTest = startLen; ; lenTest++)

{

...

if (lenTest == \_matchDistances[offs])

{

if (lenTest < numAvailableBytesFull)

{

...

if (lenTest2 >= 2)

{

...

if (curAndLenPrice < optimum.Price)

{

...

}

}

}

...

}

}

}

}

}

}

**2**

**1**

**7**

**6**

**5**

**4**

**3**

**2**

**1**

**Nested Conditional logic**

**Inner Classes**