ProjetFinal_CDAA_gr_AH

1.0

Généré par Doxygen 1.9.5

1 Index hiérarchique **1**

1.1 Hiérarchie des classes . . . 1

2 Index des classes **3**

2.1 Liste des classes . . . 3

3 Documentation des classes **5**

3.1 Référence de la classe AjoutContact . . . 5

3.1.1 Documentation des constructeurs et destructeur . . . 5

3.1.1.1 AjoutContact() . . . 5

3.1.2 Documentation des fonctions membres . . . 6

3.1.2.1 close() . . . 6

3.1.2.2 initialisationFenetre() . . . 6

3.1.2.3 VerifsChampsContact() . . . 6

3.2 Référence de la classe ajoutContact . . . 7

3.2.1 Description détaillée . . . 7

3.3 Référence de la classe ajoutInteraction . . . 7

3.3.1 Description détaillée . . . 7

3.3.2 Documentation des constructeurs et destructeur . . . 7

3.3.2.1 ajoutInteraction() . . . 7

3.3.3 Documentation des fonctions membres . . . 8

3.3.3.1 close() . . . 8

3.4 Référence du modèle de la structure date::detail::choose_trunc_type< T > . . . 8

3.5 Référence de la classe contact . . . 9

3.5.1 Documentation des fonctions membres . . . 9

3.5.1.1 operator==() . . . 9

3.5.2 Documentation des fonctions amies et associées . . . 10

3.5.2.1 operator<< . . . 10

3.6 Référence de la classe Contact . . . 10

3.6.1 Description détaillée . . . 10

3.7 Référence de la classe date::day . . . 11

3.8 Référence du modèle de la classe date::detail::decimal_format_seconds< Duration > . . . 11

3.9 Référence de la classe detailInteraction . . . 11

3.9.1 Description détaillée . . . 12

3.10 Référence de la classe DetailInteraction . . . 12

3.10.1 Documentation des constructeurs et destructeur . . . 12

3.10.1.1 DetailInteraction() . . . 12

3.10.2 Documentation des fonctions membres . . . 13

3.10.2.1 afficherDonneesInteraction() . . . 13

3.10.2.2 close() . . . 13

3.11 Référence de la classe ficheContact . . . 13

3.11.1 Description détaillée . . . 13

3.12 Référence de la classe FicheContact . . . 14

3.12.1 Documentation des constructeurs et destructeur . . . 14
3.12.1.1 FicheContact() . . . 14
3.12.2 Documentation des fonctions membres . . . 14
3.12.2.1 afficherDonneesContact() . . . 15
3.12.2.2 close() . . . 15
3.12.2.3 verifsChampsContact() . . . 15
3.13 Référence du modèle de la structure date::fields< Duration > . . . 16
3.14 Référence de la classe gestionBDD . . . 16
3.14.1 Description détaillée . . . 17
3.14.2 Documentation des constructeurs et destructeur . . . 17
3.14.2.1 gestionBDD() . . . 17
3.14.3 Documentation des fonctions membres . . . 18
3.14.3.1 ajouterNouveauContact() . . . 18
3.14.3.2 ajouterNouvelleInteraction() . . . 18
3.14.3.3 ajouterNouvelleTache() . . . 19
3.14.3.4 changerDerniereSuppressionContact() . . . 19
3.14.3.5 connexionBDD() . . . 19
3.14.3.6 modifierContact() . . . 20
3.14.3.7 modifierInteraction() . . . 20
3.14.3.8 rechercherContactSelonEntreprise() . . . 21
3.14.3.9 rechercherContactSelonNom() . . . 21
3.14.3.10 recupererIdentifiantInteraction() . . . 22
3.14.3.11 remplirTableauContactsParAlphabetique() . . . 22
3.14.3.12 remplirTableauContactsParDate() . . . 22
3.14.3.13 remplirTableauInteParDate() . . . 22
3.14.3.14 remplirTableauTachesContact() . . . 23
3.14.3.15 remplirTableauTotaliteTaches() . . . 23
3.14.3.16 retournerDerniereSuppressionContact() . . . 23
3.14.3.17 retournerDonneesContact() . . . 24
3.14.3.18 retournerDonneesInteraction() . . . 24
3.14.3.19 retournerIdentiteContact() . . . 24
3.14.3.20 retournerNbTotalContacts() . . . 25
3.14.3.21 supprimerContact() . . . 25
3.14.3.22 supprimerInteraction() . . . 25
3.14.3.23 supprimerTache() . . . 25
3.15 Référence de la classe gestionListeContacts . . . 26
3.15.1 Description détaillée . . . 27
3.15.2 Documentation des constructeurs et destructeur . . . 27
3.15.2.1 gestionListeContacts() . . . 27
3.15.2.2 ~gestionListeContacts() . . . 27
3.15.3 Documentation des fonctions membres . . . 27
3.15.3.1 ajouterContact() . . . 27

3.15.3.2 chercherCParEntreprise() . . . 28
3.15.3.3 chercherCParIntervalleDate() . . . 28
3.15.3.4 chercherCParNom() . . . 28
3.15.3.5 modifierContact() . . . 29
3.15.3.6 trierCParDateCreation() . . . 29
3.15.3.7 trierCParOrdreAlphabetique() . . . 30
3.15.4 Documentation des fonctions amies et associées . . . 30
3.15.4.1 compare_Date . . . 30
3.15.4.2 compare_Nom . . . 31
3.16 Référence de la classe gestionListeInteractions . . . 31
3.16.1 Description détaillée . . . 32
3.16.2 Documentation des constructeurs et destructeur . . . 32
3.16.2.1 ∼gestionListeInteractions() . . . 32
3.16.3 Documentation des fonctions membres . . . 32
3.16.3.1 ajouterInterraction() . . . 32
3.16.3.2 chercherInteParIntervalleDate() . . . 33
3.16.3.3 modifierInterraction() . . . 33
3.16.3.4 supprimerInterraction() . . . 34
3.16.3.5 trierInteParDate() . . . 34
3.16.4 Documentation des fonctions amies et associées . . . 34
3.16.4.1 compare_DateInteractions . . . 34
3.17 Référence de la classe gestionListeTaches . . . 35
3.17.1 Description détaillée . . . 35
3.17.2 Documentation des constructeurs et destructeur . . . 35
3.17.2.1 ∼gestionListeTaches() . . . 36
3.17.3 Documentation des fonctions membres . . . 36
3.17.3.1 ajouterToDo() . . . 36
3.17.3.2 chercherTDParIntervalleDate() . . . 36
3.17.3.3 supprimerToDo() . . . 36
3.17.3.4 trierTDParDate() . . . 37
3.17.4 Documentation des fonctions amies et associées . . . 37
3.17.4.1 compare_DateToDo . . . 37
3.18 Référence du modèle de la classe date::hh_mm_ss< Duration > . . . 38
3.19 Référence de la classe Interraction . . . 38
3.19.1 Description détaillée . . . 38
3.20 Référence de la classe interraction . . . 39
3.20.1 Documentation des fonctions membres . . . 39
3.20.1.1 operator==() . . . 39
3.20.2 Documentation des fonctions amies et associées . . . 40
3.20.2.1 operator<< . . . 40
3.21 Référence de la structure date::last_spec . . . 40
3.22 Référence de la classe listeContacts . . . 40

3.22.1 Documentation des constructeurs et destructeur . . . 41
3.22.1.1 listeContacts() . . . 41
3.22.1.2 ∼listeContacts() . . . 41
3.22.2 Documentation des fonctions membres . . . 41
3.22.2.1 ajouterContact() . . . 41
3.22.2.2 chercherCParEntreprise() . . . 42
3.22.2.3 chercherCParIntervalleDate() . . . 42
3.22.2.4 chercherCParNom() . . . 43
3.22.2.5 modifierContact() . . . 43
3.22.2.6 trierCParDateCreation() . . . 44
3.22.2.7 trierCParOrdreAlphabetique() . . . 44
3.22.3 Documentation des fonctions amies et associées . . . 44
3.22.3.1 compare_Date . . . 45
3.22.3.2 compare_Nom . . . 45
3.23 Référence de la classe ListeContacts . . . 46
3.23.1 Description détaillée . . . 46
3.24 Référence de la classe ListeTaches . . . 46
3.24.1 Description détaillée . . . 47
3.24.2 Documentation des constructeurs et destructeur . . . 47
3.24.2.1 ListeTaches() . . . 47
3.24.3 Documentation des fonctions membres . . . 47
3.24.3.1 afficherIdentiteContact() . . . 48
3.24.3.2 close() . . . 48
3.25 Référence de la classe ListeTotaliteTaches . . . 48
3.25.1 Description détaillée . . . 49
3.25.2 Documentation des constructeurs et destructeur . . . 49
3.25.2.1 ListeTotaliteTaches() . . . 49
3.25.3 Documentation des fonctions membres . . . 49
3.25.3.1 close() . . . 49
3.26 Référence de la structure date::local_t . . . 50
3.27 Référence de la classe date::month . . . 50
3.28 Référence de la classe date::month_day . . . 50
3.29 Référence de la classe date::month_day_last . . . 50
3.30 Référence de la classe date::month_weekday . . . 50
3.31 Référence de la classe date::month_weekday_last . . . 50
3.32 Référence du modèle de la structure date::detail::no_overflow< R1, R2 > . . . 51
3.33 Référence du modèle de la structure date::parse_manip< Parsable, CharT, Traits, Alloc > . . . 51
3.34 Référence de la classe PremiereFenetre . . . 51
3.34.1 Documentation des constructeurs et destructeur . . . 51
3.34.1.1 PremiereFenetre() . . . 52
3.35 Référence de la structure date::detail::rld . . . 52
3.36 Référence de la structure date::detail::rs . . . 52

3.37 Référence de la structure date::detail::ru . . . 52
3.38 Référence du modèle de la classe date::detail::save_istream< CharT, Traits > . . . 53
3.39 Référence du modèle de la classe date::detail::save_ostream< CharT, Traits > . . . 53
3.40 Référence du modèle de la structure date::detail::static_gcd< Xp, Yp > . . . 53
3.41 Référence de la structure date::detail::static_gcd< 0, 0 > . . . 53
3.42 Référence du modèle de la structure date::detail::static_gcd< Xp, 0 > . . . 53
3.43 Référence du modèle de la structure date::detail::static_pow10< exp > . . . 54
3.44 Référence de la structure date::detail::static_pow10< 0 > . . . 54
3.45 Référence du modèle de la classe date::detail::string_literal< CharT, N > . . . 54
3.46 Référence de la classe toDo . . . 54
3.46.1 Documentation des constructeurs et destructeur . . . 54
3.46.1.1 toDo() . . . 54
3.46.2 Documentation des fonctions membres . . . 55
3.46.2.1 operator==() . . . 55
3.46.3 Documentation des fonctions amies et associées . . . 55
3.46.3.1 operator<< . . . 55
3.47 Référence de la classe ToDo . . . 56
3.47.1 Description détaillée . . . 56
3.48 Référence de la structure date::detail::undocumented . . . 56
3.49 Référence de la structure date::detail::unspecified_month_disambiguator . . . 56
3.50 Référence de la classe date::weekday . . . 56
3.51 Référence de la classe date::weekday_indexed . . . 57
3.52 Référence de la classe date::weekday_last . . . 57
3.53 Référence du modèle de la structure date::detail::width< n, d, w, should_continue > . . . 57
3.54 Référence du modèle de la structure date::detail::width< n, d, w, false > . . . 57
3.55 Référence de la classe date::year . . . 58
3.56 Référence de la classe date::year_month . . . 58
3.57 Référence de la classe date::year_month_day . . . 58
3.58 Référence de la classe date::year_month_day_last . . . 59
3.59 Référence de la classe date::year_month_weekday . . . 59
3.60 Référence de la classe date::year_month_weekday_last . . . 59

Index **61**

Chapitre 1

Index hiérarchique

1.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

ajoutContact . . . 7
date::detail::choose_trunc_type< T > . . . 8
contact . . . 9
Contact . . . 10
date::day . . . 11
date::detail::decimal_format_seconds< Duration > . . . 11
date::detail::decimal_format_seconds< typename std::common_type< Duration, std::chrono::seconds >::type > . . . 11
detailInteraction . . . 11
ficheContact . . . 13
date::fields< Duration > . . . 16
gestionBDD . . . 16
gestionListeContacts . . . 26
gestionListeInteractions . . . 31
gestionListeTaches . . . 35
date::hh_mm_ss< Duration > . . . 38
Interraction . . . 38
interraction . . . 39
date::last_spec . . . 40
listeContacts . . . 40
ListeContacts . . . 46
date::local_t . . . 50
date::month . . . 50
date::month_day . . . 50
date::month_day_last . . . 50
date::month_weekday . . . 50
date::month_weekday_last . . . 50
date::detail::no_overflow< R1, R2 > . . . 51
date::parse_manip< Parsable, CharT, Traits, Alloc > . . . 51
QDialog
- AjoutContact . . . 5
- DetailInteraction . . . 12
- ListeTaches . . . 46
- PremiereFenetre . . . 51
- ajoutInteraction . . . 7

