

Activité : Calculs en aquariophilie

Thierry Vaira <tvaira@free.fr>

01/02/2016 (rev. 1)

Table des matières

Calculs en aquariophilie	1
Détermination du volume et poids d'un aquarium	1
Détermination de la puissance de chauffage	1
Autres	2

Calculs en aquariophilie

Lien : www.aquaportail.com

Code source : [test-calculs-aquarium.zip](#)

Détermination du volume et poids d'un aquarium

Il vous faut connaître les longueur, largeur et hauteur en cm, on calculera ensuite le volume en litres.

Si on indique l'épaisseur du verre en mm pour avoir le volume interne, il est possible d'estimer le poids de l'aquarium. Le calcul est réalisé depuis une densité de 2,53 pour du verre sodo-calcique de qualité courante sans tenir compte du poids des renforts, entretoises, raidisseurs, verre de couverture, ...

```
void calcul_cuве()
{
    // en cm
    double longueur = 150;
    double largeur = 100;
    double hauteur = 100;
    // en mm
    double epaisseurVerre = 20;
    double densiteVerre = 2.53;

    cout << "Aquarium (en cm) : " << longueur << "x" << largeur << "x" << hauteur << endl << endl;
    cout << "Volume brut = " << (longueur*largeur*hauteur/1000.) << " litres" << endl;

    double longueurCorrigee = ((longueur*10.)-(2*epaisseurVerre))/10.;
    double largeurCorrigee = ((largeur*10.)-(2*epaisseurVerre))/10.;
    double hauteurCorrigee = ((hauteur*10.)-epaisseurVerre)/10.;
    double volumeVerre = ((longueur*hauteur*2.)+(largeur*hauteur*2.)+(longueur*largeur))*epaisseurVerre/10.;

    cout << "Volume interne = " << (longueurCorrigee*largeurCorrigee*hauteurCorrigee/1000.) << " litres" << endl;
    cout << "1 cm de hauteur dans la cuве vaut : " << (longueurCorrigee*largeurCorrigee/1000.) << " litres" << endl;
    cout << "Poids cuве à vide = " << (volumeVerre*densiteVerre/1000.) << " Kg" << endl;
}
```

Détermination de la puissance de chauffage

Il vous faut connaître les dimensions de votre aquarium (L x l x h) en cm et les 2 températures demandées en Celcius : celle de la pièce dans laquelle va être l'aquarium est au minimum de 10C et doit être inférieure à celle de l'eau de l'aquarium.

On calcule la puissance indicative du chauffage en Watts (par multiple de 25 W).

```

void calcul_chauffage()
{
    // en cm
    double longueur = 150;
    double largeur = 100;
    double hauteur = 100;
    //
    double temperaturePiece = 25;
    double temperatureEau = 27;
    double delta = temperatureEau-temperaturePiece;
    double surface;

    cout << "Aquarium (en cm) : " << longueur << "x" << largeur << "x" << hauteur << endl << endl;

    if(delta < 0.)
    {
        temperatureEau = temperaturePiece;
        delta = 0.;
    }
    surface = ((longueur*hauteur*2.)+(largeur*hauteur*2.)+(longueur*largeur));
    double chauffage = (2.*0.6*surface*delta/1000.);
    double chauffageWatts = (chauffage-(fmod(chauffage, 25))+25);

    cout << "Température de la pièce = " << temperaturePiece << " C" << endl;
    cout << "Température de l'eau = " << temperatureEau << " C" << endl;
    cout << "Delta = " << delta << " C" << endl;
    cout << "Surface = " << surface << " cm^2" << endl;
    cout << "Chauffage en watts = " << chauffage << " W" << endl;
    cout << "Chauffage en watts (par multiple de 25W) = " << chauffageWatts << " W" << endl; // par multiple de
    25
}

