# Oppgave 1

1a)

import java.util.ArrayList;

public abstract class Verdipapir implements Comparable<Verdipapir> {

    protected final String isin;

    protected String ticker;

    protected String navn;

    protected static ArrayList<Sluttverdi> sluttverdier;

    public Verdipapir(String isin, String ticker, String navn) {

        this.isin = isin;

        this.ticker = ticker;

        this.navn = navn;

    }

    public String getIsin() {

        return this.isin;

    }

    public String getTicker() {

        return this.ticker;

    }

    public void setTicker(String Ticker) {

        this.ticker = Ticker;

    }

    public String getNavn() {

        return this.navn;

    }

    public void setNavn(String navn) {

        this.navn = navn;

    }

    public String toString() {

        return getTicker() + " " + getNavn();

    }

    public boolean equals(Verdipapir verdipapir) {

        return this.getIsin() == verdipapir.isin;

    }

    public int compareTo(Verdipapir verdipapir) {

        return this.getNavn().compareTo(verdipapir.getNavn());

    }

}

1b)

OmsetteligeVerdipapirer:

import java.util.GregorianCalendar;

public abstract class OmsetteligeVerdipapir extends Verdipapir {

    protected GregorianCalendar tidspunkt;

    protected double kursverdi;

    protected int antallHandler;

    public OmsetteligeVerdipapir(String isin, String Ticker, String navn, GregorianCalendar tidspunkt, double kursverdi,

            int antallHandler) {

        super(isin, Ticker, navn);

        if (kursverdi < 0 || antallHandler < 0) {

            return;

        }

        this.tidspunkt = tidspunkt;

        this.kursverdi = kursverdi;

        this.antallHandler = antallHandler;

    }

    public GregorianCalendar getTid() {

        return this.tidspunkt;

    }

    public void setTid(GregorianCalendar tidspunkt) {

        this.tidspunkt = tidspunkt;

    }

    public double getKursverdi() {

        return this.kursverdi;

    }

    public void setKursverdi(double kursverdi) {

        this.kursverdi = kursverdi;

    }

    public int getAntallHandler() {

        return this.antallHandler;

    }

    public void setAntallHandler(int antallHandler) {

        this.antallHandler = antallHandler;

    }

    public abstract void nullstill();

}

Aksje:

import java.util.GregorianCalendar;

public class Aksje extends OmsetteligeVerdipapir {

    enum Sektor {

        TELEKOM, MINERAL, LANDBRUK, SKOGBRUK, FANGST, FISKE, HELSE, BUTIKK, BYGG, UNDERVISNING, DATA

    }

    private double palydende;

    private Sektor sektor;

    public Aksje(String isin, String Ticker, String navn, GregorianCalendar tidspunkt, double kursverdi,

            int antallHandler, double palydende, Sektor sektor) {

        super(isin, Ticker, navn, tidspunkt, kursverdi, antallHandler);

        this.palydende = palydende;

        this.sektor = sektor;

    }

    @Override

    public void nullstill() {

        setKursverdi(0);

        setAntallHandler(0);

        setTid(null);

    }

}

Tegningsrett:

import java.util.GregorianCalendar;

public class Tegningsrett extends OmsetteligeVerdipapir {

    private String tegningsdato;

    private double tegningskursen;

    private Aksje aksje;

    public Tegningsrett(String isin, String Ticker, String navn, GregorianCalendar tidspunkt, double kursverdi,

            int antallHandler, String tegningsdato, double tegningskursen, Aksje aksje) {

        super(isin, Ticker, navn, tidspunkt, kursverdi, antallHandler);

        this.tegningsdato = tegningsdato;

        this.tegningskursen = tegningskursen;

        this.aksje = aksje;

    }

    public String getTegningsdato() {

        return this.tegningsdato;

    }

    public void setTegningsdato(String tegningsdato) {

        this.tegningsdato = tegningsdato;

    }

    public double getTegningskursen() {

        return this.tegningskursen;

    }

    public void setTegningskursen(double tegningskursen) {

        this.tegningskursen = tegningskursen;

    }

    public Aksje getAksje() {

        return this.aksje;

    }

    public void setAksje(Aksje aksje) {

        this.aksje = aksje;

