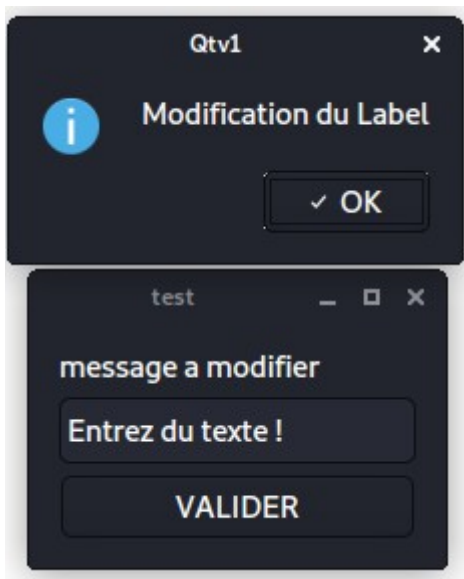


1. Test 1**2. Premier exercice : Convertisseur Kms → Miles (6 pts)****3. Différence entre *QgridLayout* et *QVBoxLayout*.****QgridLayout :**

Permet de disposer le layout des objet sous forme de grille

QVBoxLayout :

Permet de disposer le layout des objets en verticale.

```
layGeneral->addWidget(labelTitre,1,1,1,4);
```

dispose l'objet dans la grille de la façon suivante :

premier ligne, premier colonne, largeur 1 ligne et de longueur 4 colonnes ;

```
layGeneral->addWidget(lineEditKm1,3,1);
```

dispose l'objet dans la grille de la façon suivante :

troisième ligne, premier colonne, sans largeur et longueur;

4. Expliquez l'intérêt d'hériter de *QWidget* dans la classe *Convkm*

L'intérêt d'hériter de *QWidget* dans la class *Convkm*, nous permet d'utiliser les methode de celle-ci.(nous créons ainsi un widget de base ou nous regroupons nos objets)

5. Expliquez ces 4 lignes de connect, en utilisant la fiche signal et slot.

```
connect(lineEditKm1,SIGNAL(editingFinished()),this,SLOT(convertirKmMilles()));  
connect(lineEditMilles2,SIGNAL(editingFinished()),this,SLOT(convertirMillesKm()));
```

```
));  
connect(radioKmM, SIGNAL(clicked()), this, SLOT(desactiverMilles()));  
connect(radioMKM, SIGNAL(clicked()), this, SLOT(desactiverKm()));
```

ligne 1 : Nous permet d'associer l'action d'écrire dans lineEditKm1 au slot convertirKmMilles()

ligne 2 : Nous permet d'associer l'action d'écrire dans lineEditMilles2 au slot convertirMillesKm()

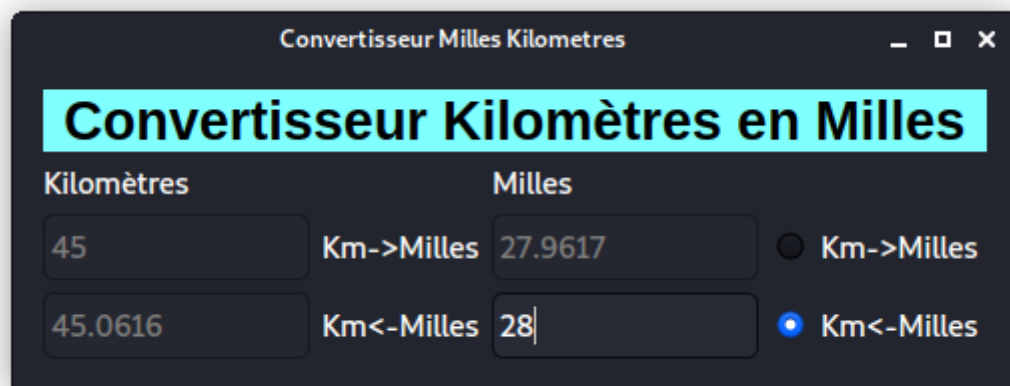
ligne 3 : Nous permet d'associer l'action de cliquer dans radioKmM au slot desactiverMilles()

ligne 4 : Nous permet d'associer l'action de cliquer dans radioMKM au slot desactiverKm()

