

# Лекция 0: Введение в программирование

Сергей Мыц

кафедра Информатики, БГУИР

предмет “Программирование”, ИиТП, первый курс,  
весенний семестр 2015

# Содержание лекции

## 1 Информатика

- Что это такое
- Как устроено
- Почему это интересно

## 2 Саморазвитие

- Как развиваться
- Зачем это всё

## 3 Разработка ПО

- Из чего состоит
- Инструменты
- Базисные идеи
- Ориентиры

## 4 Этот курс

- Что рассмотрим
- Правила игры
- Цели и профит

# Содержание

## 1 Информатика

- Что это такое
- Как устроено
- Почему это интересно

## 2 Саморазвитие

- Как развиваться
- Зачем это всё

## 3 Разработка ПО

- Из чего состоит
- Инструменты
- Базисные идеи
- Ориентиры

## 4 Этот курс

- Что рассмотрим
- Правила игры
- Цели и профит

## Варианты названий

- Информатика
- В англоязычной литературе как Computer science.
- На русском также "Компьютерные науки"

# Происхождение названия

Информатика = информация  $\times$  автоматика

# Про что она?

Про методы и алгоритмы автоматической работы с информацией

# Информация

Как её:

- собрать
- хранить
- обработать
- передать
- проанализировать
- оценить

# Зачем всё это делать с информацией?

Для принятия решений.



# Содержание

## 1 Информатика

- Что это такое
- Как устроено
- Почему это интересно

## 2 Саморазвитие

- Как развиваться
- Зачем это всё

## 3 Разработка ПО

- Из чего состоит
- Инструменты
- Базисные идеи
- Ориентиры

## 4 Этот курс

- Что рассмотрим
- Правила игры
- Цели и профит

# Структура информатики

Дальше посмотрим на разные составные части информатики

# Математические основы

- теория кодирования
- теория игр
- теория графов
- математическая логика
- теория чисел
- теория вероятности и математическая статистика

# Алгоритмы и структуры данных

- алгоритмы - процедуры последовательных и параллельных вычислений для решения широкого круга задач.
- структуры данных - организация и обращение с данными.

# Искусственный интеллект

- искусственный интеллект - создание и изучение систем с автономным интеллектом или самостоятельным поведением
- автоматический вывод - логический вывод на основе правил, доказательство теорем.
- компьютерное зрение - идентификация трёхмерных объектов по двумерному изображению.
- машинное обучение - автоматическое создание набора правил и аксиом на основе входных данных.
- обработка естественных языков - системы и алгоритмы анализа, понимания и генерации естественных (человеческих) языков
- робототехника - алгоритмы управления поведением роботов

# Связь и безопасность

- компьютерные сети - алгоритмы и протоколы для надёжного обмена данными через различные общие или выделенные среды
- компьютерная безопасность - практические аспекты безопасности компьютерных систем и сетей
- криптография - применение математического аппарата линейной алгебры, теории чисел, теории сложности, теории вероятности для создания и взлома шифров, анализа криптографических протоколов

# Архитектура компьютера

- архитектура компьютера - проектирование, организация и проверка компьютерной системы (процессор, память и другие подсистемы)
- операционные системы - системы предоставляющие основу для более простого создания и использования компьютерных программ

# Компьютерная графика

- компьютерная графика - алгоритмы для создания, изменения и интеграции графических изображений
- обработка изображений - получение информации из изображений с помощью вычислительных процедур



# Одновременные (concurrent) вычисления, параллельные (parallel) и распределённые системы

- консиггенсу - теория и практика одновременных вычислений; безопасность данных в многозадачной и многопоточной среде
- параллельные вычисления - вычисления с использованием множества потоков выполнения, разработка алгоритмов для решения проблем в мультипроцессорной среде с максимальным ускорением по сравнению с последовательным выполнением

# Базы данных

- реляционные базы данных - набор теоретических и алгоритмических основ построения реляционных баз данных
- структурированное хранение - нереляционные базы данных (например, NoSQL)
- извлечение данных (data mining) - алгоритмы для поиска, обработки и объединения информации из неорганизованных массивов документов и баз данных

# Теория вычислений

- теория автоматов - автоматные логические структуры для решения задач
- теория вычислимости - что вычислимо в рамках текущей модели компьютера
- теория вычислительной сложности - классы и границы сложности различных вычислений
- теория квантовых вычислений - модели вычислений с использованием квантовой суперпозиции битов