    }

    @Override

    public void nullstill() {

        setKursverdi(0);

        setAntallHandler(0);

        setTid(null);

    }

}

1c)

Aksjefond:

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Map;

public class Aksjefond extends Verdipapir {

    private String forvalter;

    private ArrayList<Aksje> aksjer;

    public Aksjefond(String isin, String Ticker, String navn, String type, String forvalter) {

        super(isin, Ticker, navn, type);

        this.forvalter = forvalter;

    }

    public String getForvalter() {

        return this.forvalter;

    }

    public void setForvalter(String forvalter) {

        this.forvalter = forvalter;

    }

    public ArrayList<Aksje> getAksjer() {

        return this.aksjer;

    }

    public void setAksjer(ArrayList<Aksje> aksjer) {

        this.aksjer = aksjer;

    }

    public void leggTilAksje(Aksje aksje) {

        int id = aksjer.indexOf(aksje);

        if (id == -1) {

            aksjer.add(aksje);

        }

    }

    public void fjernAksje(Aksje aksje) {

        int id = aksjer.indexOf(aksje);

        if (id != -1) {

            aksjer.remove(id);

        }

    }

    public int antallAksjer() {

        Map<String, String> antallAksjerSelskap = new HashMap<String, String>();

        for (Aksje aksje : aksjer) {

            if (!antallAksjerSelskap.containsKey(aksje.navn)) {

                antallAksjerSelskap.put(aksje.navn, "Selskap");

            }

        }

        return antallAksjerSelskap.size();

    }

}

1d)

Handel klassen blir da seende slik ut:

import java.util.GregorianCalendar;

public class Handel {

    private String selger, kjøper;

    private GregorianCalendar tidspunkt; // dato+klokkeslett for handelen

    private double kurs; // salgskurs, pris pr. verdipapir

    private OmsetteligeVerdipapir omsetteligeVerdipapir; // Referanse til det omsettelige verdipapiret

    private int volum; // antall solgte verdipapirer

    public Handel(String selger, String kjøper, GregorianCalendar tidspunkt, double kurs, int volum,

            OmsetteligeVerdipapir omsetteligeVerdipapir) {

        this.selger = selger;

        this.kjøper = kjøper;

        this.tidspunkt = tidspunkt;

        this.kurs = kurs;

        omsetteligeVerdipapir.setKursverdi(kurs);

        omsetteligeVerdipapir.setAntallHandler(omsetteligeVerdipapir.getAntallHandler() + 1);

        omsetteligeVerdipapir.setTid(tidspunkt);

    }

    public double getTotalBelop() {

        return this.volum \* kurs;

    }

    public String getSelger() {

        return selger;

    }

    public String getKjøper() {

        return kjøper;

    }

    public GregorianCalendar getTidspunkt() {

        return tidspunkt;

    }

    public double getKurs() {

        return kurs;

    }

    public int getVolum() {

        return volum;

    }

}

1e)

Sluttverdi klassen:

import java.util.GregorianCalendar;

public class Sluttverdi {

    final public Verdipapir vp;

    final public GregorianCalendar dato;

    final public double sluttkurs;

    final public int antallHandler;

    public Sluttverdi(Verdipapir vp, GregorianCalendar dato, double sluttkurs, int antallHandler) {

        this.vp = vp;

        this.dato = dato;

        this.sluttkurs = sluttkurs;

        this.antallHandler = antallHandler;

    }

}

Slik blir da oppdaterte Verdipapir klassen seende ut:

import java.util.ArrayList;

public abstract class Verdipapir implements Comparable<Verdipapir> {

    protected final String isin;

    protected String ticker;

    protected String navn;

    protected static ArrayList<Sluttverdi> sluttverdier;

    public Verdipapir(String isin, String ticker, String navn) {

        this.isin = isin;

        this.ticker = ticker;

        this.navn = navn;

    }

    public String getIsin() {

        return this.isin;

    }

    public String getTicker() {

        return this.ticker;

    }

    public void setTicker(String Ticker) {

        this.ticker = Ticker;

    }

    public String getNavn() {

        return this.navn;

    }

    public void setNavn(String navn) {

        this.navn = navn;

    }

    public String toString() {

        return getTicker() + " " + getNavn();