6. Copie du code de ConvKm.cpp

```
#include "convkm.h"  
ConvKM::ConvKM(QWidget *parent)  
: QWidget(parent)  
{  
    // QTextCodec::setCodecForCStrings(QTextCodec::codecForName("UTF-8"));  
    lineEditKm1=new QLineEdit(this);  
    lineEditKm2=new QLineEdit(this);  
    lineEditMilles1=new QLineEdit(this);  
    lineEditMilles2=new QLineEdit(this);  
    labelTitre= new QLabel(this);  
    labelTitre->setText("Convertisseur Kilomètres en Milles");  
    QFont fontTitre( "Arial", 20, QFont::Bold);  
    QPalette pal(0x80ffff); // définition d'une couleur  
    labelTitre->setAutoFillBackground(true);  
    labelTitre->setPalette(pal); // affecte la couleur de font au Label  
    labelTitre->setFont(fontTitre);  
    labelTitre->setAlignment(Qt::AlignHCenter);  
    labelKm=new QLabel(this);  
    labelKm->setText("Kilomètres");  
    labelMilles=new QLabel(this);  
    labelMilles->setText("Milles");  
    labelConv1=new QLabel(this);  
    labelConv1->setText("Km->Milles");  
    labelConv2=new QLabel(this);  
    labelConv2->setText("Km<-Milles");  
    radioKmM=new QRadioButton(this);  
    radioKmM->setText("Km->Milles");  
    radioMKM=new QRadioButton(this);  
    radioMKM->setText("Km<-Milles");  
    QValidator* validator = new QDoubleValidator();  
    lineEditKm1->setValidator(validator);  
    lineEditKm2->setEnabled(false);  
    QValidator* validator2 = new QDoubleValidator();  
    lineEditMilles2->setValidator(validator2);  
    lineEditMilles1->setEnabled(false);  
    lineEditMilles2->setEnabled(false);  
    radioKmM->setChecked(true);  
    this->setWindowTitle("Convertisseur Milles Kilometres");  
    // Définir un layout, sous forme d'une grille  
    layGeneral = new QGridLayout(this);  
    layGeneral->addWidget(labelTitre,1,1,1,4);  
    layGeneral->addWidget(lineEditKm1,3,1);  
    layGeneral->addWidget(lineEditKm2,4,1);  
    layGeneral->addWidget(lineEditMilles1,3,3);  
    layGeneral->addWidget(lineEditMilles2,4,3);  
    layGeneral->addWidget(labelKm,2,1);  
    layGeneral->addWidget(labelMilles,2,3);  
    layGeneral->addWidget(labelConv1,3,2);  
    layGeneral->addWidget(labelConv2,4,2);
```

```
layGeneral->addWidget(radioKmM,3,4);
layGeneral->addWidget(radioMKM,4,4);
connect(lineEditKm1,SIGNAL(editingFinished()),this,SLOT(convertirKmMilles());
connect(lineEditMilles2,SIGNAL(editingFinished()),this,SLOT(convertirMillesKm()
));
connect(radioKmM,SIGNAL(clicked()),this,SLOT(desactiverMilles()));
connect(radioMKM,SIGNAL(clicked()),this,SLOT(desactiverKm()));
}
void ConvKM::convertirKmMilles()
{
    QString txt =lineEditKm1->text();
    double val = txt.toDouble();
    val*= 0.621371;
    lineEditMilles1->setText(QString::number(val));
}
void ConvKM::convertirMillesKm()
{
    //compléter le code en vous inspirant de convertirKmMilles()
    QString txt =lineEditMilles2->text();
    double var = txt.toDouble();
    var/= 0.621371;
    lineEditKm2->setText(QString::number(var));
}
void ConvKM::desactiverMilles()
{
    lineEditKm1->setText("");
    lineEditMilles1->setText("");
    lineEditKm1->setEnabled(true);
    lineEditMilles2->setEnabled(false);
}
void ConvKM::desactiverKm()
{
    //compléter le code en vous inspirant de desactiverMilles()
    lineEditKm2->setText("");
    lineEditMilles2->setText("");
    lineEditKm2->setEnabled(false);
    lineEditKm1->setEnabled(false);
    lineEditMilles2->setEnabled(true);
}
ConvKM::~ConvKM()
{
    delete lineEditKm1;
    delete lineEditKm2;
    delete lineEditMilles1;
    delete lineEditMilles2;
    delete labelTitre;
    delete labelKm;
    delete labelMilles;
    delete labelConv1;
    delete labelConv2;
    delete radioKmM;
    delete radioMKM;
    delete layGeneral;
}
```



