public class JvmComprehension {  
  
 public static void main(String[] args) {  
 int i = 1; // 1  
 Object o = new Object(); // 2  
 Integer ii = 2; // 3  
 *printAll*(o, i, ii); // 4  
 System.*out*.println("finished"); // 7  
 }  
  
 private static void printAll(Object o, int i, Integer ii) {  
 Integer uselessVar = 700; // 5  
 System.*out*.println(o.toString() + i + ii); // 6  
 }  
}

1. В JVM Подсистема загрузчиков классов (ClassLoaders) анализирует те классы, которые должны быть вызваны, в первую очередь загружается класс JvmComprehension с методом main(), при этом:

- первым срабатывает Application ClassLoader и отправляет запрос Platform ClassLoader, далее отправляется запрос в Bootstrap ClassLoader .

- если в Bootstrap ClassLoader есть такой файл, он его загружает, если нет – сообщает Platform ClassLoader, что не нашел. Соответственно, если в Platform ClassLoader есть такой класс, он его загружает, если нет - сообщает Application ClassLoader. Соответственно, если в Application ClassLoader есть такой класс, он его загружает если нет – выдает ошибку, что класс не найден (java.lang.ClassNotFoundException).

1. После загрузки класса в область памяти: Metaspace происходит подготовка классов к выполнению (связывание Linking);
2. Далее происходит инициализация класса: Выполняются static инициализаторы и инициализаторы static полей.
3. Далее выполняется метод main():

- в Stack Memory под метод main создается фрейм;

- в данном фрейме записывается примитивная переменная int i = 1;

- оператором new выделяется в куче (heap) память под объект Object();

- в данном фрейме записывается ссылочная переменная Object o;

- выделяется в куче (heap) память под объект Integer ii;

- в данном фрейме записывается ссылочная переменная Integer ii;

- при выходе из фрейма удаляются ссылочные переменные Object o и Integer ii, а также примитивная переменная int i = 1;

1. Далее выполняется метод printAll(o, i, ii):

- в Stack Memory под метод printAll(o, i, ii) создается фрейм;

- в данном фрейме записываются ссылочные переменные Object o и Integer ii, а также примитивная переменная int i = 1;

- выделяется в куче (heap) память под объект Integer uselessVar;

- в данном фрейме записывается ссылочная переменная Integer uselessVar;

- при выходе из фрейма удаляются ссылочные переменные Object o, Integer uselessVar и Integer ii, а также примитивная переменная int i = 1;

1. Далее выполняется метод System.out.println(o.toString() + i + ii):

- в Stack Memory под метод System.out.println(o.toString() + i + ii) создается фрейм;

- выделяется в куче (heap) память под объект String;

- в данном фрейме записываются ссылочные переменные на объект String и Integer ii, а также примитивная переменная int i = 1;

- при выходе из фрейма удаляются ссылочные переменные Object o, Integer uselessVar и Integer ii, а также примитивная переменная int i = 1, сборщик мусора удаляет объекты в куче (heap): Object(), Integer (), String;

1. Далее выполняется метод System.out.println("finished"):

- в Stack Memory под метод System.out.println("finished")создается фрейм;

- выделяется в куче (heap) память под объект String - "finished";

- в данном фрейме записываются ссылочные переменные на объект String - "finished";

- при выходе из фрейма сборщик мусора удаляет объект в куче (heap): String;