

```
public class JvmComprehension {  
    // Загружаются системные классы в Metaspace. Потом с помощью  
    // подсистемы загрузчика классов происходит поиск класса (по цепочке  
    // Application ClassLoader -> Platform ClassLoader -> Bootstrap ClassLoader ->  
    // Platform ClassLoader -> Platform ClassLoader), как найдёт происхождение  
    // проверка и подготовка класса и происходит инициализация класса. В момент  
    // инициализации происходит так же загрузка метода «printAll» т.к. он  
    // статичен. Загрузка происходит в Metaspace.
```

```
    public static void main(String[] args) {  
        // Вызывается класс main и в этот момент создаётся фрейм «main()» в  
        // Stack
```

```
        int i = 1;                // 1  
        // В фрейме main() создаётся переменная «i» со значением 1.
```

```
        Object o = new Object();    // 2  
        // в heap выделяется память для объекта Object(). В фрейме main создаётся  
        // переменная «o» со ссылкой на область памяти в heap выделенную для  
        // объекта.
```

```
        Integer ii = 2;            // 3  
        // В фрейме main() создаётся переменная «ii» со значением 2.
```

```
        printAll(o, i, ii);        // 4  
        // В Stack создаётся фрейм printAll. В фрейме printAll создаются переменные  
        // «o», «i», «ii». «o» со ссылкой на объект Object() созданный ранее.  
        // Переменная i со значением 1 и переменная ii со значением 2
```

```
        System.out.println("finished"); // 7  
        // открывается фрейм println и после печати строки удаляется.
```

```
    }  
    //фрейм main() удаляется из Stack
```

```
    private static void printAll(Object o, int i, Integer ii) {  
        Integer uselessVar = 700;    // 5  
        //в heap создаётся область памяти со значением 700 и на него создаётся  
        //ссылка в фрейм printAll.
```

```
        System.out.println(o.toString() + i + ii); // 6  
        // создаётся фрейм println. Создаём фрейм toString со ссылкой на String pull,  
        // фрейм toString удаляется. В heap создаётся строка содержимым объекта
```

«0» объединённый с «1», далее в heap создаётся строка содержащая объединение предыдущей строки и «2», ссылка на эту строку помещается в фрейм println. Сообщение печатается и фрейм println удаляется.

```
    }  
//фрейм printAll удаляется
```

```
}
```

// удаление объектов из кучи происходит под управлением JVM в момент stop-the-world.