**public** **class** Graph {

**final** **int** V;

**int** E;

Bag<Integer>[] adj;

**public** Graph(**int** V) {

**this**.V = V;

**this**.E = 0;

adj = (Bag<Integer>[]) **new** Bag[V];

**for** (**int** v = 0; v < V; v++) {

adj[v] = **new** Bag<Integer>();

}

}

**public** **int** V() { **return** V; }

**public** **int** E() { **return** E; }

/\*\* Adds the undirected edge v-w to this graph. \*/

**public** **void** addEdge(**int** v, **int** w) {

E++;

adj[v].add(w);

adj[w].add(v);

}

/\*\* Returns the vertices adjacent to vertex <tt>v</tt>. \*/

**public** Iterable<Integer> adj(**int** v) {

**return** adj[v];

}

/\*\* Returns the degree of vertex <tt>v</tt>. \*/

**public** **int** degree(**int** v) {

**return** adj[v].size();

}

/\*\* Returns a string representation of this graph. \*/

**public** String toString() {

StringBuilder s = **new** StringBuilder();

s.append(V + " vertices, " + E + " edges " + "\n");

**for** (**int** v = 0; v < V; v++) {

s.append(v + ": ");

**for** (**int** w : adj[v]) {

s.append(w + " ");

}

s.append("\n");

}

**return** s.toString();

}

}