QMainWindow
    FicheContact 14
QWidget
    ListeTotaliteTaches 48
date::detail::rld 52
date::detail::rs 52
date::detail::ru 52
date::detail::save_istream< CharT, Traits > 53
    date::detail::save_ostream< CharT, Traits > 53
date::detail::save_istream< CharT, std::char_traits< CharT > > 53
date::detail::static_gcd< Xp, Yp > 53
date::detail::static_gcd< 0, 0 > 53
date::detail::static_gcd< Xp, 0 > 53
date::detail::static_pow10< exp > 54
date::detail::static_pow10< 0 > 54
date::detail::string_literal< CharT, N > 54
toDo 54
ToDo 56
date::detail::undocumented 56
date::detail::unspecified_month_disambiguator 56
date::weekday 56
date::weekday_indexed 57
date::weekday_last 57
date::detail::width< n, d, w, should_continue > 57
date::detail::width< n, d, w, false > 57
date::year 58
date::year_month 58
date::year_month_day 58
date::year_month_day_last 59
date::year_month_weekday 59
date::year_month_weekday_last 59

Chapitre 2

Index des classes

2.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

AjoutContact 5
ajoutContact
Interface concernant la fenêtre de l'ajout de contact 7
ajoutInteraction
Interface concernant la fenêtre de l'ajout d'interaction 7
date::detail::choose_trunc_type< T > 8
contact 9
Contact
Classe définissant les attributs et méthodes des contacts gérés par l'application 10
date::day 11
date::detail::decimal_format_seconds< Duration > 11
detailInteraction
Interface concernant la fenêtre de consultation d'interaction 11
DetailInteraction 12
ficheContact
Interface concernant la fenêtre de consultation de contact 13
FicheContact 14
date::fields< Duration > 16
gestionBDD
Classe définissant les attributs et méthodes liées à la fenêtre de gestion de la base de données 16
gestionListeContacts
Classe définissant les attributs et méthodes permettant de gérer la liste de contacts de l'application 26
gestionListeInteractions
Classe définissant les attributs et méthodes permettant de gérer la liste d'interactions d'un contact 31
gestionListeTaches
Classe définissant les attributs et méthodes permettant de gérer la liste de tâches d'un contact 35
date::hh_mm_ss< Duration > 38
Interraction
Classe définissant les attributs et méthodes des interractions pour chaque contact géré par l'application 38
interraction 39
date::last_spec 40
listeContacts 40

ListeContacts
Classe définissant les attributs et méthodes de la liste de contacts gérée par l'application . . . 46
ListeTaches
Interface concernant la fenêtre listant les tâches d'un contact . . . 46
ListeTotaliteTaches
Interface concernant la fenêtre listant les tâches de tous les contacts . . . 48
date::local_t . . . 50
date::month . . . 50
date::month_day . . . 50
date::month_day_last . . . 50
date::month_weekday . . . 50
date::month_weekday_last . . . 50
date::detail::no_overflow< R1, R2 > . . . 51
date::parse_manip< Parsable, CharT, Traits, Alloc > . . . 51
PremiereFenetre . . . 51
date::detail::rld . . . 52
date::detail::rs . . . 52
date::detail::ru . . . 52
date::detail::save_istream< CharT, Traits > . . . 53
date::detail::save_ostream< CharT, Traits > . . . 53
date::detail::static_gcd< Xp, Yp > . . . 53
date::detail::static_gcd< 0, 0 > . . . 53
date::detail::static_gcd< Xp, 0 > . . . 53
date::detail::static_pow10< exp > . . . 54
date::detail::static_pow10< 0 > . . . 54
date::detail::string_literal< CharT, N > . . . 54
toDo . . . 54
ToDo
Classe définissant les attributs et méthodes des tâches pour chaque contact géré par l'application . . . 56
date::detail::undocumented . . . 56
date::detail::unspecified_month_disambiguator . . . 56
date::weekday . . . 56
date::weekday_indexed . . . 57
date::weekday_last . . . 57
date::detail::width< n, d, w, should_continue > . . . 57
date::detail::width< n, d, w, false > . . . 57
date::year . . . 58
date::year_month . . . 58
date::year_month_day . . . 58
date::year_month_day_last . . . 59
date::year_month_weekday . . . 59
date::year_month_weekday_last . . . 59

Chapitre 3

Documentation des classes

3.1 Référence de la classe AjoutContact

Signaux

— void **fermerFenetre** ()

Fonctions membres publiques

— AjoutContact (QWidget ∗parent=nullptr)
le constructeur de la classe
— ∼**AjoutContact** ()
le destructeur de la classe
— void initialisationFenetre ()
fonction permettant d'initialiser la fenetre avec l'aspect et les données affichées lors de son ouverture
— bool VerifsChampsContact ()
fonction permettant de vérifier si toutes les conditions nécéssaires à la création d'un contact ont bien été remplies
— bool close ()
surcharge de la fonction close() de la fenêtre

3.1.1 Documentation des constructeurs et destructeur

3.1.1.1 AjoutContact()

```
AjoutContact::AjoutContact (
            QWidget * parent = nullptr )  [explicit]
```

le constructeur de la classe

on créé une instance d'interface

Avertissement

toutes les connexions signaux/slots liées a des éléments de l'interface sont effectuées dans le constructeur

3.1.2 Documentation des fonctions membres

3.1.2.1 close()

```
bool AjoutContact::close ( )
```

surcharge de la fonction close() de la fenêtre

on emet le signal fermerFenetre pour actualiser la fenêtre qui va être ouverte à la place

Renvoie

l'état close() de la fenetre

3.1.2.2 initialisationFenetre()

```
void AjoutContact::initialisationFenetre ( )
```

fonction permettant d'initialiser la fenetre avec l'aspect et les données affichées lors de son ouverture

on initialise les dimensions du champ permettant d'afficher la photo du contact

Avertissement

on cache l'url de la photo, afin qu'il ne soit pas visible pour les utilisateurs mais récupérable facilement

3.1.2.3 VerifsChampsContact()

```
bool AjoutContact::VerifsChampsContact ( )
```

fonction permettant de vérifier si toutes les conditions nécéssaires à la création d'un contact ont bien été remplies

Avertissement

on vérifie que tous les champs du contact sont remplis (pas de valeur nulle)
on vérifie que l'on a ajouté une photo pour le contact
on vérifie que l'adresse mail possède un "@"
on vérifie que le numéro de téléphone se compose de 10 chiffres uniquement

Renvoie

un booléen avec la valeur "True" si toutes les conditions sont vérifiées, "False" sinon

3.2 Référence de la classe ajoutContact

Interface concernant la fenêtre de l'ajout de contact.

`#include <ajoutcontact.h>`

3.2.1 Description détaillée

Interface concernant la fenêtre de l'ajout de contact.

Classe définissant les attributs et méthodes liées à la fenêtre d'ajout de contact.

3.3 Référence de la classe ajoutInteraction

Interface concernant la fenêtre de l'ajout d'interaction.

`#include <ajoutinteraction.h>`

Signaux

- void **fermerFenetre** ()
- void **traiterInteraction** (int &)

Fonctions membres publiques

- ajoutInteraction (QWidget *parent=nullptr, QString idContact="")
  le constructeur de la classe
- ~**ajoutInteraction** ()
  le destructeur de la classe
- bool close ()
  surcharge de la fonction close() de la fenêtre

3.3.1 Description détaillée

Interface concernant la fenêtre de l'ajout d'interaction.

Classe définissant les attributs et méthodes liées à la fenêtre d'ajout d'interaction.

3.3.2 Documentation des constructeurs et destructeur

3.3.2.1 ajoutInteraction()

```
ajoutInteraction::ajoutInteraction (
            QWidget * parent = nullptr,
            QString idContact = "" )  [explicit]
```

le constructeur de la classe

Paramètres

idContact	correspond à l'identifiant du contact pour lequel on veut ajouter une interaction

on créé une instance d'interface

Avertissement

toutes les connexions signaux/slots liées a des éléments de l'interface sont effectuées dans le constructeur

3.3.3 Documentation des fonctions membres

3.3.3.1 close()

```
bool ajoutInteraction::close ( )
```

surcharge de la fonction close() de la fenêtre

on emet le signal fermerFenetre pour actualiser la fenêtre qui va être ouverte à la place

Renvoie

l'état close() de la fenetre

3.4 Référence du modèle de la structure date::detail::choose_trunc_type< T >

Types publics

— using **type** = typename std::conditional< digits< 32, std::int32_t, typename std::conditional< digits< 64, std::int64_t, std::int64_t >::type >::type

Attributs publics statiques

— static const int **digits** = std::numeric_limits<T>::digits

3.5 Référence de la classe contact

Fonctions membres publiques

- **contact** (const std::string &, const std::string &, const std::string &, const std::string &, const std::string &, const std::string &)
- ~**contact** ()
  le destructeur de la classe
- std::string **getNomC** () const
- std::string **getPrenomC** () const
- std::string **getEntrepriseC** () const
- std::string **getMailC** () const
- std::string **getTelC** () const
- std::string **getUrlPhotoC** () const
- date::year_month_day **getDateCreationC** () const
- date::year_month_day **getDateModifC** () const
- gestionListeInteractions **getListeInte** () const
- void **setNomC** (const std::string &)
- void **setPrenomC** (const std::string &)
- void **setEntrepriseC** (const std::string &)
- void **setMailC** (const std::string &)
- void **setTelC** (const std::string &)
- void **setUrlPhotoC** (const std::string &)
- void **setDateCreationC** (const date::year_month_day &)
- void **setDateModifC** (const date::year_month_day &)
- void **setListeInte** (const gestionListeInteractions &)
- bool operator== (const contact &c)
  operateur ==
- void **modifierDonneesContact** (const std::string &, const std::string &, const std::string &, const std::string &, const std::string &, const std::string &)
- std::string **conversionDateCreaToString** ()
- std::string **conversionDateModifToString** ()

Amis

- std::ostream & operator<< (std::ostream &, const contact &)
  Fonction amie permettant de prédéfinir l'affichage en console d'un contact.

3.5.1 Documentation des fonctions membres

3.5.1.1 operator==()

```
bool contact::operator== (
            const contact & c )  [inline]
```

operateur ==

Voir également

Opérateur permettant de vérifer si un contact est bien celui que l'on cherche (utilisé notamment dans la fonction supprimerContact() de la classe listeContacts)

Paramètres

c	correspond au contact à vérifier

Renvoie

un booléen qui vaut "True" si le contact est bien celui que l'on cherche, "False" sinon

Avertissement

corps de l'operateur défini directement içi pour réduire le nombre de fonctions implémentées dans le .cpp

3.5.2 Documentation des fonctions amies et associées

3.5.2.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const contact & c )  [friend]
```

Fonction amie permettant de prédéfinir l'affichage en console d'un contact.

Voir également

Utilisée essentiellement pour afficher les attributs des contacts lors des tests dans le main

Paramètres

os	correspond à la sortie console contenant la chaine de caractères
c	correspond au contact que l'on souhaite afficher

Renvoie

la chaîne de caractères affichée dans la console

3.6 Référence de la classe Contact

Classe définissant les attributs et méthodes des contacts gérés par l'application.

```
#include <contact.h>
```

3.6.1 Description détaillée

Classe définissant les attributs et méthodes des contacts gérés par l'application.

3.7 Référence de la classe date::day

Fonctions membres publiques

- CONSTCD11 **day** (unsigned d) NOEXCEPT
- CONSTCD14 day & **operator++** () NOEXCEPT
- CONSTCD14 day **operator++** (int) NOEXCEPT
- CONSTCD14 day & **operator--** () NOEXCEPT
- CONSTCD14 day **operator--** (int) NOEXCEPT
- CONSTCD14 day & **operator+=** (const days &d) NOEXCEPT
- CONSTCD14 day & **operator-=** (const days &d) NOEXCEPT
- CONSTCD11 **operator unsigned** () const NOEXCEPT
- CONSTCD11 bool **ok** () const NOEXCEPT

3.8 Référence du modèle de la classe date::detail::decimal_format_seconds< Duration >

Types publics

- using **precision** = std::chrono::duration< rep, std::ratio< 1, static_pow10< width >::value > >

Fonctions membres publiques

- CONSTCD11 **decimal_format_seconds** (const Duration &d) NOEXCEPT
- CONSTCD14 std::chrono::seconds & **seconds** () NOEXCEPT
- CONSTCD11 std::chrono::seconds **seconds** () const NOEXCEPT
- CONSTCD11 precision **subseconds** () const NOEXCEPT
- CONSTCD14 precision **to_duration** () const NOEXCEPT
- CONSTCD11 bool **in_conventional_range** () const NOEXCEPT
- template<class CharT , class Traits >
  std::basic_ostream< CharT, Traits > & **print** (std::basic_ostream< CharT, Traits > &os, std::true_type) const
- template<class CharT , class Traits >
  std::basic_ostream< CharT, Traits > & **print** (std::basic_ostream< CharT, Traits > &os, std::false_type) const

Attributs publics statiques

- static unsigned CONSTDATA **width** = trial_width < 19 ? trial_width : 6u

Amis

- template<class CharT , class Traits >
  std::basic_ostream< CharT, Traits > & **operator**<< (std::basic_ostream< CharT, Traits > &os, const decimal_format_seconds &x)

3.9 Référence de la classe detailInteraction

Interface concernant la fenêtre de consultation d'interaction.

`#include <detailinteraction.h>`

3.9.1 Description détaillée

Interface concernant la fenêtre de consultation d'interaction.

Classe définissant les attributs et méthodes liées à la fenêtre de consultation d'interaction.

3.10 Référence de la classe DetailInteraction

Signaux

- void **fermerFenetre** ()
- void **traiterInteraction** (int &)

Fonctions membres publiques

- DetailInteraction (QWidget *parent=nullptr, QString idInteraction="")
  le constructeur de la classe
- ~**DetailInteraction** ()
  le destructeur de la classe
- void afficherDonneesInteraction (QSqlQuery &)
  fonction permettant d'afficher les données de l'interaction que l'on consulte dans les champs associés de l'interface
- bool close ()
  surcharge de la fonction close() de la fenêtre

3.10.1 Documentation des constructeurs et destructeur

3.10.1.1 DetailInteraction()

```
DetailInteraction::DetailInteraction (
            QWidget * parent = nullptr,
            QString idInteraction = "" )  [explicit]
```

le constructeur de la classe

Paramètres

idInteraction	correspond à l'identifiant de l'interaction que l'on souhaite consulter

on créé une instance d'interface

Voir également

on appelle la fonction retournerDonneesInteraction() de la classe gestionBDD pour récupérer les données de l'interaction que l'on veut consulter
on appelle la fonction afficherDonneesInteraction() de cette classe pour afficher par défaut les données de l'interaction que l'on consulte

Avertissement

toutes les connexions signaux/slots liées a des éléments de l'interface sont effectuées dans le constructeur

3.10.2 Documentation des fonctions membres

3.10.2.1 afficherDonneesInteraction()

```
void DetailInteraction::afficherDonneesInteraction (
            QSqlQuery & qu )
```

fonction permettant d'afficher les données de l'interaction que l'on consulte dans les champs associés de l'interface

Paramètres

qu	correspond au résultat de la requête retournant les données de l'interaction à consulter

3.10.2.2 close()

```
bool DetailInteraction::close ( )
```

surcharge de la fonction close() de la fenêtre

on emet le signal fermerFenetre pour actualiser la fenêtre qui va être ouverte à la place

Renvoie

l'état close() de la fenetre

3.11 Référence de la classe ficheContact

Interface concernant la fenêtre de consultation de contact.

