

# MGL7460 – Réalisation et maintenance des logiciels

Lisibilité et style

Jacques Berger

# Objectifs

Introduire les problématiques liées au style

# Prérequis

Aucun

# Introduction

Où placez-vous les accolades?

# Personnalité

Personnalité créatrice

L'art de la programmation

# Affirmation

Le code est plus souvent lu qu'écrit. La lisibilité est donc très importante.

# Style

Le style comprend :

- La forme du code

- La mise en page de la documentation dans le code

# Différence

Documentation vs commentaires

# Standards de programmation

S'entendre sur le style

Éviter les conflits

Éviter la perte de temps lors des revues de code

# Standards de programmation

Uniformiser le code

Améliorer la lisibilité

Éviter de perdre du temps à changer le style d'un autre

# Discussion

Devrait-on écrire un document de standards?

# Sujets

La position des accolades

Les espaces (dans les parenthèses, après les mots-clés, etc.)

L'indentation (nombre d'espace, tabulation)

La longueur des instructions

# Sujets

Déclarations de variables

Mise en page des commentaires

L'ordre d'apparition des éléments dans un fichier ou une classe (variables membres au début, constructeurs, destructeurs, méthodes publiques, etc)

Nomenclature

# Discussion

Si les standards évoluent, devrait-on faire évoluer le code également?

# Discussion

Que faire contre ceux qui refusent de suivre les standards?

# Discussion

Est-ce pertinent d'avoir un comité de standards?

# Outils

Code beautifier

Checkstyle

# Recommandations

Utiliser un outil comme checkstyle

Configurer l'IDE pour se préoccuper le moins possible du style

Suivre les standards du langage si possible, sinon écrire un document de standards