

Formule

$$m = \frac{C \cdot \frac{r}{n}}{1 - (1 + \frac{r}{n})^{-N}}$$

Légende

- C Capital initial
 m Mensualité
 n Nombre de paiements par an
 N Nombre total d'annuités (nombre d'années x remboursements par an)
 r Taux d'intérêt nominal annuel

Fichier Édition Affichage Rechercher Terminal Aide

Calcul des mensualités pour une opération financière:

Capital initial : 5000

Nombre d'années d'emprunt : 5

Taux d'intérêt nominal annuel (en %) : 3

Vous devez payer votre crédit pendant 60 mensualités.

Avec pour chaque mois un crédit de 89.84 €.

Le coup total du crédit s'élève à 5390.61 €.

La somme de vos intérêts est de 390.61 €.

Appuyez sur <ENTRÉE> pour fermer cette fenêtre...

main.cpp

1

```
#include <iostream>
#include <iomanip>
#include <math.h>
using namespace std;

int main()
{
    double C, N, r, m;
    const double n=12;

    // Saisie
    cout << "Calcul des mensualités pour une opération financière:" << endl;
    cout << "Capital initial : ";
    cin >> C;
    cout << "Nombre d'années d'emprunt : ";
    cin >> N;
    cout << "Taux d'intérêt nominal annuel (en %) : ";
    cin >> r;

    // Traitement
    r /= 100;
    m = C*r/n/(1-pow(1+r/n, -N*n));

    // Affichage
    cout << "Vous devez payer votre crédit pendant " << n*N << " mensualités." << endl;
    cout << fixed << setprecision(2);
    cout << "Avec pour chaque mois un crédit de " << m << " €." << endl;
    cout << "Le coup total du crédit s'élève à " << m*n*N << " €." << endl;
    cout << "La somme de vos intérêts est de " << m*n*N-C << " €." << endl;

    return 0;
}
```

```

QT      += core gui

greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

TARGET = finance
TEMPLATE = app

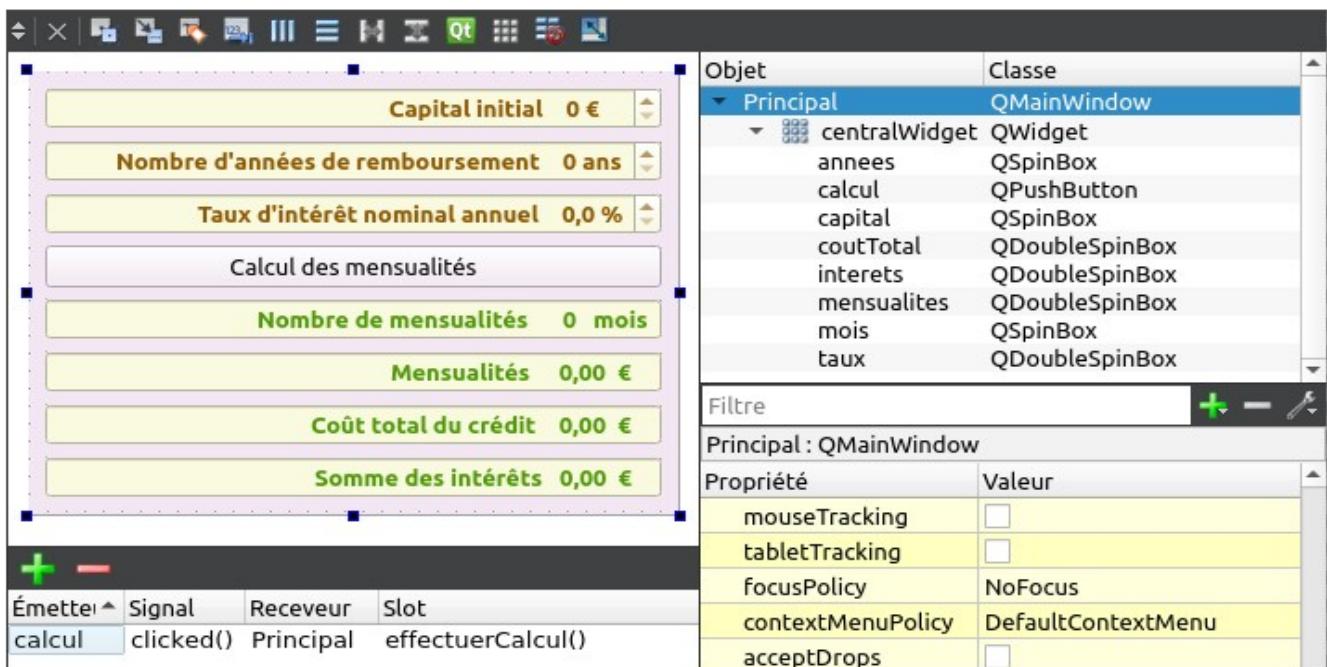
DEFINES += QT_DEPRECATED_WARNINGS

CONFIG += c++11

SOURCES += main.cpp principal.cpp
HEADERS += principal.h
FORMS += principal.ui

# Default rules for deployment.
qnx: target.path = /tmp/$${TARGET}/bin
else: unix:!android: target.path = /opt/$${TARGET}/bin
!isEmpty(target.path): INSTALLS += target

