Чистые функции

Рефакторинг

Детерминированные функции

Детерминированная функция всегда возвращает одинаковое значение при определённом вводе

```
int SumAB(int a, int b) {
    return a + b;
}
```

```
int lastDigitOfNumber(int number) {
    return number % 10;
}
```

Недетерминированные функции

Недетерминированная функция *не* всегда будет возвращать одинаковое значение при определённом вводе.

Самый яркий пример функция rand()

```
int getNumberInRange(int minValue, int maxValue) {
    return rand() % (maxValue - minValue+1) + minValue;
}
```

```
int getRandomNumber() {
    return rand();
}
```

Побочные эффекты

В императивных языках некоторые функции в процессе выполнения своих вычислений могут модифицировать значения глобальных переменных, осуществлять операции ввода-вывода, реагировать на исключительные ситуации, вызывая их обработчики. Такие функции называются **функциями с побочными эффектами**. Другим видом побочных эффектов является модификация переданных в функцию параметров (переменных), когда в процессе вычисления выходного значения функции изменяется и значение входного параметра.

Примеры побочных эффектов

- Работа с консолью
- Работа с файлами
- Изменение глобальных переменных
- Генерация и обработка исключительных ситуаций
- Модификация аргументов функции

Чистые функции

Детерминированная + без побочных эффектов = чистая функция

Чистые функции:

- проще читать
- проще отлаживать
- проще тестировать
- не зависят от порядка, в котором они вызываются
- просто запустить параллельно (одновременно)

Задание

- Выбрать любую практическую или самостоятельную работу;
- Разделить задания на функции;
- В каждой функции выделить подфункции, которые выполняют какое-либо атомарное действие;
- Вынести из функций побочные эффекты, чтобы максимальное кол-во функций было чистыми;
- Переименовать переменные;
- Оформить отчёт в .pdf или .md о проделанной работе