

Exercice 1.

```

public class Chambre {
    int nombreLits;
    double superficie;
    int nombreLitsReserves;
    int numeroChambre;

    → public Chambre(int nombreLits, double superficie, int numeroChambre) {
        this.nombreLits = nombreLits;
        this.superficie = superficie;
        this.numeroChambre = numeroChambre;
        this.nombreLitsReserves = 0;
    }

    → public Chambre(double superficie, int numeroChambre) {
        this(1, superficie, numeroChambre);
    }

    → public int litsLibres() {
        return this.nombreLits - this.nombreLitsReserves;
    }

    → public boolean reserver(int nombrePersonne) {
        if (this.litsLibres() >= nombrePersonne) {
            this.nombreLitsReserves += nombrePersonne;
            return true;
        }
        return false;
    }

    → public boolean reserver() {
        return this.reserver(1);
    }

    → public boolean transfertReservation(Chambre chambre) {
        if (chambre.reserver(this.nombreLitsReserves)) {
            this.nombreLitsReserves = 0;
            return true;
        }
        return false;
    }

    → public int capacite() {
        return this.nombreLits;
    }

    → public boolean cMonNumero(int numero) {
        return this.numeroChambre == numero;
    }

    → public int numeroChambre() {
        return this.numeroChambre;
    }
}

public class Hotel {
    private ArrayList<Chambre> chambres;

    → public Hotel() {
        chambres = new ArrayList<Chambre>();
    }

    → public Chambre chercherChambre(int numero) {
        for (Chambre ch : chambres) {
            if (ch.cMonNumero(numero)) return ch;
        }
        return null;
    }

    → public void reserverLits(int nombrePersonne) {
        for (Chambre ch : chambres) {
            if (ch.litsLibres() > nombrePersonne)

```

```

public ArmeRaffale(int cartouche, int raffale){
    super(cartouche);
    this.raffale=raffale;
}
public void tirer(){
    for(int i=0; i<this.raffale;i++) super.tirer();
}

```

```

class ArmeSensibles extends Arme{
    int temperatureCritique;
    int temperature;
}

```

```

ArmeSensibles(int cartouche, int temperature, int temperatureCritique) {
    super(cartouche);
    this.temperature=temperature;
    this.temperatureCritique=temperatureCritique;
    this.refroidir();
}

```

```

boolean critique(){
    return (this.temperature>this.temperatureCritique);
}
void refroidir() { if ( this.critique() ) this.temperature=this.temperatureCritique-1;}

```

```

public void tirer(){
    super.tirer();
    temperature++;
    this.refroidir();
}

```

```

public class Soldat {
    ArrayList<Arme> arsenal;
}

```

```

void tirMassif(){
    for(Arme a:arsenal)a.tirer();
}

```

```

void cooling(){
    for(Arme a:arsenal) if (a instanceof ArmeSensibles) ((ArmeSensibles) a).refroidir();
}

```

```

void ajouterArme(int cartouche, int temperature, int temperatureCritique){
    this.arsenal.add(new ArmeSensibles(cartouche,temperature,temperatureCritique));
}

```