# Научные вычисления

- вычислительные методы - построение и оценка мат. моделей для решения научных задач с помощью компьютера
- численный анализ - приближенные численные решения математических задач (нахождение корней, интегрирование, решение уравнений, примерные решения особых функций)
- символьные вычисления - оперирование и решение выражениями в символьной форме (компьютерная алгебра)
- вычислительная физика - анализ и симуляция сложных физических систем
- вычислительная химия - вычислительная модель теоретической химии для определения химической структуры и свойств

# Языки программирования и компиляторы

- теория компиляторов - теория проектирования компиляторов, основана на теории автоматов
- прагматика языков программирования - таксономия языков программирования, сильные и слабые стороны, парадигмы
- теория языков программирования - проектирование, разработка, анализ и классификация языков программирования
- формальная семантика - строгая интерпретация языков путём описания в математических терминах
- теория типов - формальный анализ типов данных в программе и использование этих типов для понимания свойств программ (безопасность и т.п.)

# Разработка программного обеспечения (ПО) aka Software Engineering

- формальные методы - математические подходы для описания и выводов о проектировании ПО
- проектирование алгоритмов - применение идей теории алгоритмов для творческого создания решений для задач в реальной жизни
- разработка ПО - принципы и практики проектирования, разработки и тестирования программ, а также инженерные практики
- программирование - использование языков программирования для реализации алгоритмов

# Разработка программного обеспечения (ПО) aka Software Engineering

- человеко-компьютерное взаимодействие - изучение и проектирование компьютерных интерфейсов для использования людьми
- реверс инжиниринг (reverse engineering, обратное проектирование) - применение научного подхода для понимания и воспроизведения работы произвольных уже существующих программ

# Содержание

## 1 Информатика

- Что это такое
- Как устроено
- Почему это интересно

## 2 Саморазвитие

- Как развиваться
- Зачем это всё

## 3 Разработка ПО

- Из чего состоит
- Инструменты
- Базисные идеи
- Ориентиры

## 4 Этот курс

- Что рассмотрим
- Правила игры
- Цели и профит



# Почему информатика и программирование это круто?

Область стремительно развивается, постоянно появляется что-то новое, а старые идеи находят интересные и необычные применения.

# Почему информатика и программирование это круто?

- Судя по всему, это уже необратимо.
- Компьютерные технологии проникли во все сферы нашей жизни и со временем интеграция будет ещё более тесной.
- Перспективы развития впечатляют.

# Почему информатика и программирование это круто?

"Любая достаточно развитая технология неотличима от магии."

Артур Кларк, писатель фантаст и футуролог

# Почему информатика и программирование это круто?

И именно программисты - волшебники такого будущего.

# Содержание

## 1 Информатика

- Что это такое
- Как устроено
- Почему это интересно

## 2 Саморазвитие

- Как развиваться
- Зачем это всё

## 3 Разработка ПО

- Из чего состоит
- Инструменты
- Базисные идеи
- Ориентиры

## 4 Этот курс

- Что рассмотрим
- Правила игры
- Цели и профит

# Уровень сложности решаемых задач

В программировании как индустрии возможны очень разные задачи: от периодически выполняемых типовых до уникальных.

## Уровень сложности решаемых задач

Дальнейшие рассуждения рассчитаны на то, что человек хочет делать классные, необычные и сложные вещи.

# Уровень сложности решаемых задач

Для этого требуется уметь понимать и решать проблему на всех её уровнях.



# Какие навыки надо развивать, чтобы стать хорошим программистом?

Воспользуемся этапами работы с информацией как основными пунктами.

# Нужно уметь искать информацию

- поисковые системы
- википедия
- статьи в профессиональных блогах
- книги

# Открытые источники образования

- открытые курсы
  - mit opencourseware
  - coursera
  - lectorium.tv
  - и другие
- ШАД

# Общение

- онлайн форумы и профессиональные ресурсы
- конференции, доклады, рассказы
- личное общение

# Нужно уметь организовывать найденное

- личные заметки
- закладки
- мини-конспекты
- подобранная литература для справки

# Нужно иметь возможность обрабатывать

- иметь базис знаний в различных областях информатики
- уметь выделять важное
- английский

# Нужно делиться

- можно просто делиться друг с другом интересными и полезными идеями, подходами, книгами и статьями
- можно писать программы и библиотеки, которыми будут пользоваться другие люди (Open Source)
- можно писать статьи

