



Российское
общество
Знание

Всероссийская просветительская акция
«Поделись своим знанием»

Знание.Лекторий

Учебный процесс Института математики и
информационных технологий ПетрГУ

Ю. А. Богоявленский, к.т.н., доцент

Петрозаводский Государственный Университет
Институт математики и информационных технологий
Кафедра Информатики и математического обеспечения

Учебный процесс ИМИТ. Кафедра ИМО

Заведующий кафедрой Информатики и Математического
обеспечения, к. т. н., доцент,

Ю. А. Богоявленский
ybgv@cs.karelia.ru
<http://cs.karelia.ru>



Содержание

- Введение.
- Математика и информатика.
- Математики — лауреаты премии Тьюринга.
- Профессиональные задачи.
- Почему математическая культура полезна профессионалам ИКТ.
- Определяющая роль математической культуры в сфере ИКТ.
- Как организовать учебную работу.

Содержание

- Современные учебные планы.
- Информационно-вычислительная инфраструктура ИМИТ.
- ОС UNIX → Linux.
- Программирование “в малом”. Язык C на все времена.
- Командный интерфейс против графического.
- От учебы к профессиональной карьере.

Введение. Решение задач.

Новый важный этап интеллектуального развития.

Цель студента — сформировать культуру и навыки решения профессиональных **задач**.

- **Мотивация.** Очень хотеть решить задачу.
- **Знания.** Владеть объектами, их свойствами и связями предметной области.
- **Анализ.** Уметь выполнять исчерпывающий анализ условий и требований постановки задачи.
- **Синтез.** Строить путь от исходных данных к решению задачи.

Введение. Математика и информатика

Математика — формирует культуру решения задач.

Крайне полезный ресурс — [Математическая энциклопедия](#)

Центральное понятие — [Алгоритм](#)

«Dixit Algorizmi» (Сказал Аль-Хорезми) — из переводов на латынь («De numero Indorum» (Индийский счет), XII век) трудов IX века математика Мухаммеда ибн Муса аль-Хорезми.

1936 г. Теория (стандартизация) алгоритмов. Математики:

- **Алан Тьюринг (Alan Turing)** Машина Тьюринга;
- **Эмиль Пост (Post Emil)** Машина Поста;
- **Алонзо Чёрч (Alonzo Church)** λ -исчисление;
- **Андрей Андреевич Марков (младший)** Нормальные алгоритмы Маркова.

Черная овца в Австралии.

[Очень полезная книга. Д. Пойа Как решать задачу](#)

Математики и архитектура ЭВМ

Джон фон Нейман (John von Neumann) основной автор работы:

Burks A. W., Goldstine H. H., Neumann J. Preliminary Discussion of the Logical Design of an Electronic Computing Instrument. —Institute for Advanced Study, Princeton, N. J., July 1946.

В этой работе были сформулированы основные принципы логической структура ЭВМ, названные позже **«Архитектура фон Неймана»** и реализованные в десятках ЭВМ по всему миру.

В 1946 г. он возглавил проект по созданию ЭВМ IAS с этой архитектурой

Алан Тьюринг в 1946 г. разработал проект ЭВМ ACE (Automatic Computing Engine, Автоматическая вычислительная машина) с хранимой в памяти программой. Проект был реализован в 1950 г.

Подавляющее большинство современных ЭВМ поддерживают **Архитектуру фон Неймана.**

Математики — лауреаты премии Тьюринга

Премия Тьюринга (**Turing Award**) — самая престижная премия в информатике, вручаемая с 1966 г. Ассоциацией вычислительной техники (Association for Computing Machinery **ACM**)

- 1971.** Джон Маккарти — автор термина «искусственный интеллект» (1956), языка **Лисп**, парадигмы **функционального программирования**.
- 1977.** Джон Бэкус — руководитель разработки языка программирования **Фортран**, изобретатель формы **Бэкуса — Наура** для определения синтаксиса формальных языков.
- 1983.** Деннис Ритчи — один из авторов языка **C**, соавтор учебника **K&R**, участник разработки **ОС Unix**.
- 1986.** Роберт Тарьян — автор множества **алгоритмов на графах**, соавтор структур данных **“Фибоначчиева куча”** и **“Расширяющееся дерево”**.

