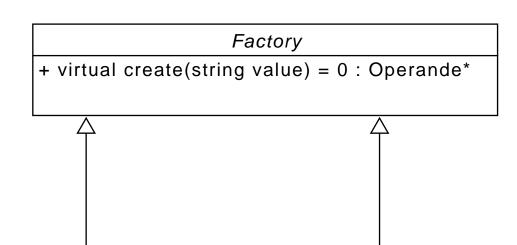
La classe Controleur implémente les designs patterns Singleton et Memento non détaillés dans cet UML

Controleur

- Controleur()
- Controleur(const Controleur& m)
- operator=(const Controleur& m) : Controleur&
- virtual ~Controleur()
- + empiler(Operande* val) : void
- + empiler (Factory& facto, string value): void
- + executer() : void
- + pileString(int size) : string
- + commande(string cmd): void
- + returnPile() const : stack
- + static getInstance() : Controleur&
- + static libererInstance(): void



${\bf Factory Litterale}$

- FactoryLitterale()
- FactoryLitterale(const FactoryLitterale& m)
- operator=(const FactoryLitterale& m) : FactoryLitterale&
- virtual ~FactoryLitterale()
- + static getInstance(): FactoryLitterale&
- + static libererInstance(): void
- + create(string val) : Operande*

FactoryOperateur

- FactoryOperateur()
- FactoryOperateur(const FactoryLitterale& m)
- operator=(const FactoryOperateur& m) : FactoryOperateur&
- virtual ~FactoryOperateur()
- + static getInstance() : FactoryOperateur&
- + static libererInstance(): void
- + create(string val) : Operande*

Les factorys implementent le design pattern Singleton non détaillé dans cet UML.