Fiche du cours

65 h

Titre :

CPP430 - Qt 6 : Applications GUI & Multiplateformes

Description :

Construire des applications desktop multiplateformes (Windows/macOS/Linux) en C++/Qt 6 avec CMake : modèle objet Qt (signals/slots), Widgets et Qt Quick/QML, Model–View, persistance (QSettings, SQLite), réseau (REST/JSON, WebSocket), concurrence (QThread/QtConcurrent), tests (Qt Test), journalisation (QLoggingCategory), i18n (Qt Linguist) et packaging (windeployqt/macdeployqt/linuxdeploy). Capstone : une app “portfolio-read

Objectifs :

- Maîtriser le modèle objet Qt et les signals/slots.*
- Concevoir des interfaces avec Widgets et Qt Quick/QML (et choisir le bon outil).*
- Implémenter Model–View (QAbstractItemModel) pour des listes/tableaux performants.*
- Consommer des API REST (QNetworkAccessManager), parser JSON, gérer l’état hors-ligne.*
- Gérer la persistance : QSettings, Qt SQL (SQLite), sérialisation.*
- Exécuter des tâches en arrière-plan (QThread, Worker, QtConcurrent) sans bloquer l’UI.*
- Tester (Qt Test), tracer/logguer proprement, localiser l’application, et packager pour la distribution.

Chapitres :

1. Setup & outillage : Qt 6, Qt Creator/VS Code, CMake, structure de projet*
2. Modèle Qt & signals/slots : QObject, propriétés, connexions modernes*
3. Widgets I : layouts, formulaires, validation, dialogues, ressources (.qrc)*
4. Model–View : QList/QTable + QAbstractItemModel (bases), délégués*
5. QML/Qt Quick I : composants, bindings, états/animations, styles*
6. C++ ↔ QML : QQmlEngine/Context, types enregistrés, données en temps réel*
7. Données & persistance : QSettings, Qt SQL (SQLite), migrations simples*
8. Réseau : QNetworkAccessManager, REST/JSON, erreurs, WebSocket (aperçu)*
9. Concurrence : QThread, Worker objects, QtConcurrent, barre de progression*
10. UX & qualité : accessibilité, raccourcis, thèmes/icônes, QLoggingCategory*
11. Tests & i18n : Qt Test, tests UI ciblés, Qt Linguist (traductions)*

12. Packaging & Capstone : windeployqt/macdeployqt/linuxdeploy, signatures (aperçu), livraison de l’app

À la fin :

Vous saurez bâtir une application Qt 6 complète : UI Widgets ou QML bien structurée, Model–View performant, persistance SQLite, API réseau robuste, tâches asynchrones non bloquantes, tests, i18n et packaging — un livrable multiplateforme prêt à montrer en portfolio ou à déployer en interne.
Vous appliquerez le modèle objet Qt (signals/slots), une gestion d’état propre, la validation de formulaires et un journaling clair (QLoggingCategory) pour diagnostiquer les erreurs en production.
Le livrable inclura un projet CMake propre, des tests Qt Test, des scripts de déploiement (windeployqt/macdeployqt/linuxdeploy) et un README d’exploitation, afin de faciliter la distribution sur Windows, macOS et Linux.