Математики — лауреаты премии Тьюринга

- 1999.** Фредерик Брукс (младший) — автор бестселлера “**Мифический человеко-месяц**” и “закона Брукса”: *Если проект не укладывается в сроки, то добавление рабочей силы задержит его еще больше..* Возглавлял разработки семейства мейнфреймов **IBM System/360** (теперь – IBM zSeries) и их операционной системы **OS/360**.
- 2002.** Рональд Ривест, Ади Шамир и Леонард Адлеман — авторы системы шифрования (**RSA Rivest — Shamir — Adleman**, 1977 год), которая широко применяется, в том числе в защищенном веб протоколе **https**.
- 2004.** Винтон Серф — вместе с Робертом Каном под руководством Леонарда Клейнрока в 1974 г. разработал семейство протоколов **TCP/IP**.
- 2013.** Лесли Лэмпорт — заложил основы теории распределённых систем (**автоматный подход, логические часы, подпись Лэмпорта**). Автор настольной издательской системы **L^AT_EX**, лауреат еще шести премий.

Математики — лауреаты премии Тьюринга

2020. Джеффри Ульман — его учебники по **компиляторам, теории вычислений и базам данных** считаются стандартом.

Не лауреат премии Тьюринга

1988. Ван Якобсон — разработал 4 алгоритма работы протокола TCP (в том числе **AIMD**), позволившие предотвращать коллапсы перегрузки Интернет в 1980 г.г. и перейти к ее широкому внедрению.

См. [M. Allman, V. Paxson, E. Blanton Request for Comments: 5681 TCP Congestion Control](#)

Выводы

- **ЭВМ по сути** есть математические машины (**МСРПУ и ЭВМ!**). Они разрабатывались для ускорения вычислений по алгоритмам на идейной основе машины Тьюринга, под руководством математиков и согласно разработанной ими архитектуре.
- Основные достижения по развитию современных ИКТ получены математиками или учеными, владеющими математической культурой.

Профессиональные задачи.

■ Алгоритмы

- ▶ Знать основные классы. Владеть основами оценки сложности.
- ▶ Выбирать, комбинировать, модифицировать, разрабатывать.

■ Структуры данных

- ▶ Знать базовые структуры и их свойства.
- ▶ Понимать алгоритмы работы с элементами и их временную сложность.
- ▶ Выбирать, комбинировать, применять.

■ Компьютерные сети

- ▶ Знать уровни абстракции протоколов.
- ▶ Разрабатывать сетевые системы.

Компетенции ИКТ = Математическая + Инженерная (технология разработки ПО)

Профессиональные задачи.

■ Web технологии

- ▶ Знать проткол http, языки html, css, xml.
- ▶ Владеть навыками разработки на языках PHP, JavaScript.
- ▶ Применять программные каркасы (frameworks).

■ Технология разработки программного обеспечения

- ▶ Знать модели жизненного цикла продукта.
- ▶ Владеть технологиями и инструментами командной разработки.
- ▶ Формировать, моделировать анализировать требования к продукту.
- ▶ Выбирать архитектуру. Оценивать стоимость. Кодировать.
- ▶ Тестировать. Оценивать качество. Сопровождать.

■ Информационные системы

- ▶ Владеть теорией СУБД и навыками их применения
- ▶ Выбирать, проектировать архитектуру.
- ▶ Эксплуатировать. Сопровождать.

Почему математическая культура полезна профессионалам ИКТ

- Математика — язык описания и исследования в ИКТ.
 - ▶ Формирует и развивает абстрактное логическое мышление.
 - ▶ Выбатывает навык регулярного подхода к решению задач.
 - ▶ Приучает не игнорировать “мелочи”.
- Абстракция — математика — информатика.
- Стремительное расширение сферы ИКТ — инвестиции. Калейдоскоп новых парадигм, концепций, стандартов, инструментов, прикладных системы
- Базовые математические идеи появляются **гораздо реже**. Пример — распараллеливание и не Неймановские архитектуры.
- Математическая культура позволяет профессионалу **быстро осваивать** любые современные и будущие концепции, методы и технологии ИКТ, обеспечивая ему **долговременную** карьеру.