7. Exercice complet : Décoder trame GPS : (14 pts)

1 partie 1 et 2

```
#include "tramegps.h"
trameGps::trameGps(QWidget *parent)
    : QWidget(parent)
{
    trameLabel= new QLabel(this) ;
    trameLabel->setText("trame brut:");
    titleLabel= new QLabel(this);
    titleLabel->setText("Analyse de trame GPS");
    QFont fontTitre( "Arial", 20, QFont::Bold);
    QPalette pal(0x80ffff); // définition d'une couleur
    titleLabel->setAutoFillBackground(true);
    titleLabel->setPalette(pal); // affecte la couleur de font au Label
    titleLabel->setFont(fontTitre);
    titleLabel->setAlignment(Qt::AlignHCenter);
    trameLine = new QLineEdit(this);
    getPushButton = new QPushButton(this);
    getPushButton->setText("GetData");
    logitudeLabel = new QLabel(this);
    logitudeLabel->setText("Longitude:");
    altitudeLabel= new QLabel(this);
    altitudeLabel->setText("Altitude:");
    latitudeLabel= new QLabel(this);
    latitudeLabel->setText("Latitude:");
    nbrSateliteLabel= new QLabel(this);
    nbrSateliteLabel->setText("nbr satellite");
    logitudeLine = new QLineEdit(this);
    altitudeLine = new QLineEdit(this);
    latitudeLine = new QLineEdit(this);
    nbrSateliteLine = new QLineEdit(this);
    // Définir un layout, sous forme d'une grille
    layGeneral=new QGridLayout(this);
    layGeneral->addWidget(titleLabel,1,1,1,4);
    layGeneral->addWidget(trameLabel,2,1);
    layGeneral->addWidget(trameLine,2,2,1,3);
    layGeneral->addWidget(getPushButton,3,1);
    layGeneral->addWidget(logitudeLabel,4,1);
    layGeneral->addWidget(logitudeLine,4,2);
    layGeneral->addWidget(altitudeLabel,4,3);
    layGeneral->addWidget(altitudeLine,4,4);
```

```

layGeneral->addWidget(latitudeLabel,5,1);
layGeneral->addWidget(latitudeLine,5,2);
layGeneral->addWidget(nbrSateliteLabel,5,3);
layGeneral->addWidget(nbrSateliteLine,5,4);
//desactivation des champs
logitudeLine->setEnabled(false);
latitudeLine->setEnabled(false);
altitudeLine->setEnabled(false);
nbrSateliteLine->setEnabled(false);
//signal
connect(trameLine,SIGNAL(editingFinished()),this,SLOT(getTrameLine()));
connect(getPushButton,SIGNAL(clicked()),this,SLOT(getData()));
}
//partie 1
void trameGps::getTrameLine(){
    QString trame = trameLine->text();
    traiterTrame(trame);
}
//partie 2
void trameGps::getData(){
    QString trame;
    QFile file("/home/venom/Documents/BTS/C/QT/QT1/NMEA/Trame.txt");
    if (!file.open(QIODevice::ReadOnly)){
        QMessageBox::information(0,"error","impossible de trouver le fichier");
    }
    QTextStream in(&file);
    trame = in.readLine();
    trameLine->setText(trame);
    traiterTrame(trame);
}
//Traite la trame
void trameGps::traiterTrame(QString trame){
    QString trame2;
    int degree;
    double rest;
    QStringList trameArray= trame.split(",");
    int taille=trameArray.size();
    //initialization
    latitudeLine->setText("");
    logitudeLine->setText("");
    altitudeLine->setText("");
    nbrSateliteLine->setText("");
    //Boucle
    for(int i=0;i<taille;i++){
        if(trameArray[i]=='N'){
            ptSplit=trameArray[i-1].split(".");
            degree=ptSplit[0].toInt()/100;
            rest=ptSplit[0].toInt()%100;
            trame2=QString::number(degree)+"° "+QString::number(rest)+","+ptSplit[1];
            latitudeLine->setText(trame2);
        }
        if(trameArray[i]=='E'||trameArray[i]=='W'){
            ptSplit=trameArray[i-1].split(".");
            degree=ptSplit[0].toInt()/100;
            rest=ptSplit[0].toInt()%100;
            trame2=QString::number(degree)+"° "+QString::number(rest)+","+ptSplit[1];
            logitudeLine->setText(trame2);
        }
        if(trameArray[i]=='M'){
            altitudeLine->setText(trameArray[i-1]);
            nbrSateliteLine->setText(trameArray[i-3]);
            //sort de la boucle
            i=taille;
        }
    }
}
trameGps::~trameGps()
{
    delete trameLabel;
    delete titleLabel;
    delete trameLine;
    delete getPushButton;
    delete logitudeLabel;
}