```



```

#ifndef PRINCIPAL_H
#define PRINCIPAL_H

#include "ui_principal.h"

class Principal : public QMainWindow, private Ui::Principal
{
    Q_OBJECT

public:
    explicit Principal(QWidget *parent = nullptr);
private slots:
    void effectuerCalcul();
private:
    const int n=12;
};

#endif // PRINCIPAL_H

```

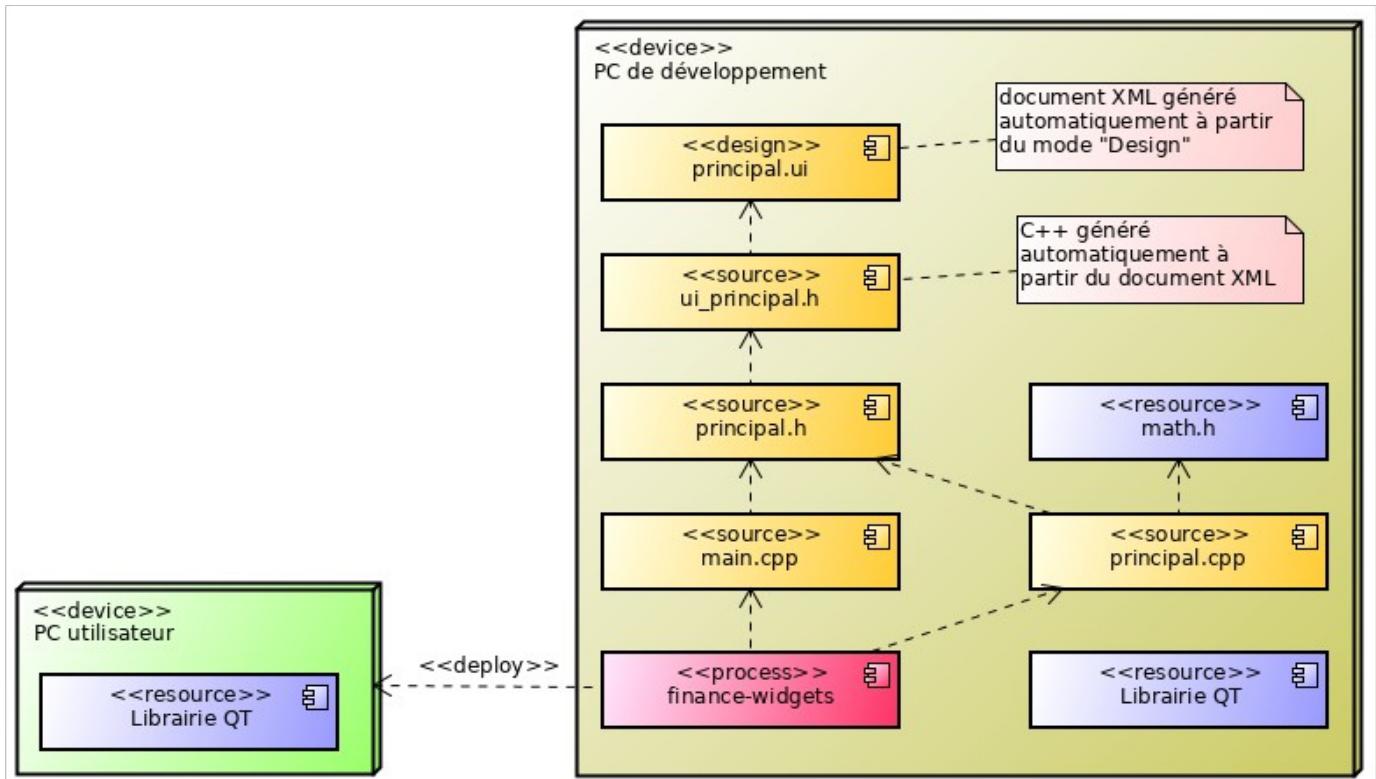
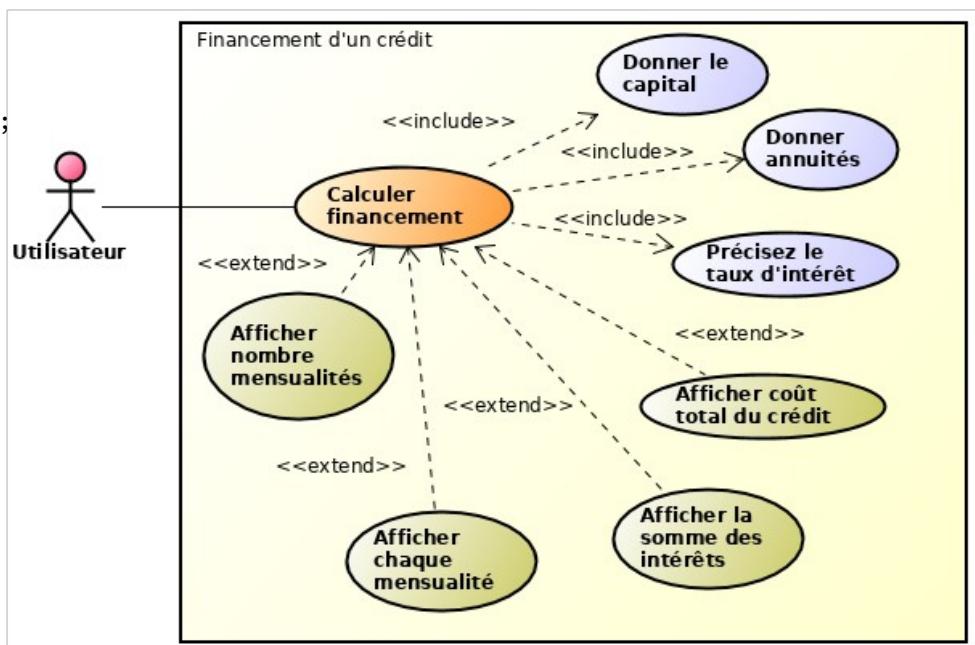
```
#include "principal.h"
#include <math.h>

Principal::Principal(QWidget *parent) : QMainWindow(parent)
{
    setupUi(this);
}

void Principal::effectuerCalcul()
{
    // Récupération des différentes valeurs
    int C = capital->value();
    int N = annees->value() * n;
    double r = taux->value() / 100;

    // Calcul des mensualités
    double m = C*r/n/(1-pow(1+r/n, -N));

    // Affichage du résultat
    mois->setValue(N);
    mensualites->setValue(m);
    coutTotal->setValue(m*N);
    interets->setValue(m*N-C);
}
```