    }

    public boolean equals(Verdipapir verdipapir) {

        return this.getIsin() == verdipapir.isin;

    }

    public abstract void dagSlutt();

    public double[] getSluttKurser() {

        double[] sluttkurser = new double[this.sluttverdier.size()];

        for (int i = 0; i < this.sluttverdier.size(); i++) {

            sluttkurser[i] = this.sluttverdier.get(i).sluttkurs;

        }

        return sluttkurser;

    }

    public int compareTo(Verdipapir verdipapir) {

        return this.getNavn().compareTo(verdipapir.getNavn());

    }

}

Slik blir oppdaterte Aksje klassen seende ut:

import java.util.GregorianCalendar;

public class Aksje extends OmsetteligeVerdipapir {

    enum Sektor {

        TELEKOM, MINERAL, LANDBRUK, SKOGBRUK, FANGST, FISKE, HELSE, BUTIKK, BYGG, UNDERVISNING, DATA

    }

    private double palydende;

    private Sektor sektor;

    public Aksje(String isin, String Ticker, String navn, GregorianCalendar tidspunkt, double kursverdi,

            int antallHandler, double palydende, Sektor sektor) {

        super(isin, Ticker, navn, tidspunkt, kursverdi, antallHandler);

        this.palydende = palydende;

        this.sektor = sektor;

    }

    @Override

    public void dagSlutt() {

        Sluttverdi sluttverdi = new Sluttverdi(this, tidspunkt, kursverdi, antallHandler);

        sluttverdier.add(sluttverdi);

    }

    @Override

    public void nullstill() {

        setKursverdi(0);

        setAntallHandler(0);

        setTid(null);

    }

}

Slik blir oppdaterte Tegningsrett seende ut:

import java.util.GregorianCalendar;

public class Tegningsrett extends OmsetteligeVerdipapir {

    private String tegningsdato;

    private double tegningskursen;

    private Aksje aksje;

    public Tegningsrett(String isin, String Ticker, String navn, GregorianCalendar tidspunkt, double kursverdi,

            int antallHandler, String tegningsdato, double tegningskursen, Aksje aksje) {

        super(isin, Ticker, navn, tidspunkt, kursverdi, antallHandler);

        this.tegningsdato = tegningsdato;

        this.tegningskursen = tegningskursen;

        this.aksje = aksje;

    }

    public String getTegningsdato() {

        return this.tegningsdato;

    }

    public void setTegningsdato(String tegningsdato) {

        this.tegningsdato = tegningsdato;

    }

    public double getTegningskursen() {

        return this.tegningskursen;

    }

    public void setTegningskursen(double tegningskursen) {

        this.tegningskursen = tegningskursen;

    }

    public Aksje getAksje() {

        return this.aksje;

    }

    public void setAksje(Aksje aksje) {

        this.aksje = aksje;

    }

    @Override

    public void dagSlutt() {

        Sluttverdi sluttverdi = new Sluttverdi(this, tidspunkt, kursverdi, antallHandler);

        sluttverdier.add(sluttverdi);

    }

    @Override

    public void nullstill() {

        setKursverdi(0);

        setAntallHandler(0);

        setTid(null);

    }

}

Slik blir oppdaterte Aksjefond seende ut:

import java.util.ArrayList;

import java.util.GregorianCalendar;

import java.util.HashMap;

import java.util.Map;

public class Aksjefond extends Verdipapir {

    private String forvalter;

    private ArrayList<Aksje> aksjer;

    public Aksjefond(String isin, String Ticker, String navn, String forvalter) {

        super(isin, Ticker, navn);

        this.forvalter = forvalter;

    }

    public String getForvalter() {

        return this.forvalter;

    }

    public void setForvalter(String forvalter) {

        this.forvalter = forvalter;

    }

    public ArrayList<Aksje> getAksjer() {

        return this.aksjer;

    }

    public void setAksjer(ArrayList<Aksje> aksjer) {

        this.aksjer = aksjer;

    }

    public void leggTilAksje(Aksje aksje) {

        int id = aksjer.indexOf(aksje);

        if (id == -1) {

            aksjer.add(aksje);

        }

    }

    public void fjernAksje(Aksje aksje) {

        int id = aksjer.indexOf(aksje);

        if (id != -1) {

            aksjer.remove(id);