1959 г. ЭВМ Урал-2



1959 г. ЭВМ Урал-2. Характеристики

- 5000 целочисленных операций/сек;
- Оперативная память 4096 20-разрядных слов на магнитных сердечниках, возможен доступ к 40-разрядным словам.
- Ввод данных и команд с перфорированной киноленты.
- Вывод на печатающее устройство со скоростью до 20 чисел/с.
- Количество ламп 2100.
- Потребляемая мощность 25 кВт.
- Занимаемая площадь 150—200 м²

Определяющая роль математической культуры в ИКТ

Важно владение именно математической культурой.

... доказательство, контрпример, необходимое и достаточное условие ...

Математический анализ — база ВСЕХ направлений математики.

Владение его основами и формирует математическую культуру и позволяет легко осваивать и применять нужные разделы математики:

- Дискретная математика (в т.ч. комбинаторика) — большинство задач, в т.ч:
 - ▶ Формальные языки — gcc, компиляторы компиляторов.
 - ▶ Графы, деревья, очереди, стеки и др.
 - ▶ Поиск, сортировка, кратчайшие пути, оптимизация, маршрутизация в компьютерных сетях ... !
 - ▶ Криптография, оптимизация.
- Линейная алгебра — Игры, вирт реальность.
- Теория вероятностей — финансы, Data Mining.
- Геометрия, дифф. уравнения — Машинная графика, игры.
- Линейная алгебра, мат. статистика — Криптография.

Определяющая роль математической культуры в ИКТ

Из книги:

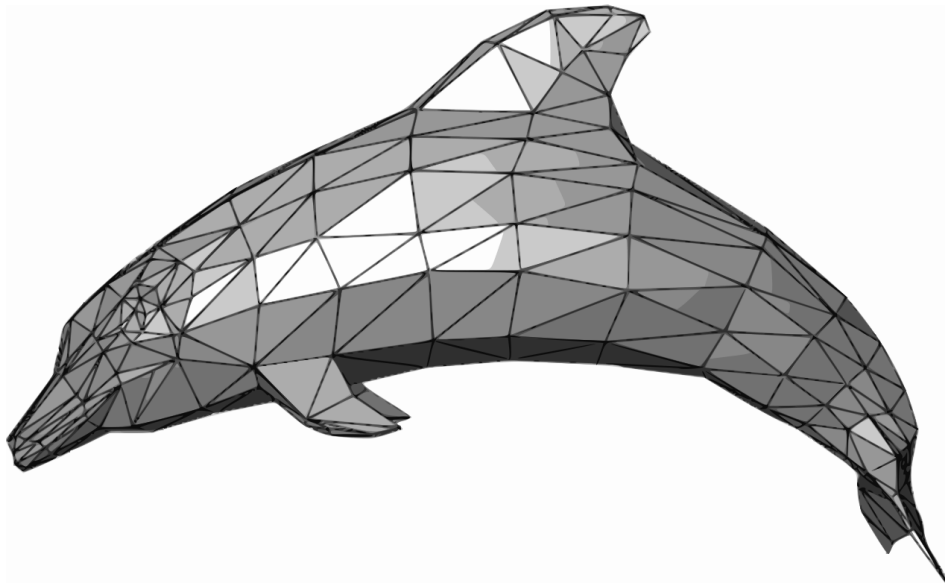
Д. ПОЙА МАТЕМАТИКА И ПРАВДОПОДОБНЫЕ РАССУЖДЕНИЯ

Математик, физик и инженер.

— Физик верит, — сказал математик, — что 60 делится на все числа. Он замечает, что 60 делится на 1, 2, 3, 4, 5 и 6. Он проверяет несколько других чисел, например 10, 20 и 30, взятых, как он говорит, наугад. Так как 60 делится и на них, то он считает экспериментальные данные достаточными.

— Да, но взгляни на инженера, — возразил физик. — Инженер подозревает, что все нечётные числа простые. Во всяком случае, 1 можно рассматривать как простое число, доказывает он. Затем идут 3, 5 и 7, все, несомненно, простые. Затем идет 9 — досадный случай; 9, по-видимому, не является простым числом. Но 11 и 13, конечно, простые. Возвратимся к 9, — говорит он, — я заключаю, что 9 должно быть ошибкой эксперимента.

Модель изображения дельфина: треугольная сетка



Как организовать учебную работу

Я слышу и забываю,
Я вижу и запоминаю,
Я делаю и понимаю.
Конфуций
Мыслитель и философ
5 в до н.э.

