**Паттерны проектирования**

**ИСиТ,ПОИБМС-2**

**Лекция 12**

**Чистый код (Clean Code)**

1. **Мартин Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста. — СПб.: Питер, 2011. — 464**
2. **Бьерн Страуструп:** *логика должны быть достаточно прямолинейной, чистый код хорошо решает одну задачу*.
3. **Грэди Буч**: ч*истый код прост и прямолинеен; чистый код читается, как хорошо написанная проза.*
4. **Дэйв Томас**: *чистый код может читаться и усовершенствоваться другими разработчиками; в чистом коде используются содержательные имена; для выполнения одной операции в нем используется один путь: код обладает минимальными зависимостями.*
5. **Майкл Физер**: *вы не найдете никаких очевидных возможностей для улучшения кода.*
6. **Рон Джеффрис**: *код* *содержит минимальное количество сущностей (классов, методов, функций и т.д.)*.
7. **Уорд Каннингем**: *вы работаете с чистым кодом, если каждая функция делает примерно то, что вы ожидали.*
8. **Правило бойскаута:** *оставь место стоянки чище, чем оно было до твоего прихода.*

**Именование**

1. **Именование:** имена должны передавать намерения программиста
2. **Именование:** имена переменных

******

1. **Именование:** имена классов и методов



1. **Именование:** избегать в именах дезинформацию (list, ht,…).
2. **Именование:** избегать символы l, o, O, 0.
3. **Именование:** осмысленное использование имен.

****

1. **Именование:** имена должны быть произносимы.
2. **Именование:** имена должны быть удобными для поиска (короткие имена часто повторяются); короткие имена только для локальных переменных.
3. **Именование:** схемы кодирования имен мешают;венгерская запись Windows API, префиксы для переменных и классов не применяют; IXXX – для интерфейсов, XXXFactory – абстрактные фабрики; i,j,k … - для локальных счетчиков.
4. **Именование:** имена методов - начинаются с глаголов: getXXX, postXXX, deleteXXX, copyXXX, scratchXXX, appendXXX, replaceXXX, payXXX, addXXX.
5. **Именование:** тезаурус терминов используемых в именах (Driver, Controller, Manager).
6. **Именование:** применение паттерн-имен (XXXFactory, XXXDecorator, XXXVisitor, …).
7. **Именование:**  применение структур-имен (Queue, HashTable, List, Stack).
8. **Именование:** контекст имен переменных, лучше если переменные являются элементами класса: Address.City, Address.Zip, Student.ID.
9. **Именование:** контекст имен методов, Students.GetAll, Students.Append.
10. **Именование:** контекст не должен быть избыточным Students.GetAllStudents, Student.StudentID.
11. **Именование:** используйте для контекста пространства имен.
12. **Функции:** главное положительное качество – компактность.
13. **Функции:** отступы в функции не более 3х; чем больше аргументов те сложнее функции.
14. **Функции:** правило одной операции.
15. **Функции:** операции должны соответствовать уровню абстракции самой функции.
16. **Функции:** код читается сверху вниз.
17. **Функции:** уровень абстракции понижаются; функция вызывает функции более низкого уровня абстракции.
18. **Функции:** функции с оператором **switch** опускаются на самый нижний уровень абстракции (содержат конкретные коды).
19. **Функции:** избегать применения возвращаемых аргументов.
20. **Функции:** количество аргументов не больше трех; лучше без аргументов; список аргументов считаем за 1 аргумент (List); если аргументов много, то надо упаковывать в классы.
21. **Функции:** стандартные унарные формы: преобразование, проверка, событие.

****

1. **Функции:** избегать аргументы типа bool.
2. **Функции:** не должно быть побочных эффектов.
3. **Функции:** лучше всего, если функция не возвращает значение, а меняет состояниеобъекта.
4. **Функции:** разделение на запросы и команды

****

****

1. **Функции:** генерация try/catch/finally исключений вместо кода возврата.
2. **Функции:** структурное программирование (Эдгар Дейкстра), go to, return, break, continue.
3. **Комментарии:** лучше без комментария, чем комментарий с ошибкой.
4. **Комментарии:** комментарий не компенсирует плохого кода.
5. **Комментарии:** комментарий должен пояснять бизнес-логику.
6. **Комментарии:** объявляет паттерн.
7. **Комментарии:** Open API, Javadoc,…
8. **Комментарии:** журнализменений в комментариях.
9. **Комментарии:** ссылки на авторов.
10. **Комментарии:** оставшийся закомментированный код – плохая практика.
11. **Комментарии:** заголовки функций.
12. **Форматирование:** важно**.**
13. **Форматирование:** вертикальное форматирование, упорядочивание классов и объектов.
14. **Форматирование:** горизонтальное форматирование, выравнивание, отступы.
15. **Объекты и структуры:**
16. Когда Игнац Земмельвейс в 1847 году впервые порекомендовал врачам мыть руки перед осмотром пациентов, его советы были отвергнуты на том основании, что у врачей слишком много работы и на мытье рук у них нет времени.