```
#include <fichecontact.h>
```

3.11.1 Description détaillée

Interface concernant la fenêtre de consultation de contact.

Classe définissant les attributs et méthodes liées à la fenêtre de consultation de contact.

3.12 Référence de la classe FicheContact

Signaux

- void **fermerFenetre** ()
- void **actualiserApresSupplnte** ()

Fonctions membres publiques

- FicheContact (QWidget *parent=nullptr, QString idContact="")
  le constructeur de la classe
- ~**FicheContact** ()
  le destructeur de la classe
- void afficherDonneesContact (QSqlQuery &)
  fonction permettant d'afficher les données du contact que l'on consulte dans les champs associés de l'interface
- void **afficherListeInteractions** (QSqlQuery &)
- bool verifsChampsContact ()
  fonction permettant de vérifier si toutes les conditions nécéssaires à la mise à jour d'un contact ont bien été remplies
- bool close ()
  surcharge de la fonction close() de la fenêtre

3.12.1 Documentation des constructeurs et destructeur

3.12.1.1 FicheContact()

```
FicheContact::FicheContact (
            QWidget * parent = nullptr,
            QString idContact = "" )
```

le constructeur de la classe

Paramètres

idContact	correspond à l'identifiant du contact que l'on souhaite consulter

on créé une instance d'interface

Voir également

on appelle la fonction actualiserFenetre() de cette classe pour afficher par défaut les données du contact que l'on consulte

Avertissement

toutes les connexions signaux/slots liées a des éléments de l'interface sont effectuées dans le constructeur

3.12.2 Documentation des fonctions membres

3.12.2.1 afficherDonneesContact()

```
void FicheContact::afficherDonneesContact (
            QSqlQuery & qu )
```

fonction permettant d'afficher les données du contact que l'on consulte dans les champs associés de l'interface

Paramètres

qu	correspond au résultat de la requête retournant les données du contact à consulter

on attribue à chaque champ de l'interface une donnée récupérée grâce à la requete passée en paramètres

Avertissement

si l'application ne parvient pas a retrouver la photo du contact à partir de son url, elle considère que le contact n'en a pas (son url de photo est indiqué comme vide)

3.12.2.2 close()

```
bool FicheContact::close ( )
```

surcharge de la fonction close() de la fenêtre

on emet le signal fermerFenetre pour actualiser la fenêtre qui va être ouverte à la place

Renvoie

l'état close() de la fenetre

3.12.2.3 verifsChampsContact()

```
bool FicheContact::verifsChampsContact ( )
```

fonction permettant de vérifier si toutes les conditions nécéssaires à la mise à jour d'un contact ont bien été remplies

Avertissement

on vérifie que tous les champs du contact sont remplis (pas de valeur nulle)
on vérifie que le contact possède bien une photo
on vérifie que l'adresse mail possède un "@"
on vérifie que le numéro de téléphone se compose de 10 chiffres uniquement

Renvoie

un booléen avec la valeur "True" si toutes les conditions sont vérifiées, "False" sinon

3.13 Référence du modèle de la structure date::fields< Duration >

Fonctions membres publiques

- **fields** (year_month_day ymd_)
- **fields** (weekday wd_)
- **fields** (hh_mm_ss< Duration > tod_)
- **fields** (year_month_day ymd_, weekday wd_)
- **fields** (year_month_day ymd_, hh_mm_ss< Duration > tod_)
- **fields** (weekday wd_, hh_mm_ss< Duration > tod_)
- **fields** (year_month_day ymd_, weekday wd_, hh_mm_ss< Duration > tod_)

Attributs publics

- year_month_day **ymd** {nanyear/0/0}
- weekday **wd** {8u}
- hh_mm_ss< Duration > **tod** {}
- bool **has_tod** = false

3.14 Référence de la classe gestionBDD

Classe définissant les attributs et méthodes liées à la fenêtre de gestion de la base de données.

`#include <gestionbdd.h>`

Fonctions membres publiques

- gestionBDD (QString &)
  le constructeur de la classe
- ~**gestionBDD** ()
  le destructeur de la classe
- void connexionBDD (QString &)
  fonction permettant d'établir une connexion avec la base de données
- void **fermerConnexion** ()
  fonction permettant de fermer la connexion à la base de données
- QString retournerNbTotalContacts ()
  fonction permettant d'exécuter une requête qui retourne le nombre total de contacts stockés dans la base de données
- QString retournerDerniereSuppressionContact ()
  fonction permettant d'exécuter une requête qui retourne la dernière date de suppression de contacts stockée dans la base de données
- QSqlQuery remplirTableauContactsParDate ()
  fonction permettant d'exécuter une requête qui retourne la liste des contacts stockés dans la base de données, ordonnés par date de création
- QSqlQuery remplirTableauContactsParAlphabetique ()
  fonction permettant d'exécuter une requête qui retourne la liste des contacts stockés dans la base de données, ordonnés par ordre alphabétique
- QSqlQuery rechercherContactSelonNom (const QString &, bool &)
  fonction permettant d'exécuter une requête qui retourne la liste des contacts stockés dans la base de données dont le nom correspond à une certaine recherche passée en paramètres
- QSqlQuery rechercherContactSelonEntreprise (const QString &, bool &)
  fonction permettant d'exécuter une requête qui retourne la liste des contacts stockés dans la base de données dont l'entreprise correspond à une certaine recherche passée en paramètres
- void ajouterNouveauContact (const QString &, const QString &, const QString &, const QString &, const QString &, const QString &, const QString &, const QString &)

fonction permettant d'exécuter une requête qui ajoute un nouveau contact dans la base de données

— QSqlQuery retournerDonneesContact (const QString &)

fonction d'exécuter une requête qui retourne les données d'un seul contact identifié par son identifiant passé en paramètres

— void modifierContact (const QString &, const QString &, const QString &, const QString &, const QString &, const QString &, const QString &)

fonction permettant d'exécuter une requête qui met à jour les données d'un contact identifié par son identifiant passé en paramètres

— void supprimerContact (const QString &)

fonction permettant d'exécuter une requête qui supprime un contact dans la base de données

— void changerDerniereSuppressionContact (const QString &)

fonction permettant d'exécuter une requête qui met à jour la date de dernière suppression de contact dans la base de données

— QSqlQuery remplirTableauInteParDate (const QString &)

fonction permettant d'exécuter une requête qui retourne la liste des interraction d'un contact stockées dans la base de données, ordonnées par identifiant descendant

— void supprimerInteraction (const QString &)

fonction permettant d'exécuter une requête qui supprime une interaction dans la base de données

— void ajouterNouvelleInteraction (const QString &, const QString &, const QString &, const int &)

fonction permettant d'exécuter une requête qui ajoute une nouvelle interaction dans la base de données

— int recupererIdentifiantInteraction ()

fonction permettant d'exécuter une requete qui retourne l'identifiant de la dernière interaction ajoutée dans la base de données

— void ajouterNouvelleTache (const QString &, const QString &, const int &)

fonction permettant d'exécuter une requête qui ajoute une nouvelle tâche dans la base de données

— QSqlQuery retournerDonneesInteraction (const QString &)

fonction permettant d'exécuter une requête qui retourne les données d'une seule interaction identifié par son identifiant passé en paramètres

— void modifierInteraction (const QString &, const QString &, const QString &)

fonction permettant d'exécuter une requête qui met à jour les données d'une interaction identifiée par son identifiant passé en paramètres

— QSqlQuery retournerIdentiteContact (const QString &)

fonction permettant d'exécuter une requête qui retourne l'identité (prénom + nom) d'un contact identifié par l'identifiant passé en paramètres

— QSqlQuery remplirTableauTachesContact (const QString &)

fonction permettant d'exécuter une requête qui retourne la liste des tâches d'un contact stockées dans la base de données

— QSqlQuery remplirTableauTotaliteTaches ()

fonction permettant d'exécuter une requête qui retourne la liste de l'ensemble des tâches stockées dans la base de données

— void supprimerTache (const QString &)

fonction permettant de supprimer une tâche dans la base de données

3.14.1 Description détaillée

Classe définissant les attributs et méthodes liées à la fenêtre de gestion de la base de données.

3.14.2 Documentation des constructeurs et destructeur

3.14.2.1 gestionBDD()

```
gestionBDD::gestionBDD (
            QString & chemin )
```

le constructeur de la classe

Paramètres

chemin	correspond au chemin absolu du fichier de configuration de base de données choisi par l'utilisateur

Voir également

on appelle la fonction connexionBDD() pour établir une connexion avec la base de données à l'aide du fichier de configuration passé en paramètres

3.14.3 Documentation des fonctions membres

3.14.3.1 ajouterNouveauContact()

```
void gestionBDD::ajouterNouveauContact (
            const QString & n,
            const QString & p,
            const QString & e,
            const QString & m,
            const QString & t,
            const QString & u,
            const QString & dc,
            const QString & dm )
```

fonction permettant d'exécuter une requête qui ajoute un nouveau contact dans la base de données

Paramètres

n	correspond au nom du nouveau contact
p	correspond au prénom du nouveau contact
e	correspond à l'entreprise du nouveau contact
m	correspond à l'adresse-mail du nouveau contact
t	correspond au numéro de téléphone du nouveau contact
u	correspond à l'url de la photo du nouveau contact
dc	correspond à la date de création du nouveau contact
dm	correspond à la date de modification du nouveau contact

3.14.3.2 ajouterNouvelleInteraction()

```
void gestionBDD::ajouterNouvelleInteraction (
            const QString & t,
            const QString & c,
            const QString & dm,
            const int & idc )
```

fonction permettant d'exécuter une requête qui ajoute une nouvelle interaction dans la base de données

Paramètres

t	correspond au titre de la nouvelle interaction
c	correspond au contenu de la nouvelle interaction
dm	correspond à la date de modification de la nouvelle interaction
idc	correspond à l'identifiant du contact associé à la nouvelle interraction

3.14.3.3 ajouterNouvelleTache()

```
void gestionBDD::ajouterNouvelleTache (
            const QString & c,
            const QString & dr,
            const int & idi )
```

fonction permettant d'exécuter une requête qui ajoute une nouvelle tâche dans la base de données

Paramètres

c	correspond au contenu de la nouvelle tâche
dr	correspond à la date de réalisation de la nouvelle tâche
idi	correspond à l'identifiant de l'interaction associée à la nouvelle tâche

3.14.3.4 changerDerniereSuppressionContact()

```
void gestionBDD::changerDerniereSuppressionContact (
            const QString & nvdate )
```

fonction permettant d'exécuter une requête qui met à jour la date de dernière suppression de contact dans la base de données

Paramètres

nvdate	correspond à la nouvelle date de dernière suppression

3.14.3.5 connexionBDD()

```
void gestionBDD::connexionBDD (
            QString & ch )
```

fonction permettant d'établir une connexion avec la base de données

Paramètres

ch	corrspond au chemin absolu permettant de récupérer la base de données sélectionnée par l'utilisateur

Avertissement

si la connexion n'arrive pas à s'établir correctement, le programme s'arrête de lui même pour éviter les problèmes de connexion ensuite

3.14.3.6 modifierContact()

```
void gestionBDD::modifierContact (
            const QString & id,
            const QString & n,
            const QString & p,
            const QString & e,
            const QString & t,
            const QString & m,
            const QString & u )
```

fonction permettant d'exécuter une requête qui met à jour les données d'un contact identifié par son identifiant passé en paramètres

Paramètres

id	correspond à l'identifiant du contact qu'on veut mettre à jour
n	correspond au nouveau nom du contact
p	correspond au nouveau prénom du contact
e	correspond à la nouvelle entreprise du contact
m	correspond à la nouvelle adresse-mail du contact
t	correspond au nouveau numéro de téléphone du contact
u	correspond à la nouvelle url de la photo du contact

on met à jour automatiquement la date de modification du contact avec la date du jour

Avertissement

on fait en sorte que la date de modification du contact soit toujours de la forme "dd/mm/yyyy" en rajoutant des 0 devant le jour ou le mois si nécéssaire

3.14.3.7 modifierInteraction()