Профессионализм в ИКТ — **только** за счет большого объема работы по решению учебных задач.

- Учебный процесс интенсивный, **много нового**.
- **Работоспособность**: режим дня, сон, питание, физкультура.
Коты VK.
- **Меньше** прямого контроля, **больше** личной ответственности.
- Трудолюбие! Усердие!.

Как организовать учебную работу

■ Лекции.

- ▶ Активное восприятия — залог успеха в сессии, следить за ходом лекции, **задавать вопросы**.
- ▶ Конспект — обязательно, записывать новое, главное, перечитать перед следующей лекцией.

■ Лабораторные и практических занятия.

- ▶ Сдавать в срок, иметь конспекты и учебники.
- ▶ Иметь план работы - схему программы, тестовые задачи и т.п.
- ▶ Настроить среду (предоставляется кафедрой ИМО) для разработки и отладки программ в месте проживания.
- ▶ **Прояснять все непонятое, не откладывая, на консультациях у преподавателей (будет расписание) или по эл.почте .**

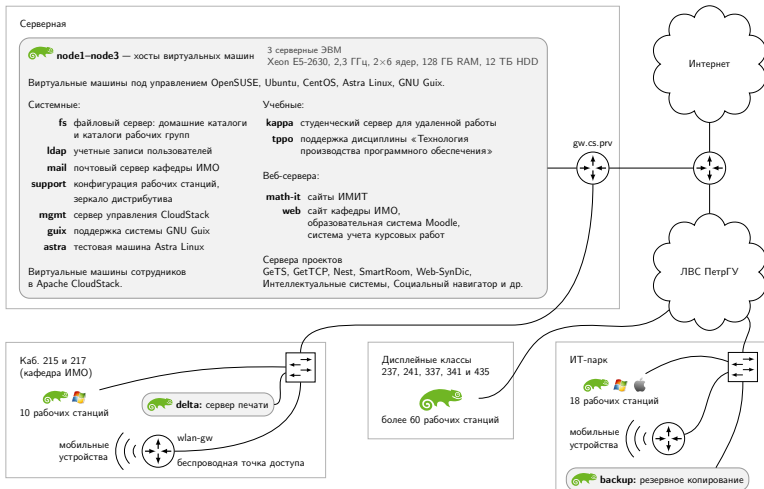
Современные учебные планы

- Бакалавриат с 1993 г..
- Магистратура с 1997 г..
- Аспирантура — давно.
- ФГОС + Computing Curricula, регулярно публикуемые Association for Computing Machinery — ACM и Institute of Electrical and Electronics Engineers Computer Society — IEEE CS.
- Компетенции ИКТ = Математическая + Инженерная (технология разработки ПО).
- ИМИТ поддерживает УП на современном уровне.
- История развития учебных планов. Anatoly Voronin, Iurii Bogoiavlenskii, Vladimir Kuznetsov "Perspectives on the emergence of computing programs propelled by local industry in Russia", ACM Inroads, December 2015, Volume 6, No. 4.

Информационно-вычислительная инфраструктура ИМИТ (ИВИ)

<http://cs.petrstu.ru/facilities/system.pdf>

Информационно-вычислительная инфраструктура Института математики и информационных технологий
Поддерживается полный набор инструментов для учебного процесса и НИР/ОКР студентов и сотрудников (около 600 пользователей).



Информационно-вычислительная инфраструктура ИМИТ (ИВИ)

- Развитая, современная.
 - ▶ ОС Linux для студентов — готовый образ — запускается в виртуальной машине (работает как приложение в вашей ОС).
 - ▶ Краткое руководство по Emacs (применяется и как IDE).
 - ▶ Использование отладчика.
 - ▶ Система видео-конференц-связи Jitsi Meet.
- Поддерживается сотрудниками кафедры ИМО — В. А. Пономарев, М. А. Крышень, В. М. Димитров, Д. Б. Чистяков.
- Предоставляет современные программные продукты и технологии для учебного процесса, исследований и разработок.
- Кафедрой подготовлены — **Инструменты для студентов**
- Нужно изучить — **Описание ИВИ**

ОС UNIX → Linux

- Разработана “Программистами для программистов”.
- Переносимая, многозадачная, многопользовательская.
- Первая ОС с реализацией набора протоколов TCP/IP.
- Первая ОС на ЯВУ. Более 95% кода реализовано на языке C.
- Для переноса на другую архитектуру достаточно разработать кросс компилятор.
- Перенесена на много архитектур - Интернет вещей, котлы ТЭЦ.