# Нужно уметь анализировать

- математика (логика, математическая статистика, алгоритмы и структуры данных и т.д.)
- иметь инструменты для упрощения работы с данными и их агрегации



# Нужно уметь оценить

- критическое отношение к утверждениям без доводов
- где, что и как может сломаться и что сделать, чтобы этого не было
- необходимость постоянного выбора оптимального решения или подхода

# Нужно уметь самостоятельно решать сложные задачи

- часто условия заданы очень абстрактно
- незнакомая предметная область
- плохо сформулированные требования

# Нужно уметь самостоятельно решать сложные задачи

- если задача интересная (не типовая), то зачастую изначально вообще непонятно как решать
- необходимость использования новых теоретических и практических инструментов
- постоянный анализ новых данных и доработка подходов

# Нужно уметь самостоятельно решать сложные задачи

- планирование разработки и личных ресурсов
- консультации, обсуждения

# Содержание

## 1 Информатика

- Что это такое
- Как устроено
- Почему это интересно

## 2 Саморазвитие

- Как развиваться
- Зачем это всё

## 3 Разработка ПО

- Из чего состоит

- Инструменты
- Базисные идеи
- Ориентиры

## 4 Этот курс

- Что рассмотрим
- Правила игры
- Цели и профит

# Зачем делать это всё сейчас?

- Лишь часть часть из описанного выше есть в университетских курсах.
- От курса к курсу качество рассказа может разниться.
- В каждом отдельном предмете даются лишь части общей картины, которые нужно уметь собирать воедино.
- И постоянно достраивать и развивать.

# Зачем делать это всё в будущем?

- Постоянно появляются новые технологии и переосмысливается использование старых.
- Требуется постоянно быть в курсе и узнавать новое.

# Зачем делать это всё в будущем?

- Чем больше знания и умения, тем менее ограничены возможности программиста.
- И тем более интересные и сложные задачи он может решать.



# Содержание

## 1 Информатика

- Что это такое
- Как устроено
- Почему это интересно

## 2 Саморазвитие

- Как развиваться
- Зачем это всё

## 3 Разработка ПО

## • Из чего состоит

- Инструменты
- Базисные идеи
- Ориентиры

## 4 Этот курс

- Что рассмотрим
- Правила игры
- Цели и профит

# Этапы разработки ПО

- Исследование предметной области
- Сбор требований
- Проектирование
- Программирование
- Тестирование
- Поддержка
- Закапывание :)

# Содержание

## 1 Информатика

- Что это такое
- Как устроено
- Почему это интересно

## 2 Саморазвитие

- Как развиваться
- Зачем это всё

## 3 Разработка ПО

- Из чего состоит

## • Инструменты

- Базисные идеи
- Ориентиры

## 4 Этот курс

- Что рассмотрим
- Правила игры
- Цели и профит

# Инструменты

- Языки и парадигмы программирования
- Текстовые редакторы
- Компиляторы и интерпретаторы
- Внешние библиотеки

## Ещё инструменты

- Механизмы управления и конфигурирования сборки
- Системы контроля версий

# Не совсем инструменты

- Формат обмена данными
- Утилиты командной строки
- Скриптовые языки

# Снова инструменты

- Отладчики
- Средства автоматизации тестирования

# Больше инструментов

- Инструменты статического анализа кода
- Профилировщики
- Средства непрерывной интеграции
- Средства генерации кода (генераторы парсеров, макросы и т.п.)



# Больше инструментов

- логгирование
- мониторинг состояния
- механизмы аналитики

# Больше инструментов

- IDE
- Редакторы графических интерфейсов
- Генераторы документации

# Больше инструментов

## Языки разметки

- html
- разметка интерфейсов
- markdown
- latex

# Содержание

## 1 Информатика

- Что это такое
- Как устроено
- Почему это интересно

## 2 Саморазвитие

- Как развиваться
- Зачем это всё

## 3 Разработка ПО

- Из чего состоит

- Инструменты

- **Базисные идеи**

- Ориентиры

## 4 Этот курс

- Что рассмотрим

- Правила игры

- Цели и профит

# Программирование как понимание

Понимаю, если могу запрограммировать.

