Multi-paradigm and Meta-programming in the Software Engineering

Kiev, 2015-2022

Programming basics

Links

- Author: https://github.com/tshemsedinov
- Telegram channel: https://t.me/HowProgrammingWorks
- Tasks: https://github.com/HowProgrammingWorks/Index/blob/master/Exercises.ru.md
- Glossary: https://github.com/HowProgrammingWorks/Dictionary/blob/master/Fundamentals.ru.md

Module SEF1

36 lectures (4 introductory, 8 on the JS language, 18 on fundamental concepts of
programming, 6 on development tools and process), 8 seminars, 9 laboratory works,
47 repositories with code examples.

1. Introduction
1.1. Abstractions, modeling, reuse
1.2. Algorythm, programm, syntax, language
1.3. Value, identifier, variable and constant, literal, assignment
1.4. Data types, scalar, reference and structure types
1.5. Operator and expression, code block, function, loop, condition
1.6. Procedural paragidm, call, stack and heap
1.7. Application building blocks: modules, dependencies, libraries, components
1.8. Programming languages and paradigms, technology stack
1.9. Development environment and code debugging
2. JavaScript, Python and C syntax
3. Collections
3.1. Array, list, set, tuple
3.2. Dictionaries, hash table and assiciative array
3.3. Stack, queue, deque
3.4. Trees, Graphs
3.5. Structs and records
4. Basic concepts

4.1. Contexts and lexical scope, closures
4.3. Object, prototype, and class
4.4. Higher order function, pure function, side effects
4.5. Callbacks and events
4.5. Partial application and curry, pipe and compose
4.6. Chaining for for methods and functions
4.7. Error handling
4.8. Iterations and loops, recursion, iterators and generators
4.9. Mixins
5. Antipatterns
5.1. Common for all paradigms
5.2. Procedural anpipatterns
5.3. Object-oriented antipatterns
5.4. Functional antipatterns
6. Development process
6.1. Refactoring
6.2. Code review
6.3. Multi-aspect optimizations
7. Additional and auxiliary techniques
7.1. Events, Timers and EventEmitter
7.2. Serialization and deserialization
7.3. Regular expressions
7.4. Memoization
7.5. Factory and Poll
7.6. Wrappers
7.7. Typed arrays
7.8. Projections
7.9. I/O and Files

Consider following:

```
const id = (x) => x;

// Usage

const res = id(5);
console.log({ res });
```