Для разработки программы, имитирующей поведение коллекции HashSet, необходимо создать класс HashSet, который будет содержать методы add, toString и get, а также массив для хранения элементов.

Начнем с объявления класса и массива для хранения элементов:

```

public class HashSet {

private Integer[] elements = new Integer[10];

```

Массив установлен на 10 элементов, но его можно изменить при необходимости. Далее необходимо написать метод add, который будет добавлять элемент в массив:

```

public void add(Integer element) {

for (int i = 0; i &lt; elements.length; i++) {

if (elements[i] == null) {

elements[i] = element;

break;

}

}

}

```

Метод принимает элемент типа Integer и проходит по всем элементам массива. Если элемент равен null, то он заменяется на переданный элемент. Если элементов больше нет, то новый элемент не добавляется.

Теперь напишем метод toString, который будет возвращать строку, содержащую все элементы множества:

```

public String toString() {

StringBuilder result = new StringBuilder();

for (int i = 0; i &lt; elements.length; i++) {

if (elements[i] != null) {

result.append(elements[i]).append(",");

}

}

if (result.length() &gt; 0) {

result.setLength(result.length() - 1);

}

return "{" + result.toString() + "}";

}

```

Метод создает объект StringBuilder и проходится по всем элементам массива. Если элемент не равен null, то он добавляется в StringBuilder. Затем последняя запятая удаляется и возвращается строка, содержащая все элементы множества.

Наконец, напишем метод get, который будет возвращать элемент по индексу:

```

public Integer get(int index) {

if (index &gt;= 0 &amp;&amp; index &lt; elements.length) {

return elements[index];

}

return null;

}

```

Метод принимает индекс элемента и проверяет его на то, что он не меньше 0 и не больше длины массива. Если индекс в диапазоне, то возвращается соответствующий элемент. Иначе возвращается null.

Полный код класса HashSet:

```

public class HashSet {

private Integer[] elements = new Integer[10];

public void add(Integer element) {

for (int i = 0; i &lt; elements.length; i++) {

if (elements[i] == null) {

elements[i] = element;

break;

}

}

}

public String toString() {

StringBuilder result = new StringBuilder();

for (int i = 0; i &lt; elements.length; i++) {

if (elements[i] != null) {

result.append(elements[i]).append(",");

}

}

if (result.length() &gt; 0) {

result.setLength(result.length() - 1);

}

return "{" + result.toString() + "}";

}

public Integer get(int index) {

if (index &gt;= 0 &amp;&amp; index &lt; elements.length) {

return elements[index];

}

return null;

}

}

```

Теперь можно создать экземпляр класса HashSet и использовать его методы:

```

HashSet set = new HashSet();

set.add(1);

set.add(2);

set.add(3);

System.out.println(set.toString());

System.out.println(set.get(0));

```

На консоли будет выведено:

```

{1,2,3}

1

```