# Базисные идеи

- управление сложностью
- работа с абстракциями
- читабельность кода и принцип наименьшего удивления

# Некоторые принципы

- KISS - keep it simple, stupid - не усложняй, глупец
- DRY - don't repeat yourself - не повторяйся
- YAGNI - you ain't gonna need it - тебе это не понадобится
- NIH - not invented here syndrome - синдром "изобретено не здесь избегай велосипедов

# Простая метрика качества кода

WTF per minute - частота удивлений и непониманий при чтении кода



# Обобщённые принципы

- Simplicity - простота
- Clarity - ясность
- Generality - общность
- Automation - автоматизация

# Содержание

## 1 Информатика

- Что это такое
- Как устроено
- Почему это интересно

## 2 Саморазвитие

- Как развиваться
- Зачем это всё

## 3 Разработка ПО

- Из чего состоит

- Инструменты

- Базисные идеи

- Ориентиры

## 4 Этот курс

- Что рассмотрим

- Правила игры

- Цели и профит

## Матрица компетентности программиста

# Содержание

## 1 Информатика

- Что это такое
- Как устроено
- Почему это интересно

## 2 Саморазвитие

- Как развиваться
- Зачем это всё

## 3 Разработка ПО

- Из чего состоит
- Инструменты
- Базисные идеи
- Ориентиры

## 4 Этот курс

- Что рассмотрим
- Правила игры
- Цели и профит

- языки программирования С и С++
- инструменты программирования
- командная строка на базе ОС Ubuntu Linux

# Содержание

## 1 Информатика

- Что это такое
- Как устроено
- Почему это интересно

## 2 Саморазвитие

- Как развиваться
- Зачем это всё

## 3 Разработка ПО

- Из чего состоит
- Инструменты
- Базисные идеи
- Ориентиры

## 4 Этот курс

- Что рассмотрим
- Правила игры
- Цели и профит

# Структура занятий

- лекции
- практические (лабораторные) занятия

# Задания

- лабораторные работы
- практические задания



# Лабораторные работы

- выдаются централизованно на лекции
- дедлайн (срок сдачи) – 1-2 недели
- заочный и очный этапы сдачи

## Лабораторные работы: заочная сдача

Сначала надо пройти онлайн ревью кода и получить одобрение преподавателя.

## Лабораторные работы: очная сдача

- запуск и демонстрация
- защита и устные ответы по теме и коду
- по результатам выставляется оценка от 1 до 10

# Практические задания

- даётся задание малой или средней сложности
- чтобы успевалось вдумчиво решиться за одно занятие
- хорошо может подойти рассмотрение отдельного инструмента или практики
- учитывается факт выполнение задания (зачёт), он равен одному доп. баллу

## Ход семестра

- все занятия и задания разделены на части точками контроля (около месяц)
- в точке контроля фиксируется состояние сданных лабораторных, практических и посещение
- после неё сдача лабораторных происходит по ведомостям для пересдач (либо по уважительной, либо без уважительной)

## Ход семестра: аттестации

По результатам работы происходит промежуточная аттестация студента:

- если выполнены обязательные требования
  - студент успешно аттестуется, получает оценку от 4 до 10 в зависимости от результатов
  - эта оценка подаётся в рамках факультетского контроля текущей успеваемости
- если требования не выполнены
  - студент получает оценку от 0 до 3 в зависимости от результатов эта оценка со всеми пропусками подаётся в рамках факультетского контроля текущей успеваемости

# Экзамен

От набранных баллов в течение семестра зависит:

- допуск к экзамену
- простота сдачи

# Содержание

## 1 Информатика

- Что это такое
- Как устроено
- Почему это интересно

## 2 Саморазвитие

- Как развиваться
- Зачем это всё

## 3 Разработка ПО

- Из чего состоит
- Инструменты
- Базисные идеи
- Ориентиры

## 4 Этот курс

- Что рассмотрим
- Правила игры
- Цели и профит



# Чего хочу добиться в общем

Понимания разных уровней абстракции и связи между различными областями знаний и языками.

# Чего хочу добиться в общем

Понимания роли программирования и возможного применения своих знаний.

# Какие навыки возможно приобрести

- читать код на Си
- писать простые прикладные программы на Си

# Какие навыки возможно приобрести

- читать простой код на C++
- писать простые прикладные программы на C++
- использование библиотек для Си и C++

# Какие навыки возможно приобрести

Владение инструментами, упрощающими разработку качественных программ.

# А если кому-то это не нужно?

- Нужно.
- Закон дырявых абстракций.