```
void gestionBDD::modifierInteraction (
            const QString & id,
            const QString & t,
            const QString & c )
```

fonction permettant d'exécuter une requête qui met à jour les données d'une interaction identifiée par son identifiant passé en paramètres

Paramètres

id	correspond à l'identifiant de l'interaction qu'on veut mettre à jour
t	correspond au nouveau titre de l'interaction
c	correspond au nouveau contenu de l'interaction

3.14.3.8 rechercherContactSelonEntreprise()

```
QSqlQuery gestionBDD::rechercherContactSelonEntreprise (
            const QString & rech,
            bool & alphab )
```

fonction permettant d'exécuter une requête qui retourne la liste des contacts stockés dans la base de données dont l'entreprise correspond à une certaine recherche passée en paramètres

Paramètres

rech	correspond à la recherche textuelle effectuée par l'utilisateur
alphab	correpond au choix du type de tri : true si c'est un tri par ordre alphabétique et false si c'est pas date de création

Avertissement

la requête renvoyant les contacts est adaptée selon le mode de tri choisi par l'utilisateur et passé en paramètres

Renvoie

le résultat d'une requête qui contient la liste des contacts dont l'entreprise correspond à une certaine recherche passée en paramètres

3.14.3.9 rechercherContactSelonNom()

```
QSqlQuery gestionBDD::rechercherContactSelonNom (
            const QString & rech,
            bool & alphab )
```

fonction permettant d'exécuter une requête qui retourne la liste des contacts stockés dans la base de données dont le nom correspond à une certaine recherche passée en paramètres

Paramètres

rech	correspond à la recherche textuelle effectuée par l'utilisateur
alphab	correpond au choix du type de tri : true si c'est un tri par ordre alphabétique et false si c'est pas date de création

Avertissement

la requête renvoyant les contacts est adaptée selon le mode de tri choisi par l'utilisateur et passé en paramètres

Renvoie

le résultat d'une requête qui contient la liste des contacts dont le nom correspond à une certaine recherche passée en paramètres

3.14.3.10 recupererIdentifiantInteraction()

```
int gestionBDD::recupererIdentifiantInteraction ( )
```

fonction permettant d'exécuter une requete qui retourne l'identifiant de la dernière interaction ajoutée dans la base de données

Renvoie

l'identifiant de la dernière interaction ajoutée dans la base de données

3.14.3.11 remplirTableauContactsParAlphabetique()

```
QSqlQuery gestionBDD::remplirTableauContactsParAlphabetique ( )
```

fonction permettant d'exécuter une requête qui retourne la liste des contacts stockés dans la base de données, ordonnés par ordre alphabétique

Renvoie

le résultat d'une requête qui contient la liste des contacts ordonnés par ordre alphabétique

3.14.3.12 remplirTableauContactsParDate()

```
QSqlQuery gestionBDD::remplirTableauContactsParDate ( )
```

fonction permettant d'exécuter une requête qui retourne la liste des contacts stockés dans la base de données, ordonnés par date de création

Renvoie

le résultat d'une requête qui contient la liste des contacts ordonnés par date de création

3.14.3.13 remplirTableauInteParDate()

```
QSqlQuery gestionBDD::remplirTableauInteParDate (
            const QString & id )
```

fonction permettant d'exécuter une requête qui retourne la liste des interraction d'un contact stockées dans la base de données, ordonnées par identifiant descendant

Paramètres

id	correspond à l'identifiant du contact dont on souhaite récupérer la liste des interactions

Renvoie

le résultat d'une requête qui contient la liste des interraction d'un contact ordonnées par identifiant descendant

3.14.3.14 remplirTableauTachesContact()

```
QSqlQuery gestionBDD::remplirTableauTachesContact (
            const QString & idc )
```

fonction permettant d'exécuter une requête qui retourne la liste des tâches d'un contact stockées dans la base de données

Paramètres

id	correspond à l'identifiant du contact dont on souhaite obtenir la liste des tâches

Renvoie

le résultat d'une requête qui contient la liste des tâches d'un contact

3.14.3.15 remplirTableauTotaliteTaches()

```
QSqlQuery gestionBDD::remplirTableauTotaliteTaches ( )
```

fonction permettant d'exécuter une requête qui retourne la liste de l'ensemble des tâches stockées dans la base de données

Renvoie

le résultat d'une requête qui contient la liste de l'ensemble des tâches

3.14.3.16 retournerDerniereSuppressionContact()

```
QString gestionBDD::retournerDerniereSuppressionContact ( )
```

fonction permettant d'exécuter une requête qui retourne la dernière date de suppression de contacts stockée dans la base de données

Renvoie

la dernière date de suppression de contacts dans la base de données

3.14.3.17 retournerDonneesContact()

```
QSqlQuery gestionBDD::retournerDonneesContact (
            const QString & id )
```

fonction d'exécuter une requête qui retourne les données d'un seul contact identifié par son identifiant passé en paramètres

Paramètres

id	correspond à l'identifiant du contact dont on souhaite récupérer les données

Renvoie

le résultat d'une requête qui contient les données du contact qui possède l'identifiant passé en paramètres

3.14.3.18 retournerDonneesInteraction()

```
QSqlQuery gestionBDD::retournerDonneesInteraction (
            const QString & id )
```

fonction permettant d'exécuter une requête qui retourne les données d'une seule interaction identifié par son identifiant passé en paramètres

Paramètres

id	correspond à l'identifiant de l'interraction dont on souhaite récupérer les données

Renvoie

le résultat d'une requête qui contient les données de l'interaction qui possède l'identifiant passé en paramètres

3.14.3.19 retournerIdentiteContact()

```
QSqlQuery gestionBDD::retournerIdentiteContact (
            const QString & idc )
```

fonction permettant d'exécuter une requête qui retourne l'identité (prénom + nom) d'un contact identifié par l'identifiant passé en paramètres

Paramètres

idc	correspond à l'identifiant du contact dont on souhaite connaître l'identité

Renvoie

l'identité du contact identifié par l'identifiant passé en paramètres

3.14.3.20 retournerNbTotalContacts()

```
QString gestionBDD::retournerNbTotalContacts ( )
```

fonction permettant d'exécuter une requête qui retourne le nombre total de contacts stockés dans la base de données

Renvoie

le nombre total de contacts stockés dans la base de données

3.14.3.21 supprimerContact()

```
void gestionBDD::supprimerContact (
            const QString & id )
```

fonction permettant d'exécuter une requête qui supprime un contact dans la base de données

Paramètres

id	correspond à l'identifiant du contact que l'on souhaite supprimer

3.14.3.22 supprimerInteraction()

```
void gestionBDD::supprimerInteraction (
            const QString & id )
```

fonction permettant d'exécuter une requête qui supprime une interaction dans la base de données

Paramètres

id	correspond à l'identifiant de l'interaction que l'on souhaite supprimer

3.14.3.23 supprimerTache()

```
void gestionBDD::supprimerTache (
            const QString & idt )
```

fonction permettant de supprimer une tâche dans la base de données

Paramètres

idt	correspond à l'identifiant de la tâche que l'on souhaite supprimer

3.15 Référence de la classe gestionListeContacts

Classe définissant les attributs et méthodes permettant de gérer la liste de contacts de l'application.

`#include <gestionlistecontacts.h>`

Fonctions membres publiques

- gestionListeContacts ()
  le constructeur de la classe
- ~gestionListeContacts ()
  le destructeur de la classe
- std::list< contact > **getListeC** () const
- unsigned int **getNbTotal** () const
- date::year_month_day **getDateDerniereSuppression** () const
- void **setNbTotal** (const unsigned int &)
- void **setDateDerniereSuppression** (const date::year_month_day &)
- void ajouterContact (const contact &)
  Fonction permettant d'ajouter un nouveau contact dans la liste.
- void **supprimerContact** (const contact &)
- void modifierContact (const contact &, const std::string &, const std::string &, const std::string &, const std::string &, const std::string &)
  Fonction permettant de modifier un contact dans la liste.
- void trierCParOrdreAlphabetique ()
  Fonction permettant de trier la liste de contacts par ordre alphabétique.
- void trierCParDateCreation ()
  Fonction permettant de trier la liste de contacts par date de création.
- std::list< contact > chercherCParNom (std::string &)
  Fonction permettant de chercher uniquement les contacts de la liste dont le nom commence de la même facon que la recherche passée en paramètres.
- std::list< contact > chercherCParEntreprise (std::string &)
  Fonction permettant de chercher uniquement les contacts de la liste dont l'entreprise commence de la même facon que la recherche passée en paramètres.
- std::list< contact > chercherCParIntervalleDate (const date::year_month_day &, const date::year_month_day &)
  Fonction permettant de chercher uniquement les contacts de la liste dont la date de création est comprise entre les 2 dates passées en paramètres.

Amis

- bool compare_Nom (const contact &, const contact &)
  Fonction amie permettant de comparer les noms et prénoms de 2 contacts, afin de déterminer lequel est le premier par ordre alphabétique.
- bool compare_Date (const contact &, const contact &)
  Fonction amie permettant de comparer les dates de création de 2 contacts, afin de déterminer laquelle est la plus récente et laquelle est la plus ancienne.

3.15.1 Description détaillée

Classe définissant les attributs et méthodes permettant de gérer la liste de contacts de l'application.

3.15.2 Documentation des constructeurs et destructeur

3.15.2.1 gestionListeContacts()

```
gestionListeContacts::gestionListeContacts ( )
```

le constructeur de la classe

initialise le nombre total de contacts de la liste à 0 lors de la création

3.15.2.2 ~gestionListeContacts()

```
gestionListeContacts::~gestionListeContacts ( )
```

le destructeur de la classe

Si la liste de contact est supprimée, son contenu est nettoyé de ses données pour gagner de la place en mémoire

3.15.3 Documentation des fonctions membres

3.15.3.1 ajouterContact()

```
void gestionListeContacts::ajouterContact (
            const contact & c )
```

Fonction permettant d'ajouter un nouveau contact dans la liste.

Paramètres

c	correspond au nouveau contact à ajouter

Avertissement

si le contact est ajouté, on ajoute 1 au nombre total de contacts de la liste

3.15.3.2 chercherCParEntreprise()

```
std::list< contact > gestionListeContacts::chercherCParEntreprise (
            std::string & recherche )
```

Fonction permettant de chercher uniquement les contacts de la liste dont l'entreprise commence de la même facon que la recherche passée en paramètres.

Paramètres

recherche	correspond à la chaine que l'on recherche dans l'entreprise d'un contact

Renvoie

une liste de contacts contenant uniquement les contacts qui remplissent la condition

Avertissement

si la recherche de l'utilisateur est vide, la fonction retourne tous les contacts de la liste

on convertit la recherche et l'entreprise de chaque contact afin de ne pas devoir respecter la casse lors de la comparaison

3.15.3.3 chercherCParIntervalleDate()

```
std::list< contact > gestionListeContacts::chercherCParIntervalleDate (
            const date::year_month_day & d1,
            const date::year_month_day & d2 )
```

Fonction permettant de chercher uniquement les contacts de la liste dont la date de création est comprise entre les 2 dates passées en paramètres.

Paramètres

d1	correspond à la date la plus ancienne de l'intervalle
d2	correspond à la date la plus récente de l'intervalle

Renvoie

une liste de contacts contenant uniquement les contacts qui remplissent la condition

3.15.3.4 chercherCParNom()

```
std::list< contact > gestionListeContacts::chercherCParNom (
            std::string & recherche )
```

Fonction permettant de chercher uniquement les contacts de la liste dont le nom commence de la même facon que la recherche passée en paramètres.

Paramètres

recherche	correspond à la chaine que l'on recherche dans un nom de contact

Renvoie

une liste de contacts contenant uniquement les contacts qui remplissent la condition

Avertissement

si la recherche de l'utilisateur est vide, la fonction retourne tous les contacts de la liste

on convertit la recherche et le nom de chaque contact afin de ne pas devoir respecter la casse lors de la comparaison

3.15.3.5 modifierContact()

```
void gestionListeContacts::modifierContact (
            const contact & c,
            const std::string & n,
            const std::string & p,
            const std::string & e,
            const std::string & m,
            const std::string & t,
            const std::string & ph )
```

Fonction permettant de modifier un contact dans la liste.

Paramètres

c	correspond au contact dont on souhaite modifier les données
n	correspond au nouveau nom du contact
p	correspond au nouveau prénom du contact
e	correspond à la nouvelle entreprise du contact
m	correspond à la nouvelle adresse-mail du contact
t	correspond au nouveau numéro de téléphone du contact
ph	correspond à l'URL de la nouvelle photo

Avertissement

on réalise une copie du contact à modifier avant de le supprimer de la liste, puis on effectue les modifications sur la copie avant de la réajouter à la liste

3.15.3.6 trierCParDateCreation()

```
void gestionListeContacts::trierCParDateCreation ( )
```

Fonction permettant de trier la liste de contacts par date de création.

Voir également

utilise la fonction amie compare_Date comme paramètre de la fonction sort() pour établir la comparaison entre 2 contacts

3.15.3.7 trierCParOrdreAlphabetique()

```
void gestionListeContacts::trierCParOrdreAlphabetique ( )
```

Fonction permettant de trier la liste de contacts par ordre alphabétique.

Voir également

utilise la fonction amie compare_Nom comme paramètre de la fonction sort() pour établir la comparaison entre 2 contacts

3.15.4 Documentation des fonctions amies et associées

3.15.4.1 compare_Date