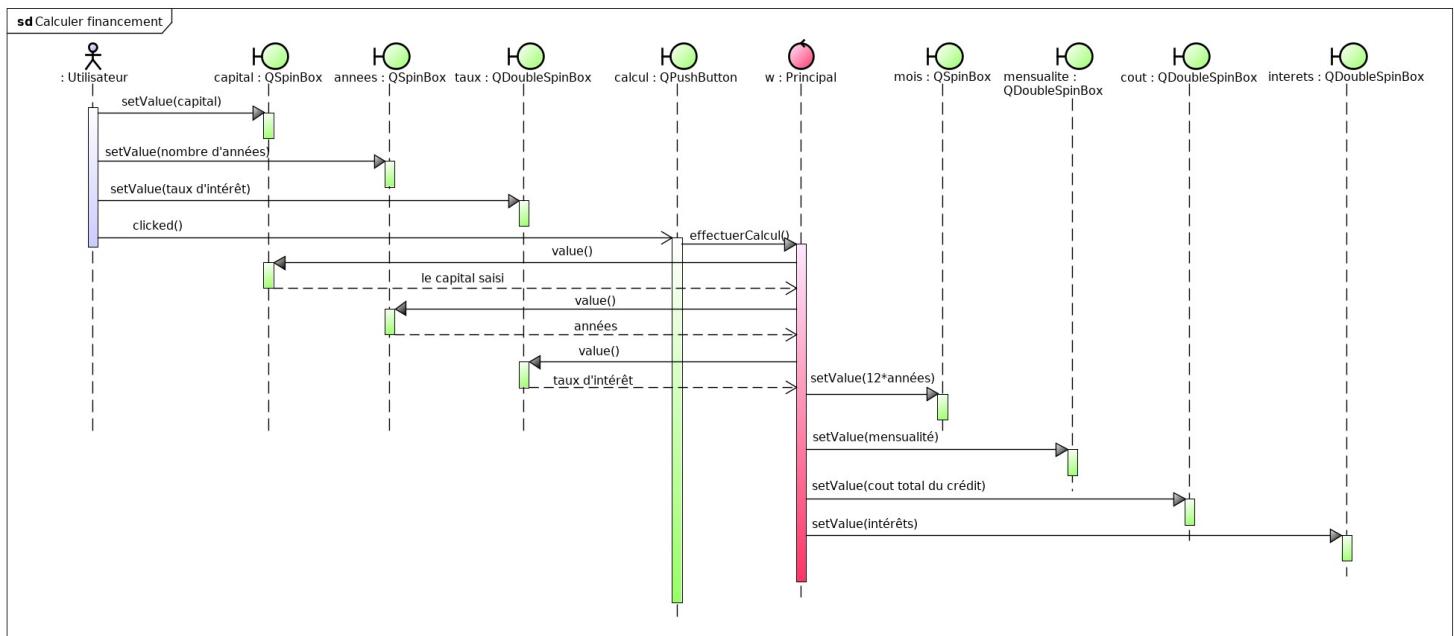
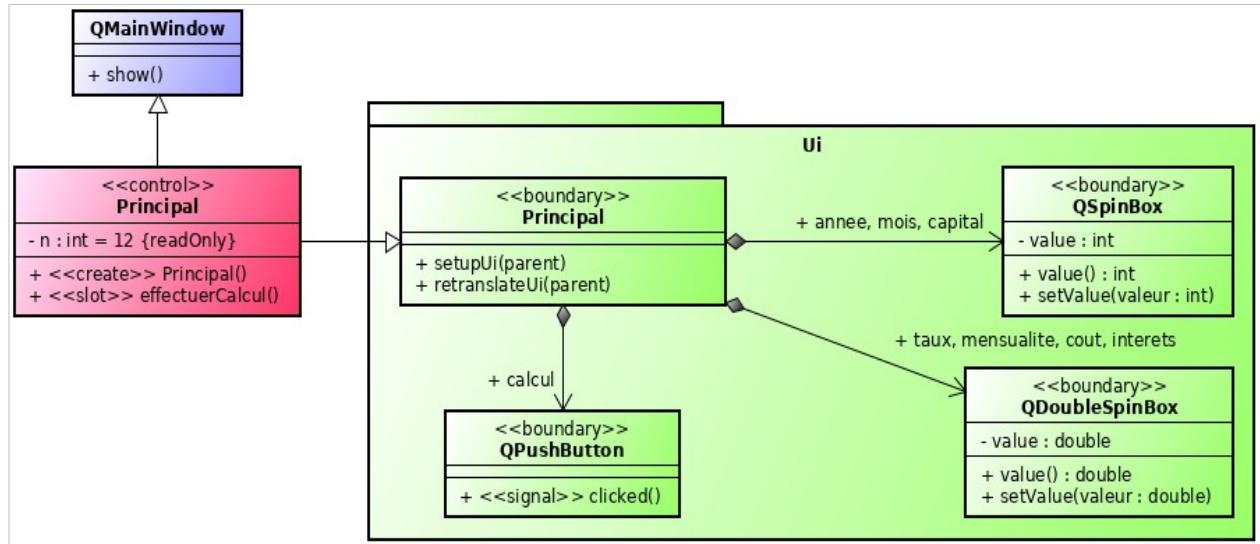
```
#include "principal.h"
#include <math.h>

Principal::Principal(QWidget *parent) : QMainWindow(parent)
{
    setupUi(this);
}

void Principal::effectuerCalcul()
{
    // Récupération des différentes valeurs
    int C = capital->value();
    int N = annees->value() * n;
    double r = taux->value() / 100;

    // Calcul des mensualités
    double m = C*r/n/(1-pow(1+r/n, -N));

    // Affichage du résultat
    mois->setValue(N);
    mensualites->setValue(m);
    coutTotal->setValue(m*N);
    interets->setValue(m*N-C);
}
```



```
#ifndef CONTROLEUR_H
#define CONTROLEUR_H

#include <QObject>
#include <math.h>

class Controleur : public QObject
{
    Q_OBJECT
    Q_PROPERTY(int capital MEMBER C)
    Q_PROPERTY(int annes WRITE annutes)
    Q_PROPERTY(double taux WRITE taux)
    Q_PROPERTY(double mensualite READ mensualite)
    Q_PROPERTY(double interets READ interets)
    Q_PROPERTY(double coutTotal READ coutTotal)
    Q_PROPERTY(int mois READ nombreMensualites)

public:
    explicit Controleur(QObject *parent = nullptr) : QObject(parent) {}

private:
    void annutes(int annes) { N = annes*n; }
    void taux(double pourcentage) { r = pourcentage/100; }
    double mensualite() { return m = C*r/n/(1-pow(1+r/n, -N)); }
    double interets() { return m*N-C; }
    double coutTotal() { return m*N; }
    int nombreMensualites() { return N; }

private:
    const int n=12;
    int C, N;
    double r, m;
};

#endif // CONTROLEUR_H
```



