

Цель, общая структура

Цель:

- ▶ После первого курса — участвовать в летних школах, быстро стартовать проект в начале второго курса
- ▶ После второго курса — быть полезными на стажировках, иметь кругозор, быть в целом готовыми на Junior-разработчика

Общая структура:

- ▶ 1 семестр — алгоритмы и структуры данных на C
- ▶ 2 семестр — ООП на C#
- ▶ 3 семестр — “продвинутое” программирование на C#
- ▶ САКОД — функциональное программирование на F#

Особенности реализации

- ▶ Ориентированность на промышленную разработку
- ▶ Начинаем с нуля, обычно слабые студенты
- ▶ В основном лекционные пары, без привязки к курсу “Информатика”
- ▶ Много домашки, проверка и общение в основном удалённо
- ▶ Отдельные часы на 2-м курсе на защиты практик
- ▶ Своя LMS, <http://hwproj.me/>

1 семестр

- ▶ Раздел 1, базовая алгоритмика
 - ▶ Введение в C/C++
 - ▶ Сложность алгоритмов, стильгайд, процесс компиляции, тестирование и отладка
 - ▶ Сортировки, git
 - ▶ Внутреннее представление данных. Файлы, структуры, указатели, .h/.c
 - ▶ О разработке программных продуктов. Контрольная
- ▶ Раздел 2, “динамические” структуры данных
 - ▶ Стеки, очереди, списки
 - ▶ Понятие абстрактного типа данных, ещё списки
 - ▶ Деревья вообще и деревья поиска
 - ▶ Самобалансирующиеся деревья: AVL, красно-чёрные, splay
 - ▶ Хеш-таблицы
 - ▶ Графы
 - ▶ Обзор парадигм программирования. Контрольная.
 - ▶ Доклады. Ещё про парадигмы программирования.
 - ▶ Автоматы, лексический анализ
 - ▶ Зачёт

2 семестр

- ▶ Введение в C#
- ▶ ООП вообще и в C#
- ▶ Модульные тесты
- ▶ Исключения и обработка ошибок
- ▶ CI, инструменты разработки, немного про лицензии
- ▶ Событийно-ориентированное программирование
- ▶ Пользовательские интерфейсы (WinForms)
- ▶ Контейнеры и генерики
- ▶ Контрольная
- ▶ Визуальное моделирование
- ▶ SOLID и общие рекомендации про хороший ООП-код
- ▶ Доклады
- ▶ Зачёт

3 семестр

- ▶ Многопоточное программирование
 - ▶ Многопоточное программирование “на низком уровне”: поддержка со стороны ОС, планировщик, Thread, гонки — **Lazy**
 - ▶ Прimitives синхронизации, кратко про модель памяти
 - ▶ Практика, “Обедающие философы” — **MyThreadPool**
 - ▶ Высокоуровневая многопоточность: пул потоков, async/await
- ▶ Сетевое программирование
 - ▶ Работа с сетью, низкий уровень (модель OSI, сокеты, консольные утилиты, клиент-сервер на .NET) — **SimpleFTP**
 - ▶ Работа с сетью, высокий уровень (HTTP вообще и в .NET, REST-сервисы, безопасность)
 - ▶ Практика, клиент для вконтакта
- ▶ Рефлексия — **MyNUnit**
- ▶ Контрольная
- ▶ Базы данных (вообще реляционные СУБД, SQL, кратко EF Core)
- ▶ GUI на WPF
 - ▶ Обзор библиотеки
 - ▶ Ещё немного обзора, практика: GUI для клиента вконтакта — **GUI для FTP**
- ▶ Веб-программирование — **MyNUnitWeb**
- ▶ Защита практик

4 семестр

- ▶ Функциональное программирование, введение в F#
- ▶ Ещё особенности F#
- ▶ Нетипизированное λ -исчисление
- ▶ Генерики в F#
- ▶ Объектно-ориентированное программирование в F#
- ▶ Вычислительные выражения в F#
- ▶ Многопоточное программирование в F#
- ▶ Синтаксический анализ на F#, часть 1 (синтаксический анализ вообще)
- ▶ Синтаксический анализ на F#, часть 2 (FParsec vs FsLex/FsYacc)
- ▶ Доклады
- ▶ Сборка мусора в .NET
- ▶ Защиты практик

Материалы

- ▶ Материалы курса прошлого года и задачи: <http://hwproj.me/courses/34>
- ▶ Исходники презентаций:
 - ▶ <https://github.com/yurii-litvinov/courses/tree/master/programming-1st-semester>
 - ▶ <https://github.com/yurii-litvinov/courses/tree/master/programming-2nd-semester>
 - ▶ <https://github.com/yurii-litvinov/courses/tree/master/programming-3rd-semester>
 - ▶ <https://github.com/yurii-litvinov/courses/tree/master/fsharp>
- ▶ Подробные программы и условия контрольных (просите доступ):
 - ▶ <https://docs.google.com/spreadsheets/d/107S8u65s6ml-2P20N665qg239zjqJP702Xiz3oDw0ok>
 - ▶ <https://docs.google.com/spreadsheets/d/1FMPQxGzAg70PkeV7RFkjLbs4CXi8LQZL4gtwMftHeak>
 - ▶ <https://docs.google.com/spreadsheets/d/1P7qGCpLluHMKs5ZQlnC4Bty73c94i9dYecpuc4Bv5LU>
 - ▶ <https://docs.google.com/spreadsheets/d/1NYwteG2ehKuhhhpE4BthSgCvB7HDnCb0EDUeyHsA1BA>