```
bool compare_Date (
            const contact & c1,
            const contact & c2 )  [friend]
```

Fonction amie permettant de comparer les dates de création de 2 contacts, afin de déterminer laquelle est la plus récente et laquelle est la plus ancienne.

Voir également

utilisée comme paramètre dans la fonction trierCParDateCreation() de cette classe

Paramètres

c1	correspond au premier contact dont on souhaite comparer la date de création
c2	correspond au deuxième contact dont on souhaite comparer la date de création

Renvoie

un booléen avec la valeur "True" si le premier contact possède la date la plus ancienne, "False" si c'est le deuxième

Avertissement

si les dates sont identiques, la date du premier contact est considérée la plus ancienne (partie "else" de la condition)

3.15.4.2 compare_Nom

```
bool compare_Nom (
            const contact & c1,
            const contact & c2 )  [friend]
```

Fonction amie permettant de comparer les noms et prénoms de 2 contacts, afin de déterminer lequel est le premier par ordre alphabétique.

Voir également

utilisée comme paramètre dans la fonction trierCParOrdreAlphabetique de cette classe

Paramètres

c1	correspond au premier contact dont on souhaite comparer le nom
c2	correspond au deuxième contact dont on souhaite comparer le nom

Renvoie

un booléen avec la valeur "True" si le premier contact est situé avant dans l'ordre alphabétique, "False" si c'est le deuxième

Avertissement

si les noms sont identiques, on compare alors les prénoms

si les noms et les prénoms sont identiques, le premier contact est considéré comme le premier par ordre alphabétique

Voir également

utilisée comme paramètre dans la fonction trierCParOrdreAlphabetique de cette classe

Paramètres

c1	correspond au premier contact dont on souhaite comparer le nom
c2	correspond au deuxième contact dont on souhaite comparer le nom

Renvoie

un booléen avec la valeur "True" si le premier contact est situé avant dans l'ordre alphabétique, "False" si c'est le deuxième

Avertissement

si les noms sont identiques, on compare alors les prénoms

si les noms et les prénoms sont identiques, le premier contact est considéré être le premier par ordre alphabétique

3.16 Référence de la classe gestionListeInteractions

Classe définissant les attributs et méthodes permettant de gérer la liste d'interactions d'un contact.

`#include <gestionlisteinteractions.h>`

Fonctions membres publiques

- **gestionListeInteractions** ()
  le constructeur de la classe
- ~gestionListeInteractions ()
  le destructeur de la classe
- std::list< interraction > **getListeInterractions** () const
- void ajouterInterraction (interraction &)
  Fonction permettant d'ajouter une nouvelle interaction dans la liste.
- void supprimerInterraction (const interraction &)
  Fonction permettant de supprimer une interaction dans la liste.
- void modifierInterraction (const interraction &, const std::string &, const std::string &)
  Fonction permettant de modifier une interaction dans la liste.
- void trierInteParDate ()
  Fonction permettant de trier la liste des interactions par date de modification.
- std::list< interraction > chercherInteParIntervalleDate (const date::year_month_day &, const date::year_month_day &)
  Fonction permettant de chercher uniquement les interactions de la liste dont la date de modification est comprise entre les 2 dates passées en paramètres.

Amis

- bool compare_DateInteractions (const interraction &, const interraction &)
  Fonction amie permettant de comparer les dates de modification de 2 interactions, afin de déterminer laquelle est la plus récente et laquelle est la plus ancienne.

3.16.1 Description détaillée

Classe définissant les attributs et méthodes permettant de gérer la liste d'interactions d'un contact.

3.16.2 Documentation des constructeurs et destructeur

3.16.2.1 ~gestionListeInteractions()

```
gestionListeInteractions::~gestionListeInteractions ( )
```

le destructeur de la classe

Si la liste d'interactions est supprimée, son contenu est nettoyé de ses données pour gagner de la place en mémoire

3.16.3 Documentation des fonctions membres

3.16.3.1 ajouterInterraction()

```
void gestionListeInteractions::ajouterInterraction (
            interraction & inte )
```

Fonction permettant d'ajouter une nouvelle interaction dans la liste.

Paramètres

inte	correspond à la nouvelle interaction à ajouter

Voir également

la fonction traiterInterraction() est appelée dans cette fonction, afin de parcourir la nouvelle interaction et d'y extraire possiblement des tâches à ajouter dans la liste de tâches du contact

3.16.3.2 chercherInteParIntervalleDate()

```
std::list< interraction > gestionListeInteractions::chercherInteParIntervalleDate (
            const date::year_month_day & d1,
            const date::year_month_day & d2 )
```

Fonction permettant de chercher uniquement les interactions de la liste dont la date de modification est comprise entre les 2 dates passées en paramètres.

Paramètres

d1	correspond à la date la plus ancienne de l'intervalle
d2	correspond à la date la plus récente de l'intervalle

Renvoie

une liste d'interaction contenant uniquement les interactions qui remplissent la condition

3.16.3.3 modifierInterraction()

```
void gestionListeInteractions::modifierInterraction (
            const interraction & i,
            const std::string & ti,
            const std::string & ci )
```

Fonction permettant de modifier une interaction dans la liste.

Paramètres

i	correspond à l'interaction dont on souhaite modifier les données
ti	correspond au nouveau titre de l'interaction
ci	correspond au nouveau contenu de l'interaction

Avertissement

on réalise une copie de l'interaction à modifier avant de la supprimer de la liste, puis on effectue les modifications sur la copie avant de la réajouter à la liste

3.16.3.4 supprimerInterraction()

```
void gestionListeInteractions::supprimerInterraction (
            const interraction & inte )
```

Fonction permettant de supprimer une interaction dans la liste.

Paramètres

inte	correspond à l'interaction à supprimer de la liste

3.16.3.5 trierInteParDate()

```
void gestionListeInteractions::trierInteParDate ( )
```

Fonction permettant de trier la liste des interactions par date de modification.

Voir également

utilise la fonction amie compare_DateInte comme paramètre de la fonction sort() pour établir la comparaison entre 2 interactions

3.16.4 Documentation des fonctions amies et associées

3.16.4.1 compare_DateInteractions

```
bool compare_DateInteractions (
            const interraction & i1,
            const interraction & i2 )  [friend]
```

Fonction amie permettant de comparer les dates de modification de 2 interactions, afin de déterminer laquelle est la plus récente et laquelle est la plus ancienne.

Voir également

utilisée comme paramètre dans la fonction trierInteParDate() de cette classe

Paramètres

i1	correspond à la première interaction dont on souhaite comparer la date de modification
i2	correspond à la deuxième interaction dont on souhaite comparer la date de modification

Renvoie

un booléen avec la valeur "True" si la première interraction possède la date la plus ancienne, "False" si c'est la deuxième

Avertissement

si les dates sont identiques, la date de la première interraction est considérée la plus ancienne (partie "else" de la condition)

3.17 Référence de la classe gestionListeTaches

Classe définissant les attributs et méthodes permettant de gérer la liste de tâches d'un contact.

`#include <gestionlistetaches.h>`

Fonctions membres publiques

- **gestionListeTaches** ()
  le constructeur de la classe
- ~gestionListeTaches ()
  le destructeur de la classe
- std::list< toDo > **getListeToDo** () const
- void ajouterToDo (const toDo &)
  Fonction permettant d'ajouter une nouvelle tâche dans la liste.
- void supprimerToDo (const toDo &)
  Fonction permettant de supprimer une tâche dans la liste.
- void trierTDParDate ()
  Fonction permettant de trier la liste des tâches par date de réalisation.
- std::list< toDo > chercherTDParIntervalleDate (const date::year_month_day &, const date::year_month_day &)
  Fonction permettant de chercher uniquement les tâches de la liste dont la date de réalisation est comprise entre les 2 dates passées en paramètres.

Amis

- bool compare_DateToDo (const toDo &, const toDo &)
  Fonction amie permettant de comparer les dates de réalisation de 2 tâches, afin de déterminer laquelle est la plus récente et laquelle est la plus ancienne.

3.17.1 Description détaillée

Classe définissant les attributs et méthodes permettant de gérer la liste de tâches d'un contact.

3.17.2 Documentation des constructeurs et destructeur

3.17.2.1 ~gestionListeTaches()

```
gestionListeTaches::~gestionListeTaches ( )
```

le destructeur de la classe

Si la liste de tâches est supprimée, son contenu est nettoyé de ses données pour gagner de la place en mémoire

3.17.3 Documentation des fonctions membres

3.17.3.1 ajouterToDo()

```
void gestionListeTaches::ajouterToDo (
            const toDo & td )
```

Fonction permettant d'ajouter une nouvelle tâche dans la liste.

Paramètres

inte	correspond à la nouvelle tâche à ajouter

3.17.3.2 chercherTDParIntervalleDate()

```
std::list< toDo > gestionListeTaches::chercherTDParIntervalleDate (
            const date::year_month_day & d1,
            const date::year_month_day & d2 )
```

Fonction permettant de chercher uniquement les tâches de la liste dont la date de réalisation est comprise entre les 2 dates passées en paramètres.

Paramètres

d1	correspond à la date la plus ancienne de l'intervalle
d2	correspond à la date la plus récente de l'intervalle

Renvoie

une liste de tâches contenant uniquement les tâches qui remplissent la condition

3.17.3.3 supprimerToDo()

```
void gestionListeTaches::supprimerToDo (
            const toDo & td )
```

Fonction permettant de supprimer une tâche dans la liste.

Paramètres

inte	correspond à la tâche à supprimer de la liste

3.17.3.4 trierTDParDate()

```
void gestionListeTaches::trierTDParDate ( )
```

Fonction permettant de trier la liste des tâches par date de réalisation.

Voir également

utilise la fonction amie compare_DateTD comme paramètre de la fonction sort() pour établir la comparaison entre 2 tâches

3.17.4 Documentation des fonctions amies et associées

3.17.4.1 compare_DateToDo

```
bool compare_DateToDo (
            const toDo & td1,
            const toDo & td2 )  [friend]
```

Fonction amie permettant de comparer les dates de réalisation de 2 tâches, afin de déterminer laquelle est la plus récente et laquelle est la plus ancienne.

Voir également

utilisée comme paramètre dans la fonction trierTDParDate() de cette classe

Paramètres

td1	correspond à la première tâche dont on souhaite comparer la date de réalisation
i2	correspond à la deuxième tâche dont on souhaite comparer la date de réalisation

Renvoie

un booléen avec la valeur "True" si la première tâche possède la date la plus ancienne, "False" si c'est la deuxième

Avertissement

si les dates sont identiques, la date de la première tâche est considérée la plus ancienne (partie "else" de la condition)

3.18 Référence du modèle de la classe date::hh_mm_ss< Duration >

Types publics

— using **precision** = typename dfs::precision

Fonctions membres publiques

— CONSTCD11 **hh_mm_ss** (Duration d) NOEXCEPT
— CONSTCD11 std::chrono::hours **hours** () const NOEXCEPT
— CONSTCD11 std::chrono::minutes **minutes** () const NOEXCEPT
— CONSTCD11 std::chrono::seconds **seconds** () const NOEXCEPT
— CONSTCD14 std::chrono::seconds & **seconds** (detail::undocumented) NOEXCEPT
— CONSTCD11 precision **subseconds** () const NOEXCEPT
— CONSTCD11 bool **is_negative** () const NOEXCEPT
— CONSTCD11 **operator precision** () const NOEXCEPT
— CONSTCD11 precision **to_duration** () const NOEXCEPT
— CONSTCD11 bool **in_conventional_range** () const NOEXCEPT

Attributs publics statiques

— static unsigned CONSTDATA **fractional_width** = dfs::width

Amis

— template<class charT , class traits >
std::basic_ostream< charT, traits > & **operator**<< (std::basic_ostream< charT, traits > &os, hh_mm_ss const &tod)
— template<class CharT , class Traits , class Duration2 >
std::basic_ostream< CharT, Traits > & **date::to_stream** (std::basic_ostream< CharT, Traits > &os, const CharT *fmt, const fields< Duration2 > &fds, const std::string *abbrev, const std::chrono::seconds *offset↩
_sec)
— template<class CharT , class Traits , class Duration2 , class Alloc >
std::basic_istream< CharT, Traits > & **date::from_stream** (std::basic_istream< CharT, Traits > &is, const CharT *fmt, fields< Duration2 > &fds, std::basic_string< CharT, Traits, Alloc > *abbrev, std::chrono::minutes *offset)

3.19 Référence de la classe Interraction

Classe définissant les attributs et méthodes des interractions pour chaque contact géré par l'application.

`#include <interraction.h>`

3.19.1 Description détaillée

Classe définissant les attributs et méthodes des interractions pour chaque contact géré par l'application.

3.20 Référence de la classe interraction

Fonctions membres publiques

- **interraction** (const std::string &, const std::string &)
- ~**interraction** ()
  le destructeur de la classe
- std::string **getTitreInte** () const
- std::string **getContenuInte** () const
- date::year_month_day **getDateModifInte** () const
- gestionListeTaches **getListeTaches** () const
- void **setTitreInte** (const std::string &)
- void **setContenuInte** (const std::string &)
- void **setDateModifInte** (const date::year_month_day &)
- void **setListeTaches** (const gestionListeTaches &)
- bool operator== (const interraction &i)
  operateur ==
- void **modifierDonneesInterraction** (const std::string &, const std::string &)
- std::string **conversionDateModifToString** ()
- gestionListeTaches **traiterInterraction** ()

Amis

- std::ostream & operator<< (std::ostream &, const interraction &)
  Fonction amie permettant de prédéfinir l'affichage en console d'une interaction.

3.20.1 Documentation des fonctions membres

3.20.1.1 operator==()

```
bool interraction::operator== (
            const interraction & i )  [inline]
```

operateur ==

Voir également

Opérateur permettant de vérifer si une interaction est bien celle que l'on cherche (utilisé notamment dans la fonction supprimerInterraction() de la classe Contact)

Paramètres

i	correspond à l'interaction à vérifier

Renvoie

un booléen qui vaut "True" si l'interaction est bien celle que l'on cherche, "False" sinon

Avertissement

corps de l'operateur défini directement içi pour réduire le nombre de fonctions implémentées dans le .cpp

3.20.2 Documentation des fonctions amies et associées

3.20.2.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const interraction & inte )  [friend]
```

Fonction amie permettant de prédéfinir l'affichage en console d'une interaction.

