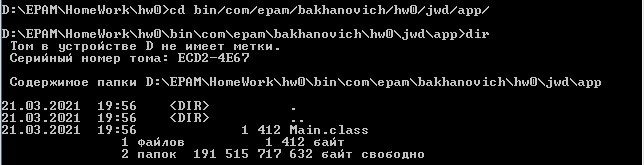
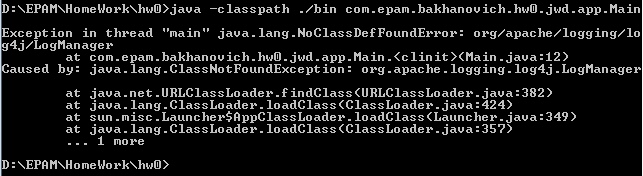


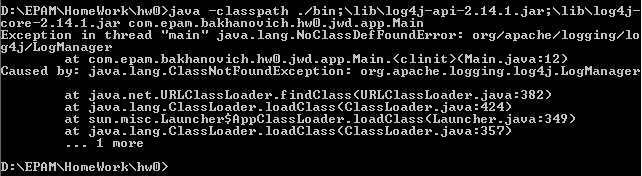
Создалась директория bin в которой по пути \com\epam\bakhanovich\hw0\jwd\app скомпилировался файл Main.class



Попытка 1



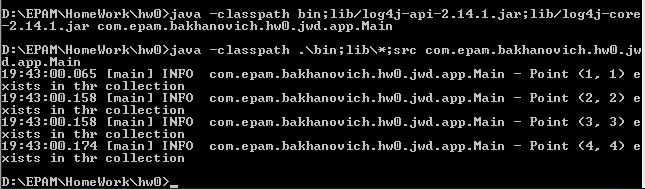
Попытка 2



Попытка 3



Попытка 4



Исходные файлы:

**Main.java**

package com.epam.bakhanovich.hw0.jwd.app;  
  
import com.epam.bakhanovich.hw0.jwd.model.Point;  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class Main {  
  
 private static final Logger *log* = LogManager.*getLogger*(Main.class);  
  
 public static void main(String[] args) {  
 List<Point> points = new ArrayList<>();  
 final Point point1 = new Point(1,1);  
 final Point point2 = new Point(2,2);  
 final Point point3 = new Point(3,3);  
 final Point point4 = new Point(4,4);  
 points.add(point1);  
 points.add(point2);  
 points.add(point3);  
 points.add(point4);  
  
 for (Point point: points  
 ) {  
 *log*.info("Point " + point.toString() + " exists in thr collection");  
 }  
 }  
}

**Point.java**

package com.epam.bakhanovich.hw0.jwd.model;  
  
import java.util.Objects;  
  
*/\*\*  
 \* A {@link Point}  
 \*/*public class Point implements Comparable<Point>{  
  
 private final double coordinateX;  
 private final double coordinateY;  
  
 */\*\*  
 \* Constructs a new {@link Point} containing coordinates X and Y.  
 \*/* public Point(double coordinate\_x, double getCoordinate\_y) {  
 coordinateX = coordinate\_x;  
 coordinateY = getCoordinate\_y;  
 }  
  
 */\*\*  
 \* Returns the coordinate X of the {@link Point}  
 \*  
 \* @return the value of the parameter coordinateX of this {@link Point}  
 \*/* public double getCoordinateX() {  
 return coordinateX;  
 }  
  
 */\*\*  
 \* Returns the coordinate Y of the {@link Point}  
 \*  
 \* @return the value of the parameter coordinateY of this {@link Point}  
 \*/* public double getCoordinateY() {  
 return coordinateY;  
 }  
  
 */\*\*  
 \* Returns 0 if X and Y coordinates of this {@link Point} and  
 \* of the {@link Point} to be compared with are equal.  
 \* Returns -1 if coordinate X of this {@link Point} less than coordinate X  
 \* of the {@link Point} to be compared with.  
 \* Returns -1 if coordinate X of this {@link Point} equal to  
 \* the coordinate X of the {@link Point} to be compared with and  
 \* coordinate Y of this {@link Point} less than coordinate Y  
 \* of the {@link Point} to be compared with.  
 \* Returns 1 otherwise.  
 \*  
 \* @param other is the {@link Point} the object to be compared with  
 \*  
 \* @return a negative integer, zero, or a positive integer as this object  
 \* is less than, equal to, or greater than the specified object  
 \*/* @Override  
 public int compareTo(Point other) {  
 if (this.coordinateX > other.coordinateX) {  
 return 1;  
 } else if (this.coordinateX < other.coordinateX) {  
 return -1;  
 } else if (this.coordinateY > other.coordinateY) {  
 return 1;  
 } else if (this.coordinateY < other.coordinateY) {  
 return -1;  
 } else return 0;  
 }  
  
 */\*\*  
 \* Prints text view of the {@link Point}. The {@link Point} is represented  
 \* in the following way: (x1, y1), where x1 = coordinate X  
 \* of this {@link Point} and y1 = coordinate Y of this {@link Point}.  
 \*  
 \* @return String, that represents tupel of the coordinates  
 \* of this {@link Point}  
 \*/* @Override  
 public String toString() {  
 StringBuilder builder = new StringBuilder();  
  
 double x = this.getCoordinateX();  
 double y = this.getCoordinateY();  
 builder.append("(")  
 .append(Math.*round*(x))  
 .append(", ")  
 .append(Math.*round*(y))  
 .append(")");  
 return builder.toString();  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(coordinateX, coordinateY);  
 }  
}

**log4j2.xml**

*<?*xml version="1.0" encoding="UTF-8"*?>*<Configuration status="INFO">  
 <Appenders>  
 <Console name="LogToConsole" target="SYSTEM\_OUT">  
 <PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/>  
 </Console>  
 <File name="ErrorFileAppender" fileName="logs/error.log" immediateFlush="false" append="false">  
 <PatternLayout pattern="%d{yyy-MM-dd HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/>  
 </File>  
 </Appenders>  
 <Loggers>  
 *<!-- avoid duplicated logs with additivity=false -->* <Logger name="com.epam" level="info" additivity="false">  
 <AppenderRef ref="LogToConsole" level="info"/>  
 <AppenderRef ref="ErrorFileAppender" level="error"/>  
 </Logger>  
 <Root level="info">  
 <AppenderRef ref="LogToConsole"/>  
 </Root>  
 </Loggers>  
</Configuration>

