**Java.MP.C4.M1.MemoryManagement Default Task07**

1. OutOfMemory heap size

JVM options: -Xmx64m -XX:PretenureSizeThreshold=1

Code:

**public class** Main {  
 **public static void** main(String[] args) **throws** InterruptedException, ClassNotFoundException {  
 StringBuilder sb = **new** StringBuilder();  
 **while**(**true**) {  
 String str = (**new** Double(**new** Random().nextDouble())).toString();  
 sb.append(str);  
 }  
 }  
}

1. OutOfMemory metaspace

JVM options:

-Xmx64m -XX:MaxMetaspaceSize=10m

Code:

**public class** Main {  
 **public static void** main(String[] args) **throws** InterruptedException, ClassNotFoundException, IOException {  
 List<Class> classNames = **new** ArrayList<Class>();  
 ZipInputStream zip = **null**;  
 **try** {  
 zip = **new** ZipInputStream(**new** FileInputStream(**"C:\\Program Files\\Java\\jdk1.8.0\_45\\jre\\lib\\rt.jar"**));  
  
 **for** (ZipEntry entry = zip.getNextEntry(); entry != **null**; entry = zip.getNextEntry()) {  
 Thread.*sleep*(100);  
 **if** (!entry.isDirectory() && entry.getName().endsWith(**".class"**)) {  
 String className = entry.getName().replace(**'/'**, **'.'**);  
 classNames.add(Class.*forName*(className.substring(0, className.length() - **".class"**.length())));  
 }  
 }  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 } **finally** {  
 **if** (zip != **null**) {  
 zip.close();  
 }  
 }  
  
 }  
}

1. StackOverflow

JVM options:

-Xmx64m -Xss1k

Code:

**public class** Main {  
 **public static void** main(String[] args) {  
 *main1*(args);  
  
 }  
  
 **public static void** main1(String[] args) {  
 *main*(args);  
  
 }  
}