Voir également

Utilisée essentiellement pour afficher les attributs des interactions lors des tests dans le main

Paramètres

os	correspond à la sortie console contenant la chaine de caractères
inte	correspond à l'interaction que l'on souhaite afficher

Renvoie

la chaîne de caractères affichée dans la console

3.21 Référence de la structure date::last_spec

3.22 Référence de la classe listeContacts

Fonctions membres publiques

- listeContacts ()
  le constructeur de la classe
- ~listeContacts ()
  le destructeur de la classe
- std::list< contact > **getListeC** () const
- unsigned int **getNbTotal** () const
- date::year_month_day **getDateDerniereSuppression** () const
- void **setNbTotal** (const unsigned int &)
- void **setDateDerniereSuppression** (const date::year_month_day &)
- void ajouterContact (const contact &)
  Fonction permettant d'ajouter un nouveau contact dans la liste.
- void **supprimerContact** (const contact &)
- void modifierContact (const contact &, const std::string &, const std::string &, const std::string &, const std::string &, const std::string &, const std::string &)
  Fonction permettant de modifier un contact dans la liste.

- void trierCParOrdreAlphabetique ()
  Fonction permettant de trier la liste de contacts par ordre alphabétique.
- void trierCParDateCreation ()
  Fonction permettant de trier la liste de contacts par date de création.
- std::list< contact > chercherCParNom (std::string &)
  Fonction permettant de chercher uniquement les contacts de la liste dont le nom commence de la même facon que la recherche passée en paramètres.
- std::list< contact > chercherCParEntreprise (std::string &)
  Fonction permettant de chercher uniquement les contacts de la liste dont l'entreprise commence de la même facon que la recherche passée en paramètres.
- std::list< contact > chercherCParIntervalleDate (const date::year_month_day &, const date::year_month_day &)
  Fonction permettant de chercher uniquement les contacts de la liste dont la date de création est comprise entre les 2 dates passées en paramètres.

Amis

- bool compare_Nom (const contact &, const contact &)
  Fonction amie permettant de comparer les noms et prénoms de 2 contacts, afin de déterminer lequel est le premier par ordre alphabétique.
- bool compare_Date (const contact &, const contact &)
  Fonction amie permettant de comparer les dates de création de 2 contacts, afin de déterminer laquelle est la plus récente et laquelle est la plus ancienne.

3.22.1 Documentation des constructeurs et destructeur

3.22.1.1 listeContacts()

```
listeContacts::listeContacts ( )
```

le constructeur de la classe

initialise le nombre total de contacts de la liste à 0 lors de la création

3.22.1.2 ~listeContacts()

```
listeContacts::~listeContacts ( )
```

le destructeur de la classe

Si la liste de contact est supprimée, son contenu est nettoyé de ses données pour gagner de la place en mémoire

3.22.2 Documentation des fonctions membres

3.22.2.1 ajouterContact()

```
void listeContacts::ajouterContact (
            const contact & c )
```

Fonction permettant d'ajouter un nouveau contact dans la liste.

Paramètres

c	correspond au nouveau contact à ajouter

Avertissement

si le contact est ajouté, on ajoute 1 au nombre total de contacts de la liste

3.22.2.2 chercherCParEntreprise()

```
std::list< contact > listeContacts::chercherCParEntreprise (
            std::string & recherche )
```

Fonction permettant de chercher uniquement les contacts de la liste dont l'entreprise commence de la même facon que la recherche passée en paramètres.

Paramètres

recherche	correspond à la chaine que l'on recherche dans l'entreprise d'un contact

Renvoie

une liste de contacts contenant uniquement les contacts qui remplissent la condition

Avertissement

si la recherche de l'utilisateur est vide, la fonction retourne tous les contacts de la liste

on convertit la recherche et l'entreprise de chaque contact afin de ne pas devoir respecter la casse lors de la comparaison

3.22.2.3 chercherCParIntervalleDate()

```
std::list< contact > listeContacts::chercherCParIntervalleDate (
            const date::year_month_day & d1,
            const date::year_month_day & d2 )
```

Fonction permettant de chercher uniquement les contacts de la liste dont la date de création est comprise entre les 2 dates passées en paramètres.

Paramètres

d1	correspond à la date la plus ancienne de l'intervalle
d2	correspond à la date la plus récente de l'intervalle

Renvoie

une liste de contacts contenant uniquement les contacts qui remplissent la condition

3.22.2.4 chercherCParNom()

```
std::list< contact > listeContacts::chercherCParNom (
            std::string & recherche )
```

Fonction permettant de chercher uniquement les contacts de la liste dont le nom commence de la même facon que la recherche passée en paramètres.

Paramètres

recherche	correspond à la chaine que l'on recherche dans un nom de contact

Renvoie

une liste de contacts contenant uniquement les contacts qui remplissent la condition

Avertissement

si la recherche de l'utilisateur est vide, la fonction retourne tous les contacts de la liste

on convertit la recherche et le nom de chaque contact afin de ne pas devoir respecter la casse lors de la comparaison

3.22.2.5 modifierContact()

```
void listeContacts::modifierContact (
            const contact & c,
            const std::string & n,
            const std::string & p,
            const std::string & e,
            const std::string & m,
            const std::string & t,
            const std::string & ph )
```

Fonction permettant de modifier un contact dans la liste.

Paramètres

i	correspond à l'interaction à supprimer
ti	correspond au nouveau titre de l'interaction
ci	correspond au nouveau contenu de l'interaction

Avertissement

on réalise une copie de l'interaction à modifier avant de la supprimer de la liste, puis on effectue les modifications sur la copie avant de la réajouter à la liste

Fonction permettant de modifier un contact dans la liste

Paramètres

c	correspond au contact dont on souhaite modifier les données
n	correspond au nouveau nom du contact
p	correspond au nouveau prénom du contact
e	correspond à la nouvelle entreprise du contact
m	correspond à la nouvelle adresse-mail du contact
t	correspond au nouveau numéro de téléphone du contact
ph	correspond à l'URL de la nouvelle photo

Avertissement

on réalise une copie du contact à modifier avant de le supprimer de la liste, puis on effectue les modifications sur la copie avant de la réajouter à la liste

3.22.2.6 trierCParDateCreation()

```
void listeContacts::trierCParDateCreation ( )
```

Fonction permettant de trier la liste de contacts par date de création.

Voir également

utilise la fonction amie compare_Date comme paramètre de la fonction sort() pour établir la comparaison entre 2 contacts

3.22.2.7 trierCParOrdreAlphabetique()

```
void listeContacts::trierCParOrdreAlphabetique ( )
```

Fonction permettant de trier la liste de contacts par ordre alphabétique.

Voir également

utilise la fonction amie compare_Nom comme paramètre de la fonction sort() pour établir la comparaison entre 2 contacts

3.22.3 Documentation des fonctions amies et associées

3.22.3.1 compare_Date

```
bool compare_Date (
            const contact & c1,
            const contact & c2 )  [friend]
```

Fonction amie permettant de comparer les dates de création de 2 contacts, afin de déterminer laquelle est la plus récente et laquelle est la plus ancienne.

Voir également

utilisée comme paramètre dans la fonction trierCParDateCreation() de cette classe

Paramètres

c1	correspond au premier contact dont on souhaite comparer la date de création
c2	correspond au deuxième contact dont on souhaite comparer la date de création

Renvoie

un booléen avec la valeur "True" si le premier contact possède la date la plus ancienne, "False" si c'est le deuxième

Avertissement

si les dates sont identiques, la date du premier contact est considérée la plus ancienne (partie "else" de la condition)

3.22.3.2 compare_Nom

```
bool compare_Nom (
            const contact & c1,
            const contact & c2 )  [friend]
```

Fonction amie permettant de comparer les noms et prénoms de 2 contacts, afin de déterminer lequel est le premier par ordre alphabétique.

Voir également

utilisée comme paramètre dans la fonction trierCParOrdreAlphabetique de cette classe

Paramètres

c1	correspond au premier contact dont on souhaite comparer le nom
c2	correspond au deuxième contact dont on souhaite comparer le nom

Renvoie

un booléen avec la valeur "True" si le premier contact est situé avant dans l'ordre alphabétique, "False" si c'est le deuxième

Avertissement

si les noms sont identiques, on compare alors les prénoms

si les noms et les prénoms sont identiques, le premier contact est considéré comme le premier par ordre alphabétique

Voir également

utilisée comme paramètre dans la fonction trierCParOrdreAlphabetique de cette classe

Paramètres

c1	correspond au premier contact dont on souhaite comparer le nom
c2	correspond au deuxième contact dont on souhaite comparer le nom

Renvoie

un booléen avec la valeur "True" si le premier contact est situé avant dans l'ordre alphabétique, "False" si c'est le deuxième

Avertissement

si les noms sont identiques, on compare alors les prénoms

si les noms et les prénoms sont identiques, le premier contact est considéré être le premier par ordre alphabétique

3.23 Référence de la classe ListeContacts

Classe définissant les attributs et méthodes de la liste de contacts gérée par l'application.

`#include <listecontacts.h>`

3.23.1 Description détaillée

Classe définissant les attributs et méthodes de la liste de contacts gérée par l'application.

3.24 Référence de la classe ListeTaches

Interface concernant la fenêtre listant les tâches d'un contact.

`#include <listetaches.h>`

Signaux

- void **fermerFenetre** ()
- void **actualiserApresSuppTache** ()

Fonctions membres publiques

- ListeTaches (QWidget *parent=nullptr, QString idContact="")
  le constructeur de la classe
- **~ListeTaches** ()
  le destructeur de la classe
- void afficherIdentiteContact (QSqlQuery &)
  fonction permettant d'afficher le nom et le prénom du contact associé à la liste de tâche consultée
- void **afficherListeTaches** (QSqlQuery &)
- bool close ()
  surcharge de la fonction close() de la fenêtre

3.24.1 Description détaillée

Interface concernant la fenêtre listant les tâches d'un contact.

Classe définissant les attributs et méthodes liées à la fenêtre listant les tâches d'un contact.

3.24.2 Documentation des constructeurs et destructeur

3.24.2.1 ListeTaches()

```
ListeTaches::ListeTaches (
            QWidget * parent = nullptr,
            QString idContact = "" )  [explicit]
```

le constructeur de la classe

Paramètres

idContact	correspond à l'identifiant du contact auquel est associé la liste de tâches que l'on consulte

on créé une instance d'interface

Voir également

on appelle la fonction actualiserFenetre() de cette classe pour afficher par défaut les données que l'on consulte

Avertissement

toutes les connexions signaux/slots liées a des éléments de l'interface sont effectuées dans le constructeur

3.24.3 Documentation des fonctions membres

3.24.3.1 afficherIdentiteContact()

```
void ListeTaches::afficherIdentiteContact (
            QSqlQuery & qu )
```

fonction permettant d'afficher le nom et le prénom du contact associé à la liste de tâche consultée

Paramètres

qu	correpond au résultat de la requête qui retourne l'identité du contact

3.24.3.2 close()

```
bool ListeTaches::close ( )
```

surcharge de la fonction close() de la fenêtre

on emet le signal fermerFenetre pour actualiser la fenêtre qui va être ouverte à la place

Renvoie

l'état close() de la fenetre

3.25 Référence de la classe ListeTotaliteTaches

Interface concernant la fenêtre listant les tâches de tous les contacts.

```
#include <listetotalitetaches.h>
```

Signaux

- void **fermerFenetre** ()
- void **actualiserApresSuppTache** ()

Fonctions membres publiques

- ListeTotaliteTaches (QWidget *parent=nullptr)
  le constructeur de la classe
- ~**ListeTotaliteTaches** ()
  le destructeur de la classe
- void **afficherListeTaches** (QSqlQuery &)
- bool close ()
  surcharge de la fonction close() de la fenêtre

3.25.1 Description détaillée

Interface concernant la fenêtre listant les tâches de tous les contacts.

Classe définissant les attributs et méthodes liées à la fenêtre listant les tâches de tous les contacts.