ОС UNIX → Linux

- Надежная - сервера эл.почты работают без перезагрузки месяцами.
- UNIX → MS DOS → Windows - Subsystem for Linux → Linux от Microsoft.
- Linux — лидер на рынке:
 - ▶ серверных ЭВМ;
 - ▶ смартфонов, планшетов;
 - ▶ высокопроизводительных вычислений (доля 99.4%.);
 - ▶ ОС сетевого оборудования (SOHO-роутеры, энтерпрайз-решения (Cisco));
 - ▶ ОС Аврора, бывшая Sailfish Mobile OS RUS;

Программирование "в малом". Язык С на все времена.

- Большое достижение авторов — компактность, лаконичность. Отсутствуют встроенные средства для выполнения распространенных операций: низкоуровневой или форматированный ввод/вывод, управление строками ... тысячи их и количество растет.
- Подобные задачи, например управление памятью, процессами, файловыми системами, сетевыми соединениями и т.п реализованы в стандартной библиотеке **libc** языка С
- В книге 1978 г. Брайана Кернигана и Денниса Ритчи “Язык программирования С” “**K&R**” — систематическое описан этот подход (40 изданий в США). Он оказались очень плодотворным и широко используется по настоящее время.

Программирование "в малом". Язык C на все времена.

- Язык и библиотека `libc` (входит в любую реализацию языка) в 1989 г. были стандартизованы ANSI, в 1990 — ISO. Стандарт обеспечивает кросс платформенную разработку.
- Регулярно занимает 1/2 места по популярности в [TIOBE Index](#)
- Существует несколько десятков реализаций `libc`, в том числе такие широко применяемые как:
 - ▶ **GNU C Library** - `glibc` - самая распространенная, для ОС Linux;
 - ▶ **Microsoft C Run-time Library**, используемая в ОС Windows;
 - ▶ **libSystem**, используемая в ОС iOS компании Apple;
 - ▶ **Bionic**, разработанная компанией Google для ОС Android.

Программирование "в малом". Язык С на все времена.

- Эффективный машинный код. Используется вместо ассемблера.
- Применяется при разработке как ОС (UNIX), так и прикладного ПО для платформ от суперЭВМ до встроенных.
- Синтаксис и структуры управления С **СТАЛИ КЛАССИКОЙ** и заимствуются языками C++, C#, Unix's C shell, D, Go, Java, JavaScript, Limbo, LPC, Objective-C, Perl, PHP, Python, Rust, Swift и др.
- Компиляторы и интерпретаторы языков программирования, СУБД и пр. реализуются, как правило, на С.
- Реализован для большинства аппаратных платформ и ОС.
- **С – фундаментальная основа квалификации программиста.**

Характеристики glibc

Общая характеристика glibc на 25.04.2021 - **25.07.23** со стр.
<https://www.openhub.net/p/glibc>

Code Locations: [git://sourceware.org/git/glibc.git](https://sourceware.org/git/glibc.git)

In a Nutshell, GNU C Library...

has had 37,147 - **40307** commits made by 604 - **720** contributors representing 1,302,579 - **1419588** lines of code

is mostly written in C with an average number of source code comments: Across all C projects on Open Hub, 19% of all source code lines are comments. This holds true for GNU C Library as well.

has a well established, mature codebase maintained by a very large development team with increasing Y-O-Y commits

took an estimated 371 - **406** years of effort (COCOMO model) starting with its first commit in February, 1995 ending with its most recent commit 7 months ago

Характеристики glibc

Основополагающе повлияла на реализации большинства современных системы программирования C++, Python, Rust и др.

Заголовочные файлы **libc** включены в стандартную библиотеку языка C++ под именами, созданными путём отсечения в имени файла расширения `.h` и добавлением символа `'c'` в начале имени, например имени `'time.h'` соответствует `'ctime'`.

