UFR IM²AG

UNIVERSITÉ Grenoble Alpes

INF362 - Quick2
Année 2016-2017

Durée : 1h.
Tout document ou calculatrice interdit.
Pour chaque question, une partie des points peut tenir compte de la présentation.
Le barême est indicatif.

```
class M<X> {
    X e;
    M<X> s;

    M(X e, M<X> s) {
        this.e = e;
        this.s = s;
    }
}
```

```
class I<X> {
    E<X> e;
    M<X> a, p, s;
    boolean c;

    I(E<X> e) {
        this.e = e;
        s = e.t;
    }

    boolean a() {
        return s != null;
    }

    X p() {
        a = p;
        p = s;
        s = s.s;
        c = true;
        return p.e;
    }

    void s() {
        if (c) {
            if (a != null)
                a.s = s;
            else
                e.t = s;
            p = a;
            c = false;
        } else {
            throw new
            RuntimeException("I.s()");
        }
    }
}
```

```
class E<X> {
    M<X> t;

    void a(X o) {
        t = new M<>(o, t);
    }

    I<X> i() {
        return new I<>(this);
    }

    void m(T<X> t) {
        I<X> i = i();
        while (i.a()) {
            if (t.t(i.p()))
                i.s();
        }
    }
}
```

```
interface T<X> {
    boolean t(X o);
}
```

```
class N extends E<F> {
    void a(V v) {
        v.v(this);
        m(new T<F>() {
            @Override
            public boolean t(F f) {
                return f.a(v);
            }
        });
    }
}
```

```
class P {
    double x, y;

    P(double x, double y) {
        this.x = x;
        this.y = y;
    }

    P(P c) {
        this(c.x, c.y);
    }

    P a(P z) {
        x += z.x;
        y += z.y;
        return this;
    }

    double n() {
        return Math.sqrt(x*x+y*y);
    }

    P m(double f) {
        x *= f;
        y *= f;
        return this;
    }

    double s(P z) {
        return x*z.x+y*z.y;
    }
}
```

```
abstract class F {
    P p;
    P v;

    F(double x, double y) {
        p = new P(x, y);
    }

    void d() {
        p.a(v);
    }

    boolean a(V v) {
        return v.v(this);
    }

    void v(double x, double y) {
        v = new P(x, y);
    }

    P i(P x) {
        if (x != null)
            return x.m(-1);
        else
            return null;
    }

    abstract P c(F f);
    abstract P c(C c);
    abstract P c(R r);
}
```

```
class C extends F {
    double r;
    N o;

    C(N o, double x, double y, double r) {
        super(x, y);
        this.o = o;
        this.r = r;
    }

    @Override
    boolean a(V v) {
        return v.v(this);
    }

    @Override
    P c(F f) {
        return i(f.c(this));
    }

    P m() {
        return new P(p.x + r, p.y + r);
    }

    @Override
    P c(C c) {
        double d = r + c.r;
        P s = c.m().m(-1).a(m());
        double n = s.n();
        if (n < d) {
            return s;
        } else
            return null;
    }

    @Override
    P c(R r) {
        double l = r.d.x/2;
        double i = (p.x+this.r) - (r.p.x+l);
        double h = r.d.y/2;
        double j = (p.y+this.r) - (r.p.y+h);
        double k = (l+this.r);
        double m = (h+this.r);

        if (i*i < l*l) {
            if (j*j < h*h)
                return new P(i, j);
            else if (j*j < m*m)
                return new P(0, j);
        } else if (i*i < k*k) {
            if (j*j < h*h)
                return new P(i, 0);
            else if (j*j < m*m) {
                    i = (i>l)?i-l:i+l;
                    j = (j>h)?j-h:j+h;
                    P s = new P(i, j);
                    if (s.n() < this.r)
                        return s;
                }
        }
        return null;
    }

    public boolean r() {
        d();
        F s = this;
        o.a(new V() {
            @Override
            public boolean v(F f) {
                if (f != s) {
                    P c = s.c(f);
                    if (c != null) {
                        c.m(1/c.n());
                        double z = s.v.s(c);
                        if (z < 0) {
                            c.m(-2*z);
                            s.v.a(c);
                        }
                    }
                }
                return false;
            }
        });
        return false;
    }
}
```

```
abstract class V {
    boolean v(N o) {
        return false;
    }

    boolean v(F f) {
        return false;
    }

    boolean v(C c) {
        return v((F) c);
    }

    boolean v(R r) {
        return v((F) r);
    }
}
```

```
class R extends F {
    P d;

    R(double x, double y,
            double l, double h) {
        super(x, y);
        d = new P(l, h);
    }

    @Override
    boolean a(V v) {
        return v.v(this);
    }

    @Override
    P c(F f) {
        return i(f.c(this));
    }

    @Override
    P c(C c) {
        return i(c.c(this));
    }

    @Override
    P c(R r) {
        return i(r.c(this));
    }
}
```

```
import javafx.scene.canvas.Canvas;
import javafx.scene.canvas.GraphicsContext;

class D extends V {
    Canvas c;
    GraphicsContext g;

    D(Canvas c) {
        this.c = c;
        g = c.getGraphicsContext2D();
    }

    boolean v(N o) {
        g.clearRect(0, 0, c.getWidth(),
                c.getHeight());
        return false;
    }

    boolean v(C c) {
        g.strokeOval(c.p.x, c.p.y,
                2*c.r, 2*c.r);
        return false;
    }

    boolean v(R r) {
        g.strokeRect(r.p.x, r.p.y,
                r.d.x, r.d.y);
        return false;
    }
}
```

```
import java.util.Random;
import javafx.animation.AnimationTimer;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.canvas.Canvas;
import javafx.stage.Stage;
import javafx.scene.layout.BorderPane;

public class A extends Application {
    @Override
    public void start(Stage m) {
        final int l=800, h=600;
        m.setTitle("Interro PROG6");
        Canvas c = new Canvas(l, h);
        BorderPane b = new BorderPane(c);
        m.setScene(new Scene(b));
        m.show();
        D d = new D(c);
        N o = new N();
        Random r = new Random();
        for (int i=0; i<10; i++) {
            C x = new C(o, r.nextDouble()*l,
                r.nextDouble()*h,
                r.nextDouble()*(l+h)/50+10);
            final int f = 4;
            x.v((r.nextDouble()-0.5d)*f,
                (r.nextDouble()-0.5d)*f);
            o.a(x);
        }
        o.a(new R(-1,0,1,h));
        o.a(new R(0,-1,l,1));
        o.a(new R(l,0,1,h));
        o.a(new R(0,h,l,1));
        double t = 20;
        o.a(new R(l/2-2*t,h/2-t,4*t,2*t));

        AnimationTimer a =
            new AnimationTimer() {
            @Override
            public void handle(long t) {
                o.a(new V() {
                    @Override
                    public boolean v(C c) {
                        return c.r();
                    }
                });
                o.a(d);
            }
        };
        a.start();
    }

    public static void main(String[] a) {
        launch(a);
    }
}
```

Questions :

1. pour chacune des classes ou interfaces ci-dessus, écrivez un paragraphe distinct qui :
   — indique quel est le rôle de la classe ou interface dans le programme ;
   — explique comment elle remplit ce rôle.
2. expliquez ce que fait le programme dans son ensemble