3.25.2 Documentation des constructeurs et destructeur

3.25.2.1 ListeTotaliteTaches()

```
ListeTotaliteTaches::ListeTotaliteTaches (
            QWidget * parent = nullptr )  [explicit]
```

le constructeur de la classe

on créé une instance d'interface

Voir également

on appelle la fonction actualiserFenetre() de cette classe pour afficher par défaut les données que l'on consulte

Avertissement

toutes les connexions signaux/slots liées a des éléments de l'interface sont effectuées dans le constructeur

3.25.3 Documentation des fonctions membres

3.25.3.1 close()

```
bool ListeTotaliteTaches::close ( )
```

surcharge de la fonction close() de la fenêtre

on emet le signal fermerFenetre pour actualiser la fenêtre qui va être ouverte à la place

Renvoie

l'état close() de la fenetre

3.26 Référence de la structure date::local_t

3.27 Référence de la classe date::month

Fonctions membres publiques

- CONSTCD11 **month** (unsigned m) NOEXCEPT
- CONSTCD14 month & **operator++** () NOEXCEPT
- CONSTCD14 month **operator++** (int) NOEXCEPT
- CONSTCD14 month & **operator--** () NOEXCEPT
- CONSTCD14 month **operator--** (int) NOEXCEPT
- CONSTCD14 month & **operator+=** (const months &m) NOEXCEPT
- CONSTCD14 month & **operator-=** (const months &m) NOEXCEPT
- CONSTCD11 **operator unsigned** () const NOEXCEPT
- CONSTCD11 bool **ok** () const NOEXCEPT

3.28 Référence de la classe date::month_day

Fonctions membres publiques

- CONSTCD11 **month_day** (const date::month &m, const date::day &d) NOEXCEPT
- CONSTCD11 date::month **month** () const NOEXCEPT
- CONSTCD11 date::day **day** () const NOEXCEPT
- CONSTCD14 bool **ok** () const NOEXCEPT

3.29 Référence de la classe date::month_day_last

Fonctions membres publiques

- CONSTCD11 **month_day_last** (const date::month &m) NOEXCEPT
- CONSTCD11 date::month **month** () const NOEXCEPT
- CONSTCD11 bool **ok** () const NOEXCEPT

3.30 Référence de la classe date::month_weekday

Fonctions membres publiques

- CONSTCD11 **month_weekday** (const date::month &m, const date::weekday_indexed &wdi) NOEXCEPT
- CONSTCD11 date::month **month** () const NOEXCEPT
- CONSTCD11 date::weekday_indexed **weekday_indexed** () const NOEXCEPT
- CONSTCD11 bool **ok** () const NOEXCEPT

3.31 Référence de la classe date::month_weekday_last

Fonctions membres publiques

- CONSTCD11 **month_weekday_last** (const date::month &m, const date::weekday_last &wd) NOEXCEPT
- CONSTCD11 date::month **month** () const NOEXCEPT
- CONSTCD11 date::weekday_last **weekday_last** () const NOEXCEPT
- CONSTCD11 bool **ok** () const NOEXCEPT

3.32 Référence du modèle de la structure date::detail::no_overflow< R1, R2 >

Types publics

— typedef std::ratio< mul< n1, d2, !value >::value, mul< n2, d1, !value >::value > **type**

Attributs publics statiques

— static const bool **value** = (n1 <= max / d2) && (n2 <= max / d1)

3.33 Référence du modèle de la structure date::parse_manip< Parsable, CharT, Traits, Alloc >

Fonctions membres publiques

— **parse_manip** (std::basic_string< CharT, Traits, Alloc > format, Parsable &tp, std::basic_string< CharT, Traits, Alloc > *abbrev=nullptr, std::chrono::minutes *offset=nullptr)

Attributs publics

— const std::basic_string< CharT, Traits, Alloc > **format_**
— Parsable & **tp_**
— std::basic_string< CharT, Traits, Alloc > * **abbrev_**
— std::chrono::minutes * **offset_**

3.34 Référence de la classe PremiereFenetre

Signaux

— void **signalAfficherChampRecherche** ()
— void **signalAfficherIntervDates** ()

Fonctions membres publiques

— PremiereFenetre (QWidget *parent=nullptr)
le constructeur de la classe
— ~**PremiereFenetre** ()
le destructeur de la classe
— void **recupererCheminBDD** ()
— void **afficherLignesTableau** (QSqlQuery &)

3.34.1 Documentation des constructeurs et destructeur

3.34.1.1 PremiereFenetre()

```
PremiereFenetre::PremiereFenetre (
            QWidget * parent = nullptr )  [explicit]
```

le constructeur de la classe

on créé une instance d'interface

Voir également

on appelle la fonction recupererCheminBDD() de cette classe pour afficher une pop-up permettant à l'utilisateur de choisir la connexion bdd à utiliser

on appelle la fonction actualiserFenetre() de cette classe pour afficher par défaut les données que l'on consulte

on appelle la fonction afficherChampRecherche() de cette classe pour afficher par défaut un champ de recherche textuel

Avertissement

toutes les connexions signaux/slots liées a des éléments de l'interface sont effectuées dans le constructeur

3.35 Référence de la structure date::detail::rld

Attributs publics

- long double & **i**
- unsigned **m**
- unsigned **M**

3.36 Référence de la structure date::detail::rs

Attributs publics

- int & **i**
- unsigned **m**
- unsigned **M**

3.37 Référence de la structure date::detail::ru

Attributs publics

- int & **i**
- unsigned **m**
- unsigned **M**

3.38 Référence du modèle de la classe date::detail::save_istream< CharT, Traits >

Fonctions membres publiques

- **save_istream** (const save_istream &)=delete
- save_istream & **operator=** (const save_istream &)=delete
- **save_istream** (std::basic_ios< CharT, Traits > &is)

Attributs protégés

- std::basic_ios< CharT, Traits > & **is_**
- CharT **fill_**
- std::ios::fmtflags **flags_**
- std::streamsize **precision_**
- std::streamsize **width_**
- std::basic_ostream< CharT, Traits > * **tie_**
- std::locale **loc_**

3.39 Référence du modèle de la classe date::detail::save_ostream< CharT, Traits >

Fonctions membres publiques

- **save_ostream** (const save_ostream &)=delete
- save_ostream & **operator=** (const save_ostream &)=delete
- **save_ostream** (std::basic_ios< CharT, Traits > &os)

3.40 Référence du modèle de la structure date::detail::static_gcd< Xp, Yp >

Attributs publics statiques

- static const std::intmax_t **value** = static_gcd<Yp, Xp % Yp>::value

3.41 Référence de la structure date::detail::static_gcd< 0, 0 >

Attributs publics statiques

- static const std::intmax_t **value** = 1

3.42 Référence du modèle de la structure date::detail::static_gcd< Xp, 0 >

Attributs publics statiques

- static const std::intmax_t **value** = Xp

3.43 Référence du modèle de la structure date::detail::static_pow10< exp >

Attributs publics statiques

— static CONSTDATA std::uint64_t **value** = h ∗ h ∗ (exp % 2 ? 10 : 1)

3.44 Référence de la structure date::detail::static_pow10< 0 >

Attributs publics statiques

— static CONSTDATA std::uint64_t **value** = 1

3.45 Référence du modèle de la classe date::detail::string_literal< CharT, N >

3.46 Référence de la classe toDo

Fonctions membres publiques

— **toDo** (const std::string &)
— toDo (const std::string &, const date::year_month_day &)
un constructeur de la classe, utilisé dans le cas où la date de réalisation de la tâche est précisée dans l'interaction
— ∼**toDo** ()
le destructeur de la classe
— std::string **getContenuTD** () const
— date::year_month_day **getDateTD** () const
— void **setContenuTD** (const std::string &)
— void **setDateTD** (const date::year_month_day &)
— bool operator== (const toDo &td)
operateur ==
— std::string **conversionDateReaToString** ()

Amis

— std::ostream & operator<< (std::ostream &, const toDo &)
Fonction amie permettant de prédéfinir l'affichage en console d'une tâche.

3.46.1 Documentation des constructeurs et destructeur

3.46.1.1 toDo()

```
toDo::toDo (
            const std::string & c,
            const date::year_month_day & d )
```

un constructeur de la classe, utilisé dans le cas où la date de réalisation de la tâche est précisée dans l'interaction

Paramètres

c	correspond au contenu de la tâche
d	correspond à la date de réalisation de la tâche

Avertissement

La date de réalisation de la tâche est la date définie dans l'interaction

3.46.2 Documentation des fonctions membres

3.46.2.1 operator==()

```
bool toDo::operator== (
            const toDo & td )  [inline]
```

operateur ==

Voir également

Opérateur permettant de vérifer si une tâche est bien celle que l'on cherche (utilisé notamment dans la fonction supprimerToDo() de la classe Contact)

Paramètres

td	correspond à la tâche à vérifier

Renvoie

un booléen qui vaut "True" si la tâche est bien celle que l'on cherche, "False" sinon

Avertissement

corps de l'operateur défini directement içi pour réduire le nombre de fonctions implémentées dans le .cpp

