Clean Code

Red Grade

PRINCIPLES
- Don’t repeat yourself
- Keep it simple, stupid
- Beware of optimization!
- Composition over inheritance

PRACTICES
- Boy scout rule
- Root cause analysis
- Version control system
- Simple refactorings
- Reflect daily

Orange Grade

PRINCIPLES
- Single level of abstraction
- Single responsibility principle
- Separation of concerns
- Source code conventions

PRACTICES
- Issue tracking
- Automated int.-tests
- Read, Read, Read
- Reviews

Yellow Grade

PRINCIPLES
- Interface segregation principle
- Dependency inversion principle
- Liskov substitution principle
- Principle of least astonishment
- Information hiding principle

PRACTICES
- Automated unit tests
- Mockups
- Code coverage analyse
- Conferences
- Complex refactoring

Green Grade

PRINCIPLES
- Open closed principle
- Tell, don´t ask
- Law of Demeter

PRACTICES
- Continuous Integration
- Static code analysis
- Inversion of control container
- Pass experiences
- Measure bugs

Blue Grade

PRINCIPLES
- Design and implementation do not overlap
- Implementation reflects design
- You ain´t gonna need it (YAGNI)

PRACTICES
- Continuous Deployment
- Iterative development
- Component oriented
- Tests first

methodpark