Библиотека **glibc** обновляется два раза в год. Ее функции, требующие работы в адресном пространстве ядра, используют системные вызовы

В руководстве к **glibc** функции разбиты на 36 групп. Информацию о версии **glibc**, установленной в ОС Linux можно получить командой:

```
/lib/libc.so.6
```

Функции libc – фундаментальная основа всех современных программных продуктов

Научная Библиотека GNU Scientific Library

GNU GSL это библиотека программ численных методов на языках C и C++. На середину 2022 г. реализовано более 1000 функций с обширным набором тестов.

Начало — май 1996 г., регулярная работа до gsl-2.7 в июне 2021 г.

GSL разработана на платформе GNU/Linux с компилятором gcc, сообщалось, что она компилируется на 11 платформах.

... done 10,819 commits by 20 contributors representing 598,767 lines of code

... took an estimated 162 years of effort (COCOMO model)

Алгоритмы из 27 предметных областей математики, например ...

Complex Numbers Roots of Polynomials Special Functions Vectors and
Matrices Permutations Combinations Sorting Linear Algebra Root-Finding
Fast Fourier Transforms Eigensystems Random Numbers

Командный интерфейс против графического

- Письменность. Иероглифы против алфавита.
- Command Line Interface — CLI — интерфейс командной строки для взаимодействия с ОС.
- Популярный — `bash` — свободная версия Bourne shell.

Простые примеры:

- Цикл в командной строке `bash`:
`for d in *.txt ; do mv $d $d.utf8 ; done`
- Простой скрипт в файле `k-m` — монтировать (присоединить) домашний каталог на удаленном сервере `каппа` в каталог `КАППА` на локальной ЭВМ):

```
#!/usr/bin/sh
cd
sshfs kappa.cs.karelia.ru: KAPPA
```


Командный интерфейс — конвейер команд

Символ | — соединение в конвейер.

Конвейер (pipe) — однонаправленный канал взаимодействия команд

```
bash du -k -c -d 1|sort -n -r|less
```

- **du - disk usage:** **-k:** блоки по 1k,, **-c:** получить суммарный объем каталога, **-d 1:** для подкаталогов ниже на 1 уровень;
- **sort - сортировать строки:** **-n:** числовая сортировка, **-r:** в обратном порядке.

```
2890548 total
2890548
2765996  ./Lectures
84592   ./Intel-guides
9736    ./GCC-guide
5528    ./My-tests
4100    ./GDB
28      ./Coding-Style
12      ./Voprosy-k-zachetu
8       ./Profilers
lines 1-10/10 (END)
```

Командный интерфейс: сложная работа с файлами

Дано: 150 УМКД дисциплин кафедры, каждый в 5 файлах lowriter.

Нужно: каждый из них перевести в pdf и объединить в один pdf.

```
#!/bin/bash
# M. A. Kryshen. Getting pdf files of UMKD
# Target directory
base="$(pwd)"
# For each directory
find "$source" -type f | xargs -d '\n' -n 1 dirname | sort -u | \
  while read name; do
    echo "$name"
    dest="$base${name:${#source}}"
    mkdir -p "$dest"

    libreoffice --headless --convert-to pdf --outdir "$dest" "$name"/*

    # Titele sheet to the begin
    mv "$dest"/*titul_list*.pdf "$dest"/000_titul_list.pdf
    # Adding empty page after title
    ln empty.pdf "$dest"/001_empty.pdf
    pdfunite "$dest"/*.pdf "$dest".pdf
    rm -f "$dest"/*.pdf
    rmdir "$dest"
done
```

От учебы к профессиональной карьере

- https://imit.petrSU.ru/files/2018/f_248_dubna.pdf
А. В. Воронин Развитие ИТ - технологий в Петрозаводском государственном университете.
- <https://imit.petrSU.ru/employers>
Работа в сфере ИКТ и ПО. Компании партнеры ИМИТ.
- <https://imit.petrSU.ru/page/vacancy>
Предложения компаний-партнеров.

Магистры выпуска 2011 г.



Спасибо за внимание.

"... Важнейшим из всех искусств является для нас искусство программирования на языке Ассемблера..."

Ю.А.Богоявленский
"О грядущих катастрофах"

ДРАМА

НА

СИСТЕМНОЙ

МАГИСТРАЛИ.

(с) зр.22302, 1991г.

