public class JvmComprehension {

*Интерпретатор грузит JvmComprehension в Mеtaspace с помощью подсистемы загрузчиков классов ClassLoader. Происходят попытки загрузки класса в Metaspace (хранение метаданных о классе: имя, поля, методы и т.п.) последовательно в Bootstrap Loader, в случае неудачи происходит попытка в Platform ClassLoader и далее в случае неудачи Application ClassLoader. В случае если последняя попытка заканчивается неудачей (не удалось обнаружить класс) выбрасывается Exception.*

public static void main(String[] args) {

*В стеке Stack Memory создается фрейм main()*

int i = 1; // 1

*Интерпретатор создает примитив во фрейме main( ) Stack Memory.*

Object o = new Object(); // 2

*Интерпретатор грузит Object в Mеtaspace с помощью подсистемы загрузчиков классов ClassLoader.*

*Создается объект класса Object и помещается в heap (кучу). В Stack Memory во фрейме main() создается переменная, которая будет ссылаться на этот объект в heap.*

Integer ii = 2; // 3

*Интерпретатор грузит Integer в Mеtaspace с помощью подсистемы загрузчиков классов ClassLoader.*

*Создается объект-оболочка класса Integer и помещается в heap (кучу). В Stack Memory во фрейме main() создается переменная, которая будет ссылаться на этот объект в heap.*

printAll(o, i, ii); // 4

*В Stack Memory создается фрейм метода printAll(). Туда копируются значения ссылок на объекты o, ii, из фрейма main(), а также копируется примитив i.*

*После выхода из метода сборщик мусора может очистить стек и кучу от фреймов, ссылок, объектов. В случае выхода из метода printAll() могут удалиться из кучи объекты uselessVar и o.toString(), а также ссылки на них и сам фрейм метода printAll() из Stack Memory.*

System.out.println("finished"); // 7

}

*В Stack Memory создается фрейм метода println () и туда копируются значения ссылка на объект String – «finished»*

private static void printAll(Object o, int i, Integer ii) {

Integer uselessVar = 700; // 5

*Создается объект-оболочка и помещается в кучу (heap), в стеке во фрейме printAll () создается переменная, которая будет ссылаться на этот объект в куче.*

System.out.println(o.toString() + i + ii); // 6

}

}

*В Stack Memory создается фрейм метода toString() и туда копируются значения ссылок на объект o, ii, из фрейма main(), а также копируется примитив i.*

*В Stack Memory создается фрейм для метода println () и туда копируются значение ссылки на переменную типа String – результат выполнения o.toString(),также копируется ссылка на объект ii, и копируется примитив i.*