

```

import java.net.*;
import java.io.*;
import java.util.regex.Pattern;
import java.util.regex.Matcher;
import java.util.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class ObtainData
{
    private JFrame frame;
    private JPanel checkBoxPanel;
    private JCheckBox iNews;
    private JCheckBox nPub;
    private JCheckBox pPub;
    private JCheckBox dComp;
    private JCheckBox tMark;
    private JButton button;
    public ObtainData()
    {
        frame = new JFrame();
        frame.setSize(600,100);
        frame.setTitle("Extra Research");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        checkBoxPanel = new JPanel();
        iNews = new JCheckBox("In News");
        checkBoxPanel.add(iNews);
        nPub = new JCheckBox("Negative Publicity");
        checkBoxPanel.add(nPub);
        pPub = new JCheckBox("Positive Publicity");
        checkBoxPanel.add(pPub);
        dComp = new JCheckBox("Direct Competition");
        checkBoxPanel.add(dComp);
        tMark = new JCheckBox("Tapped Market");
        checkBoxPanel.add(tMark);
        button = new JButton("Start Program");
        checkBoxPanel.add(button);
        frame.add(checkBoxPanel);
        frame.setLocationByPlatform(true);
        frame.setVisible(true);
        button.addActionListener(new Action());
    }
    class Action implements ActionListener
    {
        public void actionPerformed(ActionEvent a){
            String line = "";
            try{
                URL dow = new URL("http://money.cnn.com/data/markets/

```

```

dow/");
        BufferedReader in = new BufferedReader(new
InputStreamReader(dow.openStream()));
        String inputLine;
        while((inputLine = in.readLine()) != null)
        {
            line += inputLine + " ";
        }
        in.close();
    }catch(IOException e){
        System.exit(1);
    }
    ArrayList<Double> data = new ArrayList<Double>();
    int i = 0;
    i = betterFind(line, "\\d\\d,\\d\\d\\d\\d\\.\\d\\d", data, i,
true, false);
    i = betterFind(line, "<span class=.posData.>.(\\d+\\.?.\
\d+)</span>", data, i, false, true);
    i = betterFind(line, "<span class=.posData.>.(\\d+\\..\
\d+)%</span>", data, i, true, true);
    Company thing = new Company(data.get(0), data.get(1),
data.get(2), data.get(3), data.get(4), data.get(5),
iNews.isSelected(), nPub.isSelected(), pPub.isSelected(),
dComp.isSelected(), tMark.isSelected(), "Dow Jones Industrial");
    if(thing.shouldInvest())
    {
        System.out.println(thing.getName() + " looks good to
invest in.");
    }
}

private static int betterFind(String line, String regex,
ArrayList<Double> data, int i, boolean getAll, boolean wholeLine)
{
    Pattern p = Pattern.compile(regex);
    Matcher m = p.matcher(line);
    boolean stop = true;
    if(wholeLine == false)
    {
        while(m.find() && stop)
        {
            data.add(Double.parseDouble(m.group(0).substring(0,2)
+ m.group(0).substring(3,9)));
            System.out.println(data.get(i));
            i++;
            stop = getAll;
        }
    }else{
        while(m.find() && stop)

```

```

        {
            Double thing = Double.parseDouble(m.group(2));
            if(m.group(1).equals("+"))
            {
                data.add(thing);
            }else{
                data.add(0-thing);
            }
            System.out.println(data.get(i));
            i++;
            stop = getAll;
            if(data.size() == 9)
            {
                stop = false;
            }
        }
    }
    return(i);
}

public static void main()
{
    ObtainData test = new ObtainData();
}

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class Company implements Comparable<Company>
{
    private JFrame frame;
    private JPanel panel;
    private double stockPrice;
    private double previousClose;
    private double open;
    private double dayHigh;
    private double dayLow;
    private double projStockPrice;
    private double percentChange;
    private boolean inNews;
    private boolean negPublicity;
    private boolean posPublicity;
    private boolean directCompetition;
    private boolean tappedMarket;
    private final String name;
    private double change;
    private boolean projIncrease;

