```java
/*
 * Copyright (C) 2016 redxef.
 *
 * This library is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 2.1 of the License, or (at your option) any later version.
 *
 * This library is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this library; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
 * MA 02110-1301  USA
 */
package powerspy.baselib;

import java.io.*;

/**
 *
 * @author redxef
 */
public class PSInputStream extends InputStream implements IODefs {

        private final InputStream is;
        private DataPacket dp;
        private boolean reading;
        private boolean start_flag;

        /**
         * Creates a new InputStream which is set up for data transfer with the
         * PIC. This InputStream will create a DataPacket and interact with it
         * to read the values.
         *
         * @param is
         */
        public PSInputStream(InputStream is)
        {
                this.is = is;
        }

        @Override
        public int available() throws IOException
        {
                return is.available();
        }

        @Override
        public int read() throws IOException
        {
                return is.read();
        }

        @Override
        public synchronized void reset() throws IOException
        {
                is.reset();
        }

        @Override
        public synchronized void mark(int readlimit)
        {
                is.mark(readlimit);
        }

        private byte read_() throws IOException
        {
                return (byte) (read() & 0xff);
        }

        /**
         * Clears this InputStream. This method should be called after a
         * DataPacket has been read and further data is expected to arrive.
         *
         * @return this PSInputStream for chaincalling
         */
        public PSInputStream clear()
        {
                dp = new DataPacket();
                reading = false;
                return this;
        }

        /**
         * Reads a DataPacket and returns true if the Packet is finished and
         * ready for interacting with.
         *
         * @return a boolean indication the sate of the DataPacket
         *
         * @throws IOException
         */
        public boolean readPackage() throws IOException
        {
                byte r;
                while (available() > 0 || !dp.isFinished()) {
                        if (reading) {
                                if (start_flag) {
                                        dp.setType((char) read_());
                                        start_flag = false;
                                }
                                r = read_();
                                if (dp.isString() && r == END_OF_TEXT) {
                                        dp.setFinishedString();
                                } else {
                                        dp.addByte(r);
                                }
                                if (dp.isFinished()) {
                                        reading = false;
                                        return dp.isFinished();
                                }
                        } else {
                                if (read_() == START_OF_TEXT) {
                                        reading = true;
                                        start_flag = true;
                                }
                        }
                }
                return dp.isFinished();
        }

        /**
         * Returns the underlying DataPacket.
         *
         * @return the DataPacket
         */
        public DataPacket getDataPacket()
        {
                return dp;
        }

}
```