```

```

delete altitudeLabel;
delete latitudeLabel;
delete nbrSateliteLabel;
delete logitudeLine;
delete altitudeLine;
delete latitudeLine;
delete nbrSateliteLine;
delete layGeneral;

```



```

}

```

2 partie 3 et 4

```

#include "tramegps.h"
trameGps::trameGps(QWidget *parent)
    : QWidget(parent)
{
    trameLabel= new QLabel(this) ;
    trameLabel->setText("trame brut:");
    titleLabel= new QLabel(this);
    titleLabel->setText("Analyse de trame GPS");
    QFont fontTitre( "Arial", 20, QFont::Bold);
    QPalette pal(0x80ffff); // définition d'une couleur
    titleLabel->setAutoFillBackground(true);
    titleLabel->setPalette(pal); // affecte la couleur de font au Label
    titleLabel->setFont(fontTitre);
    titleLabel->setAlignment(Qt::AlignHCenter);
    trameLine = new QLineEdit(this);
    getPushButton = new QPushButton(this);
    getPushButton->setText("GetData");
    logitudeLabel = new QLabel(this);
    logitudeLabel->setText("Longitude:");
    altitudeLabel= new QLabel(this);
    altitudeLabel->setText("Altitude:");
    latitudeLabel= new QLabel(this);
    latitudeLabel->setText("Latitude:");
    nbrSateliteLabel= new QLabel(this);
    nbrSateliteLabel->setText("nbr satellite");
    logitudeLine = new QLineEdit(this);
    altitudeLine = new QLineEdit(this);
    latitudeLine = new QLineEdit(this);
    nbrSateliteLine = new QLineEdit(this);
    save= new QPushButton(this);
    save->setText("Sauvegarde");
    // Définir un layout, sous forme d'une grille
    layGeneral=new QGridLayout(this);
    layGeneral->addWidget(titleLabel,1,1,1,4);
    layGeneral->addWidget(trameLabel,2,1);
    layGeneral->addWidget(trameLine,2,2,1,3);
    layGeneral->addWidget(getPushButton,3,1);