```

```

    public Company(double stockPrice, double previousClose, double
open, double dayHigh, double dayLow, double percentChange, boolean
inNews, boolean negPublicity, boolean posPublicity, boolean
directCompetition, boolean tappedMarket, String name)
    {
        this.open = open;
        this.dayHigh = dayHigh;
        this.dayLow = dayLow;
        this.percentChange = percentChange;
        this.stockPrice = stockPrice;
        this.previousClose = previousClose;
        this.inNews = inNews;
        this.negPublicity = negPublicity;
        this.posPublicity = posPublicity;
        this.directCompetition = directCompetition;
        this.tappedMarket = tappedMarket;
        this.name = name;
        setprojStockPrice();
    }

    @Override
    public int compareTo(Company thing)
    {
        int toRet = 0;
        toRet = (int)(thing.getchange() - change);
        if(posPublicity && !thing.getposPublicity())
        {
            toRet--;
        }
        if(negPublicity && !thing.getnegPublicity())
        {
            toRet++;
        }
        if(tappedMarket && !thing.gettappedMarket())
        {
            toRet += 2;
        }
        if(toRet == 0 && !getName().equals(thing.getName()))
        {
            return(-1);
        }
        return(toRet);
    }

    public double getstockPrice()
    {
        return(stockPrice);
    }

    public void setstockPrice(double stockPrice)

```

```

{
    this.stockPrice = stockPrice;
    setprojStockPrice();
}

public double getpreviousClose()
{
    return(previousClose);
}

public void setpreviousClose(double previousClose)
{
    this.previousClose = previousClose;
    setprojStockPrice();
}

public double getprojStockPrice()
{
    setprojStockPrice();
    return(projStockPrice);
}

public void setprojStockPrice()
{
    double toRet = (stockPrice - previousClose) + stockPrice;
    boolean other = false;
    if(inNews)
    {
        if(negPublicity)
        {
            toRet -= stockPrice*.2;
        }
        else if(posPublicity)
        {
            toRet += stockPrice*.2;
        }
        else
        {
            toRet += stockPrice*.1;
        }
        other = true;
    } else if(stockPrice - previousClose < 0)
    {
        toRet = ((stockPrice - previousClose) * .5) + stockPrice;
    }

    if(directCompetition && !other)
    {
        toRet = toRet * .8;
    }
}

```

```

        if(tappedMarket && !other)
        {
            toRet = toRet * .6;
        }
        projStockPrice = toRet;
        setchange();
    }

    public boolean getinNews()
    {
        return(inNews);
    }

    public void setinNews(boolean inNews)
    {
        this.inNews = inNews;
        setprojStockPrice();
    }

    public boolean getnegPublicity()
    {
        return(negPublicity);
    }

    public void setnegPublicity(boolean negPublicity)
    {
        this.negPublicity = negPublicity;
        setprojStockPrice();
    }

    public boolean getposPublicity()
    {
        return(posPublicity);
    }

    public void setposPublicity(boolean posPublicity)
    {
        this.posPublicity = posPublicity;
        setprojStockPrice();
    }

    public boolean getdirectCompetition()
    {
        return(directCompetition);
    }

    public void setdirectCompetition(boolean directCompetition)
    {
        this.directCompetition = directCompetition;
    }

```

```

        setprojStockPrice();
    }

    public boolean gettappedMarket()
    {
        return(tappedMarket);
    }

    public void settappedMarket(boolean tappedMarket)
    {
        this.tappedMarket = tappedMarket;
        setprojStockPrice();
    }

    public String getname()
    {
        return(name);
    }

    public double getchange()
    {
        return(change);
    }

    public void setchange()
    {
        change = projStockPrice - stockPrice;
    }

    public boolean shouldInvest()
    {
        if(projStockPrice >= stockPrice)
        {
            return(true);
        }else{
            return(false);
        }
    }

    @Override
    public String toString()
    {
        return(name + " has a current stock price of $" + stockPrice +
" and has a projected stock price of $" + projStockPrice + ".");
    }
}

```