```

Autres

```

std::string IntToString(int n)
{
    std::ostringstream oss;
    oss << n;
    return oss.str();
}

std::string DoubleToString(double n)
{
    std::ostringstream oss;
    oss << n;
    return oss.str();
}

void calcul_durete()
{
    //
    double thEauOsmose = 0.; // f -> supposée à 0
    double thEau = 15.; // f -> Eau moyennement dure
    double thFinal = 8.; // f -> Eau douce
    //
    double volumeFinal = 100.;
    double volumeEauOsmose;
    double volumeEauRobinet;

    cout << "Détermination des volumes à mélanger pour une dureté voulue" << endl;

    cout << "TH eau du robinet = " << thEau << " f" << endl;
    cout << "TH souhaité = " << thFinal << " f" << endl;
    cout << "Volume souhaité = " << volumeFinal << " litres" << endl;
}

```

```

volumeEauOsmose = (volumeFinal*(thEau-thFinal))/(thEau-thEauOsmose); // Volume eau osmosé
volumeEauRobinet = (volumeFinal*(thFinal-thEauOsmose))/(thEau-thEauOsmose); // Volume eau de conduite

cout << "Il faut mélanger " << volumeEauOsmose << " litres d'eau osmosée et " << volumeEauRobinet << " litres
d'eau du robinet." << endl;
cout << endl;
}

void calculMGfromCA()
{
    double th = 15.; // f
    double gh = th*56.0774/100.0869; // all
    double tca = 10.0; // mg/l Calcium
    double r = ((th/10000.)-(tca/40078.))*24305.; // mg/l Magnésium

    cout << "Détermination du taux de Magnésium" << endl;

    cout << "TH = " << th << " f" << endl;
    cout << "GH = " << gh << " all" << endl;
    cout << "Taux de Calcium = " << tca << " mg/l" << endl;
    cout << "Taux de Magnésium = " << r << " mg/l" << endl;
    cout << endl;
}

void calculGHThfromCAMG()
{
    double tca = 10.;
    double tmg = 30.39;
    double caco3 = (100.0869/40.078*tca)+(100.0869/24.305)*tmg;
    double th = caco3/10;
    double gh = th*56.0774/100.0869;

    cout << "Calculs de dureté totale (TH - GH) à partir des taux de Calcium et de Magnésium" << endl;

    cout << "Taux de Calcium = " << tca << " mg/l" << endl;
    cout << "Taux de Magnésium = " << tmg << " mg/l" << endl;
    cout << "TH = " << th << " f" << endl;
    cout << "GH = " << gh << " all" << endl;
    cout << endl;
}

void calculCO2()
{
    double r1 = 8; // KH all
    double r2 = 7; // pH
    double tauxco2 = (ceil((3*r1*pow(10,7-r2))*100)/100); // mg/l CO2

    cout << "Détermination du taux de CO2" << endl;

    cout << "KH = " << r1 << " all" << endl;
    cout << "pH = " << r2 << " " << endl;
    cout << "Taux CO2 = " << tauxco2 << " mg/l" << endl;
    cout << endl;
}

void maketableCO2()
{
    double co2;
    double ph, kh;
    string c, s;

    s = "";
    for(kh=0; kh<=20; kh++)
    {
        if(kh==0) s += "KH / PH ";
    }
}

```

```

else s += IntToString(kh) + "\t";
//for(ph=60;ph<=80;ph++)
for(ph=60;ph<=71;ph++)
{
    if(kh!=0)
    {
        co2 = (ceil((3*kh*pow(10,7-(ph/10)))*100.)/100.);
        if(co2<15.)
            {c="ok";}
        else if(co2<40.)
            {c="juste";}
        else if(co2<100.)
            {c="danger";}
        else
            {c="mortel";}
    }
    else
    {
        co2=(ph/10.);
    }
    //s += DoubleToString(co2) + "(" + c + ")";
    s += DoubleToString(co2) + "\t" ;
}
s += "\n";
}
cout << s << endl ;

cout << "Les valeurs entre 0 et 15 mg/l de CO2 sont insuffisantes." << endl;
cout << "Les valeurs entre 15 et 40 mg/l de CO2 sont les meilleures." << endl;
cout << "Les valeurs entre 40 et 100 mg/l de CO2 sont dangeuses à très dangeuses." << endl;
cout << "Les valeurs au dessus de 100 mg/l de CO2 sont mortelles." << endl;
cout << endl;
}

```

Code source : [test-calculs-aquarium.zip](#)

[Retour au sommaire](#)