```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */


package javaapplication7;

import java.io.IOException;
import java.nio.ByteBuffer;
import java.nio.channels.FileChannel;
import java.nio.file.Paths;
import java.nio.file.Path;
import java.util.Date;
import java.util.Scanner;

public class Main {

public static void main (String [] args)
throws Exception {

          //  Scanner kb4 = new Scanner(System.in);
         //     int choice = kb4.nextInt();

          Scanner s=new Scanner(System.in);
System.out.println("YOU HAVE FOLLOWING CHOICES : ");
System.out.println("1.Create adirect byte buffer ");
System.out.println("2. Manage size ");
System.out.println("3. Clean buffer ");

System.out.println("ENTER YOUR CHOICE : ");
int i=s.nextInt();



System.out.println("ENTER SECOND NUMBER ");
int b=s.nextInt();

double result=0;//'result' will store the result of operation

switch(i)
```

```
{
case 1:
result=a+b;
                        Create_DirBuffer();

break;
case 2:
//result=a-b;
                        System.out.println("input a new size ");
            int a=s.nextInt();
                        Direct_buffer()
break;
case 3:
result=a*b;
                        Clear_buffer();
break;
case 4:
if(b==0)//when denominator becomes zero
{
System.out.println("DIVISION NOT POSSIBLE");
break;
}
else
result=a/b;

default:
System.out.println("YOU HAVE ENTERED A WRONG CHOICE");

}

System.out.println("RESULT = "+result);


      //    select_menu(choice);


long startTime = new Date().getTime();

Path path = Paths.get("c:\\thesis2\\hj_93.doc");
FileChannel fileChannel = FileChannel.open(path);
```

```
//ByteBuffer buffer = ByteBuffer.allocate(1024 * 10);
ByteBuffer buffer = ByteBuffer.allocateDirect(1024 * 10);

System.out.println("Is a direct buffer: " + buffer.isDirect());
System.out.println("Buffer has a backing array: " + buffer.hasArray());
System.out.println("Reading file... ");

int noOfBytesRead = fileChannel.read(buffer);

for (int i2 = 0; i2 < 25; i++) {

while (noOfBytesRead != -1) {

buffer.clear();
noOfBytesRead = fileChannel.read(buffer);
}

buffer.clear();
fileChannel.position(0);
noOfBytesRead = fileChannel.read(buffer);
}

fileChannel.close();

long endTime = new Date().getTime();
System.out.println("");
System.out.println("Time taken (millis): " + (endTime - startTime));

      //        System.out.println();



}
```

```
public static void select_menu(int grade) {
```

```
        int success;
        switch (grade) {
        case 1:
            System.out.println("Excellent grade");
            success = 1;
            break;
        case 2:
            System.out.println("Very good grade");
            success = 1;
            break;

        default:
            System.out.println("Invalid grade");
            success = -1;
        }
}
```

```
}
```