```

```

layGeneral->addWidget(logitudeLabel,4,1);
layGeneral->addWidget(logitudeLine,4,2);
layGeneral->addWidget(altitudeLabel,4,3);
layGeneral->addWidget(altitudeLine,4,4);
layGeneral->addWidget(latitudeLabel,5,1);
layGeneral->addWidget(latitudeLine,5,2);
layGeneral->addWidget(nbrSateliteLabel,5,3);
layGeneral->addWidget(nbrSateliteLine,5,4);
layGeneral->addWidget(save,6,4);
//desactivation des champs
logitudeLine->setEnabled(false);
latitudeLine->setEnabled(false);
altitudeLine->setEnabled(false);
nbrSateliteLine->setEnabled(false);
//signal
connect(trameLine,SIGNAL(editingFinished()),this,SLOT(getTrameLine()));
connect(getPushButton,SIGNAL(clicked()),this,SLOT(getData()));
connect(save,SIGNAL(clicked()),this,SLOT(saveData()));
}
//partie 1
void trameGps::getTrameLine(){
    QString trame = trameLine->text();
    traiterTrame(trame);
}
//partie 3
void trameGps::getData(){
    QFileDialog dialog(this);
    QString nomFichier="";
    QString trame;
    dialog.setNameFilter(tr("Text files (*.txt)"));
    if( dialog.exec()== QDialog::Accepted){
        nomFichier=dialog.selectedFiles().first();
        QFile file(nomFichier);
        if (!file.open(QIODevice::ReadOnly)){
            QMessageBox::information(0,"error","impossible de trouver le fichier");
        }
        QTextStream in(&file);
        trame = in.readLine();}
    trameLine->setText(trame);
    traiterTrame(trame);
}
//partie 4
void trameGps::saveData(){
    QFile file("/home/venom/Bureau/DataTrame.csv");
    file.open(QIODevice::WriteOnly|QIODevice::Append);
    QTextStream Cout(&file);
    QString trame= logitudeLine->text()+" "+latitudeLine->text()+" "+altitudeLine->text()+" "+nbrSateliteLine->text();
    Cout<<trame<<endl;
    file.close();
}
//traite la trame
void trameGps::traiterTrame(QString trame){
    QString trame2;
    int degree;
    double rest;
    QStringList trameArray= trame.split(",");
    int taille=trameArray.size();
    //initialization
    latitudeLine->setText("");
    logitudeLine->setText("");
    altitudeLine->setText("");
    nbrSateliteLine->setText("");
    //Boucle
    for(int i=0;i<taille;i++){
        if(trameArray[i]=='N'){
            ptSplit=trameArray[i-1].split(".");
            degree=ptSplit[0].toInt()/100;
            rest=ptSplit[0].toInt()%100;
            trame2=QString::number(degree)+"° "+QString::number(rest)+" "+ptSplit[1];
            latitudeLine->setText(trame2);
        }
        if(trameArray[i]=='E'||trameArray[i]=='W'){
            ptSplit=trameArray[i-1].split(".");

```

```
degree=ptSplit[0].toInt()/100;
rest=ptSplit[0].toInt()%100;
trame2=QString::number(degree)+"° "+QString::number(rest)+". "+ptSplit[1];
logitudeLine->setText(trame2);
}
if(trameArray[i]!='M'){
altitudeLine->setText(trameArray[i-1]);
nbrSateliteLine->setText(trameArray[i-3]);
//sort de la boucle
i=taille;
}
}
}
trameGps::~~trameGps()
{
delete trameLabel;
delete titleLabel;
delete trameLine;
delete getPushButton;
delete logitudeLabel;
delete altitudeLabel;
delete latitudeLabel;
delete nbrSateliteLabel;
delete logitudeLine;
delete altitudeLine;
delete latitudeLine;
delete nbrSateliteLine;
delete layGeneral;
delete save;
}
```

```
accepted){
File
```

```
Rea
on(
```

The screenshot shows a Qt application window titled "trameGps". The window has a dark theme. At the top, there is a cyan header bar with the text "Analyse de trame GPS". Below this, there is a text input field labeled "trame brut:" containing the string "\$GPGGA,230611.016,3907.3813,N,12102.4635,W,C". Below the input field is a button labeled "GetData". Below the button, there are four text input fields arranged in two rows. The first row contains "Longitude:" with the value "121° 2.4635" and "Altitude:" with the value "507.9". The second row contains "Latitude:" with the value "39° 7.3813" and "nbr satellite" with the value "04". At the bottom right of the window is a button labeled "Sauvegarde".

3 Entete .h


```
#ifndef TRAMEGPS_H
#define TRAMEGPS_H
#include <QtWidgets>
#include <QWidget>
class trameGps : public QWidget
{
    Q_OBJECT
private:
    QLabel *trameLabel;
    QLabel *titleLabel;
    QLineEdit *trameLine;
    QPushButton *getPushButton;
    QLabel *logitudeLabel;
    QLabel *altitudeLabel;
    QLabel *latitudeLabel;
    QLabel *nbrSateliteLabel;
    QLineEdit *logitudeLine;
    QLineEdit *altitudeLine;
    QLineEdit *latitudeLine;
    QLineEdit *nbrSateliteLine;
    QGridLayout* layGeneral;
    QPushButton *save;
public slots:
    void getTrameLine();
    void getData();
    void saveData();
public:
    trameGps(QWidget *parent = nullptr);
    void traiterTrame(QString trame);
    ~trameGps();
};
#endif // TRAMEGPS_H
```