Projets

- finance-qml
 - Finance-qml.pro
 - Headers
 - contrroleur.h
 - Sources
 - main.cpp
 - Resources
 - qml.qrc
 - /vues
 - Bouton.qml
 - ChampDeTexte.qml
 - main.qml
 - Résultat.qml
 - Saisie.qml

```

1 QT += quick
2 CONFIG += c++11
3
4 DEFINES += QT_DEPRECATED_WARNINGS
5
6 SOURCES += main.cpp
7 HEADERS += contrroleur.h
8 RESOURCES += qml.qrc
9
10 # Default rules for deployment.
11 qnx: target.path = /tmp/$${TARGET}/bin
12 else: unix:!android: target.path = /opt/$${TARGET}/bin
13 !isEmpty(target.path): INSTALLS += target

```

```
import QtQuick 2.12
import QtQuick.Controls 2.12

TextField {
    property string intitulé
    property string symbole: "€"
    property string couleur: "darkred"
    property double valeur: 0.0
    property int décimales: -1
    property alias fond: fond

    font.pointSize: 16
    font.bold: true
    rightPadding: 70
    horizontalAlignment: TextInput.AlignRight
    text: "0.0"

    anchors {
        left: parent.left
        leftMargin: 20
        right: parent.right
        rightMargin: 20
    }

    onDécimalesChanged: text = Number(valeur).toLocaleString(Qt.locale("fr_FR"), 'f',
décimales)
    onValeurChanged: text = Number(valeur).toLocaleString(Qt.locale("fr_FR"), 'f',
décimales)

    background: Rectangle {
        id: fond
        color: "#BBFFFFFF"
        radius: 7
    }

    Text {
        id: préfixe
        text: intitulé
        color: couleur
        font.pointSize: 14
        font.italic: true
        anchors {
            verticalCenter: parent.verticalCenter
            left: parent.left
            leftMargin: 10
        }
    }

    Text {
        id: suffixe
        text: symbole
        font.pointSize: 16
        color: couleur
        width: 50
        anchors {
            verticalCenter: parent.verticalCenter
            right: parent.right
            rightMargin: 10
        }
    }
}
```

```

import QtQuick 2.12
import QtQml 2.12
import QtQuick.Controls 2.12

ChampDeTexte {
    inputMethodHints: Qt.ImhDigitsOnly
    selectedTextColor: "darkred"
    selectionColor: "#20FF0000"
    fond.border.color: "darkred"

    onFocusChanged: {
        fond.color = focus ? "white" : "#BBFFFFFF"
        if (focus) selectAll()
        else {
            valeur = Number.fromString(Qt.locale("fr_FR"), text)
            deselect()
        }
    }
}