        }

    }

    public int antallAksjer() {

        Map<String, String> antallAksjerSelskap = new HashMap<String, String>();

        for (Aksje aksje : aksjer) {

            if (!antallAksjerSelskap.containsKey(aksje.navn)) {

                antallAksjerSelskap.put(aksje.navn, "Selskap");

            }

        }

        return antallAksjerSelskap.size();

    }

    @Override

    public void dagSlutt() {

        double kursverdi = 0;

        for (Aksje aksje : aksjer) {

            kursverdi += aksje.kursverdi;

        }

        kursverdi = kursverdi / aksjer.size();

        GregorianCalendar calendar = new GregorianCalendar();

        Sluttverdi sluttverdi = new Sluttverdi(this, calendar, kursverdi, 0);

        sluttverdier.add(sluttverdi);

    }

}

            }

        }

        return antallAksjerSelskap.size();

    }

    @Override

    public void dagSlutt() {

        double kursverdi = 0;

        for (Aksje aksje : aksjer) {

            kursverdi += aksje.kursverdi;

        }

        kursverdi = kursverdi / aksjer.size();

        GregorianCalendar calendar = new GregorianCalendar();

        Sluttverdi sluttverdi = new Sluttverdi(this, calendar, kursverdi, 0);

        sluttverdier.add(sluttverdi);

    }

}

# Oppgave 2

Applikasjons klassen Run:

import java.util.ArrayList;

import Eksamen.Aksje.Sektor;

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.CheckBox;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.layout.BorderPane;

import javafx.scene.layout.HBox;

import javafx.scene.layout.Pane;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

public class Run extends Application {

    static ArrayList<Verdipapir> verdipapirer;

    static SluttKursVisuelt tegning = new SluttKursVisuelt(verdipapirer);

    ComboBox<String> box = new ComboBox<>();

    CheckBox aksjer = new CheckBox("Aksjer");

    CheckBox tegningsretter = new CheckBox("Tegningsretter");

    CheckBox aksjefond = new CheckBox("Aksjefond");

    public static void main(String[] args) {

        launch(args);

    }

    public void start(Stage window) throws Exception {

        BorderPane hovedPane = new BorderPane();

        hovedPane.setCenter(tegning);

        hovedPane.setTop(lagTop());

        hovedPane.setLeft(lagListe());

        Scene scene = new Scene(hovedPane, 800, 400);

        window.setTitle("Eksamen");

        window.setScene(scene);

        window.show();

        aksjer.setOnAction(action -> {

            hovedPane.setCenter(new SluttKursVisuelt(verdipapirer));

        });

        tegningsretter.setOnAction(action -> {

            hovedPane.setCenter(new SluttKursVisuelt(verdipapirer));

        });

        aksjefond.setOnAction(action -> {

            hovedPane.setCenter(new SluttKursVisuelt(verdipapirer));

        });

    }

    protected Pane lagListe() {

        VBox vBox = new VBox();

        vBox.getChildren().addAll(new Label("Velg verdipapir:"));

        for (Verdipapir verdipapir : verdipapirer) {

            box.getItems().add(verdipapir.toString());

        }

        return vBox;

    }

    protected Pane lagTop() {

        VBox vBox = new VBox();

        HBox hBox = new HBox();

        hBox.getChildren().addAll(aksjer, tegningsretter, aksjefond);

        vBox.getChildren().add(new Label("Inkluder følgende verdipapirtyper:"));

        vBox.getChildren().addAll(hBox);

        return vBox;

    }

}

Klassen SLuttKursVisuelt:

import java.util.ArrayList;

import javafx.scene.layout.HBox;

import javafx.scene.shape.Rectangle;

public class SluttKursVisuelt extends HBox {

    public SluttKursVisuelt(ArrayList<Verdipapir> verdipapirer) {

        super();

        for (Verdipapir verdipapir : verdipapirer) {

            Rectangle r = new Rectangle();

            double[] kurser = verdipapir.getSluttKurser();

            for (int i = 0; i < kurser.length; i++) {

                r.setX(0);

                r.setY(0);

                r.setWidth(1);

                r.setHeight(kurser[i]);

                getChildren().add(r);

            }

        }

    }

}