

Курс “Программирование”

Ю.В. Литвинов

y.litvinov@spbu.ru

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#
- ▶ 4 семестр — функциональное программирование на F#

Особенности реализации

- ▶ Ориентированность на промышленную разработку
- ▶ Не требуем предварительных знаний, но ожидаем умение самостоятельно искать и обобщать информацию
- ▶ В основном лекционные пары, на первом курсе также практика в аудитории
- ▶ Много домашних работ, проверка и общение в основном удалённо
- ▶ Своя LMS, <https://hwproj.ru/>
 - ▶ Нет автоматической проверки — ценим архитектуру и качество кода выше работоспособности, код всех решений просматривается вручную
 - ▶ Не отрицаем ценность автоматизации для базовых проверок, но пока руки не дошли

Связанные курсы

- ▶ Информатика — кругозор, общая теория, но все важные вещи мы разбираем ещё раз сами
- ▶ Дискретная математика — математические основы, основные алгоритмы; поддерживаем курс задачами, ещё раз обсуждаем теорию
- ▶ Учебные практики — даём для них необходимые знания, они нам — мотивацию учиться

1 семестр

- ▶ Раздел 1, базовая алгоритмика
 - ▶ Введение в C
 - ▶ Сложность алгоритмов, стиль, 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
 - ▶ Самобалансирующие 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
 - ▶ Прimitives синхронизации, кратко про модель памяти
 - ▶ Практика, “Обедающие философы”
 - ▶ Высокоуровневая многопоточность: пул потоков, async/await
- ▶ Сетевое программирование
 - ▶ Работа с сетью, низкий уровень (модель OSI, сокеты, консольные утилиты, клиент-сервер на .NET)
 - ▶ Работа с сетью, высокий уровень (HTTP вообще и в .NET, REST-сервисы, безопасность)
 - ▶ Практика, клиент для ВКонтакте
- ▶ Рефлексия
- ▶ Контрольная
- ▶ Базы данных (вообще реляционные СУБД, SQL, кратко EF Core)
- ▶ Веб-программирование
- ▶ Практика, приложение для регистрации на конференцию
- ▶ Контейнеризация и развёртывание, практика по Docker

4 семестр

- ▶ Функциональное программирование, введение в F#
- ▶ Языковые особенности F#
- ▶ Нетипизированное λ -исчисление
- ▶ Генерики в F#
- ▶ Объектно-ориентированное программирование в F#
- ▶ Вычислительные выражения в F#
- ▶ Многопоточное программирование в F#, событийное программирование, акторная модель
- ▶ Синтаксический анализ на F#, часть 1 (синтаксический анализ вообще)
- ▶ Синтаксический анализ на F#, часть 2 (FParsec vs FsLex/FsYacc)
- ▶ Доклады

Материалы

- ▶ Первый семестр:
<https://github.com/yurii-litvinov/courses/tree/master/programming-1st-semester>
- ▶ Второй семестр:
<https://github.com/yurii-litvinov/courses/tree/master/programming-2nd-semester>
- ▶ Третий семестр:
<https://github.com/yurii-litvinov/courses/tree/master/programming-3rd-semester>
- ▶ Четвёртый семестр:
<https://github.com/yurii-litvinov/courses/tree/master/structures-and-algorithms>

Материалы включают в себя TeX-овские исходники презентаций, конспекты (кроме первого семестра, там занятия в более интерактивном формате), условия домашних задач.