```

```

import QtQuick 2.12
import QtQuick.Controls 2.12

ChampDeTexte {
    couleur: "darkgreen"
    readOnly: true
    fond.color: "#BBCCFFCC"
    fond.border.color: "darkgreen"
}

```

```

import QtQuick 2.12
import QtQuick.Controls 2.12

Button {
    font {
        bold: true
        italic: true
        pointSize: 16
    }

    anchors {
        left: parent.left
        leftMargin: 20
        right: parent.right
        rightMargin: 20
    }

    background: Rectangle {
        id: fond
        color: "orangered"
        radius: 7
        opacity: 0.4
    }
    onPressed: fond.color = "darkred"
    onReleased: fond.color = "orangered"
}

```

```
import QtQuick 2.12
import QtQuick.Window 2.12
import QtQuick.Layouts 1.12
import QtGraphicalEffects 1.0
import QtQuick.Controls 2.12

Window {
    visible: true
    width: 380
    height: 570
    title: qsTr("Calcul de mensualités")

    LinearGradient {
        anchors.fill: parent
        gradient: Gradient {
            GradientStop { position: 0.0; color: "bisque" }
            GradientStop { position: 1.0; color: "lightgreen" }
        }
    }

    Column {
        anchors.fill: parent
        padding: 20
        spacing: 20

        Saisie {
            id: capital
            intitulé: "Capital"
            décimales: 0
        }

        Saisie {
            id: années
            symbole: "ans"
            intitulé: "Nombre d'annuités"
            décimales: 0
        }

        Saisie {
            id: taux
            symbole: "%"
            intitulé: "Taux d'intérêt"
            décimales: 1
        }

        Bouton {
            text: "Calcul des mensualités"
            onClicked: {
                controleur.capital = capital.valeur
                controleur.annees = années.valeur
                controleur.taux = taux.valeur
                mensualité.valeur = controleur.mensualite
                coûtTotal.valeur = controleur.coutTotal
                intérêts.valeur = controleur.interets
                mois.valeur = controleur.mois
            }
        }

        Résultat {
            id: mois
            symbole: "mois"
            intitulé: "Nombre de mensualités"
            décimales: 0
        }
    }
}
```

```

Résultat {
    id: mensualité
    intitulé: "Mensualité"
    décimales: 2
}

Résultat {
    id: coûtTotal
    intitulé: "Coût total"
    décimales: 2
}

Résultat {
    id: intérêts
    intitulé: "Intérêts"
    décimales: 2
}
}
}
}

```

Formule

$$m = \frac{C \cdot \frac{r}{n}}{1 - (1 + \frac{r}{n})^{-N}}$$

Légende

<i>C</i>	Capital initial
<i>m</i>	Mensualité
<i>n</i>	Nombre de paiements par an
<i>N</i>	Nombre total d'annuités (nombre d'années x remboursements par an)
<i>r</i>	Taux d'intérêt nominal annuel

Calcul de mensualités

Capital	5 000 €
Nombre d'annuités	5 ans
Taux d'intérêt	3,0 %
Calcul des mensualités	
Nombre de mensualités	60 mois
Mensualité	89,84 €
Coût total	5 390,61 €
Intérêts	390,61 €