3.46.3 Documentation des fonctions amies et associées

3.46.3.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const toDo & td )  [friend]
```

Fonction amie permettant de prédéfinir l'affichage en console d'une tâche.

Voir également

Utilisée essentiellement pour afficher les attributs des tâches lors des tests dans le main

Paramètres

os	correspond à la sortie console contenant la chaine de caractères
inte	correspond à la tâche que l'on souhaite afficher

Renvoie

la chaîne de caractères affichée dans la console

3.47 Référence de la classe ToDo

Classe définissant les attributs et méthodes des tâches pour chaque contact géré par l'application.

`#include <todo.h>`

3.47.1 Description détaillée

Classe définissant les attributs et méthodes des tâches pour chaque contact géré par l'application.

3.48 Référence de la structure date::detail::undocumented

3.49 Référence de la structure date::detail::unspecified_month_disambiguator

3.50 Référence de la classe date::weekday

Fonctions membres publiques

- CONSTCD11 **weekday** (unsigned wd) NOEXCEPT
- CONSTCD14 **weekday** (const sys_days &dp) NOEXCEPT
- CONSTCD14 **weekday** (const local_days &dp) NOEXCEPT
- CONSTCD14 weekday & **operator++** () NOEXCEPT
- CONSTCD14 weekday **operator++** (int) NOEXCEPT
- CONSTCD14 weekday & **operator--** () NOEXCEPT
- CONSTCD14 weekday **operator--** (int) NOEXCEPT
- CONSTCD14 weekday & **operator+=** (const days &d) NOEXCEPT
- CONSTCD14 weekday & **operator-=** (const days &d) NOEXCEPT
- CONSTCD11 bool **ok** () const NOEXCEPT
- CONSTCD11 unsigned **c_encoding** () const NOEXCEPT
- CONSTCD11 unsigned **iso_encoding** () const NOEXCEPT
- CONSTCD11 weekday_indexed **operator[ ]** (unsigned index) const NOEXCEPT
- CONSTCD11 weekday_last **operator[ ]** (last_spec) const NOEXCEPT

Amis

- class **weekday_indexed**
- CONSTCD11 bool **operator==** (const weekday &x, const weekday &y) NOEXCEPT
- CONSTCD14 days **operator-** (const weekday &x, const weekday &y) NOEXCEPT
- CONSTCD14 weekday **operator+** (const weekday &x, const days &y) NOEXCEPT
- template<class CharT , class Traits >
  std::basic_ostream< CharT, Traits > & **operator**<< (std::basic_ostream< CharT, Traits > &os, const weekday &wd)

3.51 Référence de la classe date::weekday_indexed

Fonctions membres publiques

- CONSTCD11 **weekday_indexed** (const date::weekday &wd, unsigned index) NOEXCEPT
- CONSTCD11 date::weekday **weekday** () const NOEXCEPT
- CONSTCD11 unsigned **index** () const NOEXCEPT
- CONSTCD11 bool **ok** () const NOEXCEPT

3.52 Référence de la classe date::weekday_last

Fonctions membres publiques

- CONSTCD11 **weekday_last** (const date::weekday &wd) NOEXCEPT
- CONSTCD11 date::weekday **weekday** () const NOEXCEPT
- CONSTCD11 bool **ok** () const NOEXCEPT

3.53 Référence du modèle de la structure date::detail::width< n, d, w, should_continue >

Attributs publics statiques

- static CONSTDATA unsigned **value** = 1 + width<n%d*10, d, w+1>::value

3.54 Référence du modèle de la structure date::detail::width< n, d, w, false >

Attributs publics statiques

- static CONSTDATA unsigned **value** = 0

3.55 Référence de la classe date::year

Fonctions membres publiques

- CONSTCD11 **year** (int y) NOEXCEPT
- CONSTCD14 year & **operator++** () NOEXCEPT
- CONSTCD14 year **operator++** (int) NOEXCEPT
- CONSTCD14 year & **operator--** () NOEXCEPT
- CONSTCD14 year **operator--** (int) NOEXCEPT
- CONSTCD14 year & **operator+=** (const years &y) NOEXCEPT
- CONSTCD14 year & **operator-=** (const years &y) NOEXCEPT
- CONSTCD11 year **operator-** () const NOEXCEPT
- CONSTCD11 year **operator+** () const NOEXCEPT
- CONSTCD11 bool **is_leap** () const NOEXCEPT
- CONSTCD11 **operator int** () const NOEXCEPT
- CONSTCD11 bool **ok** () const NOEXCEPT

Fonctions membres publiques statiques

- static CONSTCD11 year **min** () NOEXCEPT
- static CONSTCD11 year **max** () NOEXCEPT

3.56 Référence de la classe date::year_month

Fonctions membres publiques

- CONSTCD11 **year_month** (const date::year &y, const date::month &m) NOEXCEPT
- CONSTCD11 date::year **year** () const NOEXCEPT
- CONSTCD11 date::month **month** () const NOEXCEPT
- template<class = detail::unspecified_month_disambiguator>
  CONSTCD14 year_month & **operator+=** (const months &dm) NOEXCEPT
- template<class = detail::unspecified_month_disambiguator>
  CONSTCD14 year_month & **operator-=** (const months &dm) NOEXCEPT
- CONSTCD14 year_month & **operator+=** (const years &dy) NOEXCEPT
- CONSTCD14 year_month & **operator-=** (const years &dy) NOEXCEPT
- CONSTCD11 bool **ok** () const NOEXCEPT

3.57 Référence de la classe date::year_month_day

Fonctions membres publiques

- CONSTCD11 **year_month_day** (const date::year &y, const date::month &m, const date::day &d) NOEXCEPT
- CONSTCD14 **year_month_day** (const year_month_day_last &ymdl) NOEXCEPT
- CONSTCD14 **year_month_day** (sys_days dp) NOEXCEPT
- CONSTCD14 **year_month_day** (local_days dp) NOEXCEPT
- template<class = detail::unspecified_month_disambiguator>
  CONSTCD14 year_month_day & **operator+=** (const months &m) NOEXCEPT
- template<class = detail::unspecified_month_disambiguator>
  CONSTCD14 year_month_day & **operator-=** (const months &m) NOEXCEPT
- CONSTCD14 year_month_day & **operator+=** (const years &y) NOEXCEPT
- CONSTCD14 year_month_day & **operator-=** (const years &y) NOEXCEPT
- CONSTCD11 date::year **year** () const NOEXCEPT
- CONSTCD11 date::month **month** () const NOEXCEPT
- CONSTCD11 date::day **day** () const NOEXCEPT
- CONSTCD14 **operator sys_days** () const NOEXCEPT
- CONSTCD14 **operator local_days** () const NOEXCEPT
- CONSTCD14 bool **ok** () const NOEXCEPT

3.58 Référence de la classe date::year_month_day_last

Fonctions membres publiques

- CONSTCD11 **year_month_day_last** (const date::year &y, const date::month_day_last &mdl) NOEXCEPT
- template<class = detail::unspecified_month_disambiguator>
  CONSTCD14 year_month_day_last & **operator+=** (const months &m) NOEXCEPT
- template<class = detail::unspecified_month_disambiguator>
  CONSTCD14 year_month_day_last & **operator-=** (const months &m) NOEXCEPT
- CONSTCD14 year_month_day_last & **operator+=** (const years &y) NOEXCEPT
- CONSTCD14 year_month_day_last & **operator-=** (const years &y) NOEXCEPT
- CONSTCD11 date::year **year** () const NOEXCEPT
- CONSTCD11 date::month **month** () const NOEXCEPT
- CONSTCD11 date::month_day_last **month_day_last** () const NOEXCEPT
- CONSTCD14 date::day **day** () const NOEXCEPT
- CONSTCD14 **operator sys_days** () const NOEXCEPT
- CONSTCD14 **operator local_days** () const NOEXCEPT
- CONSTCD11 bool **ok** () const NOEXCEPT

3.59 Référence de la classe date::year_month_weekday

Fonctions membres publiques

- CONSTCD11 **year_month_weekday** (const date::year &y, const date::month &m, const date::weekday_indexed &wdi) NOEXCEPT
- CONSTCD14 **year_month_weekday** (const sys_days &dp) NOEXCEPT
- CONSTCD14 **year_month_weekday** (const local_days &dp) NOEXCEPT
- template<class = detail::unspecified_month_disambiguator>
  CONSTCD14 year_month_weekday & **operator+=** (const months &m) NOEXCEPT
- template<class = detail::unspecified_month_disambiguator>
  CONSTCD14 year_month_weekday & **operator-=** (const months &m) NOEXCEPT
- CONSTCD14 year_month_weekday & **operator+=** (const years &y) NOEXCEPT
- CONSTCD14 year_month_weekday & **operator-=** (const years &y) NOEXCEPT
- CONSTCD11 date::year **year** () const NOEXCEPT
- CONSTCD11 date::month **month** () const NOEXCEPT
- CONSTCD11 date::weekday **weekday** () const NOEXCEPT
- CONSTCD11 unsigned **index** () const NOEXCEPT
- CONSTCD11 date::weekday_indexed **weekday_indexed** () const NOEXCEPT
- CONSTCD14 **operator sys_days** () const NOEXCEPT
- CONSTCD14 **operator local_days** () const NOEXCEPT
- CONSTCD14 bool **ok** () const NOEXCEPT

3.60 Référence de la classe date::year_month_weekday_last

Fonctions membres publiques

- CONSTCD11 **year_month_weekday_last** (const date::year &y, const date::month &m, const date::weekday_last &wdl) NOEXCEPT
- template<class = detail::unspecified_month_disambiguator>
  CONSTCD14 year_month_weekday_last & **operator+=** (const months &m) NOEXCEPT
- template<class = detail::unspecified_month_disambiguator>
  CONSTCD14 year_month_weekday_last & **operator-=** (const months &m) NOEXCEPT
- CONSTCD14 year_month_weekday_last & **operator+=** (const years &y) NOEXCEPT
- CONSTCD14 year_month_weekday_last & **operator-=** (const years &y) NOEXCEPT
- CONSTCD11 date::year **year** () const NOEXCEPT
- CONSTCD11 date::month **month** () const NOEXCEPT
- CONSTCD11 date::weekday **weekday** () const NOEXCEPT
- CONSTCD11 date::weekday_last **weekday_last** () const NOEXCEPT
- CONSTCD14 **operator sys_days** () const NOEXCEPT
- CONSTCD14 **operator local_days** () const NOEXCEPT
- CONSTCD11 bool **ok** () const NOEXCEPT

Index

~gestionListeContacts
    gestionListeContacts, 27
~gestionListeInteractions
    gestionListeInteractions, 32
~gestionListeTaches
    gestionListeTaches, 35
~listeContacts
    listeContacts, 41

afficherDonneesContact
    FicheContact, 14
afficherDonneesInteraction
    DetailInteraction, 13
afficherIdentiteContact
    ListeTaches, 47
AjoutContact, 5
    AjoutContact, 5
    close, 6
    initialisationFenetre, 6
    VerifsChampsContact, 6
ajoutContact, 7
ajouterContact
    gestionListeContacts, 27
    listeContacts, 41
ajouterInterraction
    gestionListeInteractions, 32
ajouterNouveauContact
    gestionBDD, 18
ajouterNouvelleInteraction
    gestionBDD, 18
ajouterNouvelleTache
    gestionBDD, 19
ajouterToDo
    gestionListeTaches, 36
ajoutInteraction, 7
    ajoutInteraction, 7
    close, 8

changerDerniereSuppressionContact
    gestionBDD, 19
chercherCParEntreprise
    gestionListeContacts, 27
    listeContacts, 42
chercherCParIntervalleDate
    gestionListeContacts, 28
    listeContacts, 42
chercherCParNom
    gestionListeContacts, 28
    listeContacts, 43
chercherInteParIntervalleDate
    gestionListeInteractions, 33
chercherTDParIntervalleDate
    gestionListeTaches, 36
close
    AjoutContact, 6
    ajoutInteraction, 8
    DetailInteraction, 13
    FicheContact, 15
    ListeTaches, 48
    ListeTotaliteTaches, 49
compare_Date
    gestionListeContacts, 30
    listeContacts, 44
compare_DateInteractions
    gestionListeInteractions, 34
compare_DateToDo
    gestionListeTaches, 37
compare_Nom
    gestionListeContacts, 30
    listeContacts, 45
connexionBDD
    gestionBDD, 19
Contact, 10
contact, 9
    operator<<, 10
    operator==, 9

date::day, 11
date::detail::choose_trunc_type< T >, 8
date::detail::decimal_format_seconds< Duration >, 11
date::detail::no_overflow< R1, R2 >, 51
date::detail::rld, 52
date::detail::rs, 52
date::detail::ru, 52
date::detail::save_istream< CharT, Traits >, 53
date::detail::save_ostream< CharT, Traits >, 53
date::detail::static_gcd< 0, 0 >, 53
date::detail::static_gcd< Xp, 0 >, 53
date::detail::static_gcd< Xp, Yp >, 53
date::detail::static_pow10< 0 >, 54
date::detail::static_pow10< exp >, 54
date::detail::string_literal< CharT, N >, 54
date::detail::undocumented, 56
date::detail::unspecified_month_disambiguator, 56
date::detail::width< n, d, w, false >, 57
date::detail::width< n, d, w, should_continue >, 57
date::fields< Duration >, 16
date::hh_mm_ss< Duration >, 38
date::last_spec, 40
date::local_t, 50

date::month, 50
date::month_day, 50
date::month_day_last, 50
date::month_weekday, 50
date::month_weekday_last, 50
date::parse_manip< Parsable, CharT, Traits, Alloc >, 51
date::weekday, 56
date::weekday_indexed, 57
date::weekday_last, 57
date::year, 58
date::year_month, 58
date::year_month_day, 58
date::year_month_day_last, 59
date::year_month_weekday, 59
date::year_month_weekday_last, 59
DetailInteraction, 12
    afficherDonneesInteraction, 13
    close, 13
    DetailInteraction, 12
detailInteraction, 11

FicheContact, 14
    afficherDonneesContact, 14
    close, 15
    FicheContact, 14
    verifsChampsContact, 15
ficheContact, 13

gestionBDD, 16
    ajouterNouveauContact, 18
    ajouterNouvelleInteraction, 18
    ajouterNouvelleTache, 19
    changerDerniereSuppressionContact, 19
    connexionBDD, 19
    gestionBDD, 17
    modifierContact, 20
    modifierInteraction, 20
    rechercherContactSelonEntreprise, 21
    rechercherContactSelonNom, 21
    recupererIdentifiantInteraction, 22
    remplirTableauContactsParAlphabetique, 22
    remplirTableauContactsParDate, 22
    remplirTableauInteParDate, 22
    remplirTableauTachesContact, 23
    remplirTableauTotaliteTaches, 23
    retournerDerniereSuppressionContact, 23
    retournerDonneesContact, 23
    retournerDonneesInteraction, 24
    retournerIdentiteContact, 24
    retournerNbTotalContacts, 25
    supprimerContact, 25
    supprimerInteraction, 25
    supprimerTache, 25
gestionListeContacts, 26
    ~gestionListeContacts, 27
    ajouterContact, 27
    chercherCParEntreprise, 27
    chercherCParIntervalleDate, 28
    chercherCParNom, 28
    compare_Date, 30
    compare_Nom, 30
    gestionListeContacts, 27
    modifierContact, 29
    trierCParDateCreation, 29
    trierCParOrdreAlphabetique, 30
gestionListeInteractions, 31
    ~gestionListeInteractions, 32
    ajouterInterraction, 32
    chercherInteParIntervalleDate, 33
    compare_DateInteractions, 34
    modifierInterraction, 33
    supprimerInterraction, 34
    trierInteParDate, 34
gestionListeTaches, 35
    ~gestionListeTaches, 35
    ajouterToDo, 36
    chercherTDParIntervalleDate, 36
    compare_DateToDo, 37
    supprimerToDo, 36
    trierTDParDate, 37

initialisationFenetre
    AjoutContact, 6
Interraction, 38
interraction, 39
    operator<<, 40
    operator==, 39

ListeContacts, 46
listeContacts, 40
    ~listeContacts, 41
    ajouterContact, 41
    chercherCParEntreprise, 42
    chercherCParIntervalleDate, 42
    chercherCParNom, 43
    compare_Date, 44
    compare_Nom, 45
    listeContacts, 41
    modifierContact, 43
    trierCParDateCreation, 44
    trierCParOrdreAlphabetique, 44
ListeTaches, 46
    afficherIdentiteContact, 47
    close, 48
    ListeTaches, 47
ListeTotaliteTaches, 48
    close, 49
    ListeTotaliteTaches, 49

modifierContact
    gestionBDD, 20
    gestionListeContacts, 29
    listeContacts, 43
modifierInteraction
    gestionBDD, 20
modifierInterraction
    gestionListeInteractions, 33

operator<<
contact, 10
interraction, 40
toDo, 55
operator==
contact, 9
interraction, 39
toDo, 55

PremiereFenetre, 51
PremiereFenetre, 51

rechercherContactSelonEntreprise
gestionBDD, 21
rechercherContactSelonNom
gestionBDD, 21
recupererIdentifiantInteraction
gestionBDD, 22
remplirTableauContactsParAlphabetique
gestionBDD, 22
remplirTableauContactsParDate
gestionBDD, 22
remplirTableauInteParDate
gestionBDD, 22
remplirTableauTachesContact
gestionBDD, 23
remplirTableauTotaliteTaches
gestionBDD, 23
retournerDerniereSuppressionContact
gestionBDD, 23
retournerDonneesContact
gestionBDD, 23
retournerDonneesInteraction
gestionBDD, 24
retournerIdentiteContact
gestionBDD, 24
retournerNbTotalContacts
gestionBDD, 25

supprimerContact
gestionBDD, 25
supprimerInteraction
gestionBDD, 25
supprimerInterraction
gestionListeInteractions, 34
supprimerTache
gestionBDD, 25
supprimerToDo
gestionListeTaches, 36

ToDo, 56
toDo, 54
operator<<, 55
operator==, 55
toDo, 54
trierCParDateCreation
gestionListeContacts, 29
listeContacts, 44
trierCParOrdreAlphabetique
gestionListeContacts, 30
listeContacts, 44
trierInteParDate
gestionListeInteractions, 34
trierTDParDate
gestionListeTaches, 37

VerifsChampsContact
AjoutContact, 6
verifsChampsContact
FicheContact, 15