

## Темы семинаров

### Тема 1: Визуализация информации. (09.09.20)

**Стендовый доклад** — форма представления информации, благодаря которой можно за короткий промежуток времени донести разноплановые сведения большого объема.

**Стендовые доклады используются:**

- Конференции
- Семинары
- Презентации

Преимущества	Минусы
На изучение СД можно потратить столько времени сколько необходимо	Некомфортное расположение зрителей
Есть возможность обсудить вопросы с автором исследования	
Содержит огромное количество информации	

**Требования:**

- Наглядность
- Оптимальность
- Доступность

**Этапы создания:**

1. Планирование
2. Макет
3. Оформление
4. Раздаточный материал
5. Презентация

**Советы:**

- Логическая цепочка: от левого верхнего угла до нижнего правого
- Разделение: вертикальные и горизонтальные секции, выделение логических блоков
- Выделение: названия разделов и основных позиций
- В верхней части — название, фамилия и имя автора
- В правом верхнем углу — фотография автора, эмблема заведения, другой иллюстративный материал
- Нижняя часть — контактная информация, благодарности, литература
- Цвета: не более 2-3 цветов
- Шрифт: не менее 20-24 кегля, свободно читаемый с расстояния 50-100 см
- Раздаточный материал: копии СД в формате А4

### **Структура:**

- Название, авторы, учреждения
- Краткое введение
- Цель исследования
- Материалы и методы исследования
- Результаты исследования, рисунки и графики
- Заключение и вывод

### **Программы:**

- Любой графический редактор
- Онлайн-сервисы для создания постеров: [canva.com](https://canva.com), [piktochart.com](https://piktochart.com) и т.д.
- MS PowerPoint и аналоги

## **Тема 2: Инструменты управления задачами и проектной работой. (14.09.20)**

**Управление задачами** — организация распределения заданий между участниками какой-либо группы. В качестве группы может выступать один и тот же человек. В этом случае, система управления заданиями будет представлять собой личный органайзер.

### **Элементы задачи:**

- Название
- Описание (примечания/заметки)
- Приоритет
- Время выполнения
- Приложения (attachments)
- Url
- Временные настройки (дата, время, повторяемость, длительность)
- Теги (tags)
- Списки (lists, trackers) («дом», «работа»)
- «Облако тегов»
- Режим доступа (частные, публичные списки)
- Контакты/команды
- Места (geolocations)

### **Функции ПО для управления задачами:**

- Планировщик заданий и самостоятельный или сторонний
- (другие пользователи) контроль за их выполнением
- Создание событий, привязанные к дате и времени
- Напоминания об определённых пользователем событиях
- Календарь
- Менеджер контактов
- Организация публичного доступа к задачам

**Управление проектами** — применение методов, инструментов, техник и компетенций к проекту.

**Проект** — уникальный набор процессов, состоящих из скоординированных и управляемых задач с начальной и конечной датами, предпринятых для достижения цели.

**Цель проекта:** достижение цели проекта требует получения результатов, соответствующих определенным заранее требованиям, в том числе ограничения на получения результатов, таких как время, деньги и ресурсы.

**Диаграмма Ганта** — один из способов отслеживания временных затрат.

**Создание диаграммы:**

- Realtimeboard ( <https://realtimeboard.com/ru/> )
- Teamweek ( <https://teamweek.com> )
- Canva ( [https://www.canva.com/ru\\_ru/grafiki/diagramma-ganta/](https://www.canva.com/ru_ru/grafiki/diagramma-ganta/) )

### Тема 3: Языки и среды программирования. (17.09.20)

**Язык программирования** — формальный язык, предназначенный для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (обычно — ЭВМ) под её управлением.

Jan 2020	Jan 2019	Change	Programming Language	Ratings	Change
1	1		Java	16.896%	-0.01%
2	2		C	15.773%	+2.44%
3	3		Python	9.704%	+1.41%
4	4		C++	5.574%	-2.58%
5	7	▲	C#	5.349%	+2.07%
6	5	▼	Visual Basic .NET	5.287%	-1.17%
7	6	▼	JavaScript	2.451%	-0.85%
8	8		PHP	2.405%	-0.28%
9	15	▲	Swift	1.795%	+0.61%
10	9	▼	SQL	1.504%	-0.77%
11	18	▲	Ruby	1.063%	-0.03%
12	17	▲	Delphi/Object Pascal	0.997%	-0.10%
13	10	▼	Objective-C	0.929%	-0.85%
14	16	▲	Go	0.900%	-0.22%
15	14	▼	Assembly language	0.877%	-0.32%
16	20	▲	Visual Basic	0.831%	-0.20%
17	25	▲	D	0.825%	+0.25%
18	12	▼	R	0.808%	-0.52%
19	13	▼	Perl	0.746%	-0.48%
20	11	▼	MATLAB	0.737%	-0.76%

По рейтингу TIOBE в январе 2020 года Java держал первые позиции. На Java можно создать и настольное приложение, и игру, и мобильное приложение.

## Языки программирования, которые стоит изучить в 2020 году:

- Kotlin
- HCL
- Go
- Javascript
- Typescript
- Python
- Java
- PHP

## Лучшие языки 2020 года:

- **Python** - язык программирования для искусственного интеллекта и машинного обучения
- **Java** - лучший язык программирования для серверной разработки и бэкенда
- **JavaScript** - популярный выбор для сценариев на стороне клиента
- **C++** - лучший язык программирования для общего назначения
- **C** - самый надежный язык на сегодняшний день
- **Ruby** - проверенный выбор для науки о данных и веб-разработки
- **C#** - мощный объектно-ориентированный язык от Microsoft
- **Swift** - самый эффективный язык программирования для разработки под iOS
- **PHP** - лучший серверный язык веб-разработки
- **Golang** - масштабируемый язык системного программирования от Google

**Среды программирования** — это набор инструментов, которые используются для преобразования символов в выполняемые вычисления.

## Компоненты среды:

- Редактор
- Компилятор
- Компоновщик или редактор связей
- Загрузчик
- Отладчик
- Средства тестирования
- Интерпретатор

## Лучшие IDE:

- Microsoft Visual Studio
- NetBeans
- PyCharm
- IntelliJ IDEA
- Eclipse
- Code::Blocks
- Aptana Studio 3
- Komodo

- RubyMine
- Xcode

### Требования для хорошей среды разработки:

- Сохранение файлов
- Запуск кода из среды
- Поддержка отладки
- Подсветка синтаксиса
- Автоматическое форматирование кода

### Тема 4: Язык Julia - язык научного программирования. (15.09.20)

**Julia** — высокоуровневый высокопроизводительный свободный язык программирования с динамической типизацией, созданный для математических вычислений. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков (например, MATLAB и Octave), однако имеет некоторые существенные отличия. Julia написан на Си, C++ и Scheme. В стандартный комплект входит JIT-компилятор на основе LLVM, благодаря чему, по утверждению авторов языка, приложения, полностью написанные на языке, практически не уступают в производительности приложениям, написанным на статически компилируемых языках вроде Си или C++. Большая часть стандартной библиотеки языка написана на нём же. Также язык имеет встроенную поддержку большого числа команд для распределенных вычислений.

### Возможности Julia:

- Мультиметод: обеспечивает возможность определять поведение функции в зависимости от типа передаваемых аргументов
- Динамическая типизация
- Хорошая производительность, сравнимая со статически типизированными языками как C
- Встроенная система управления пакетами
- Макросы и другие возможности метапрограммирования
- Вызов Python функций при помощи PyCall
- Вызов C функций напрямую: без дополнительных надстроек и API
- Богатые возможности для управления другими процессами
- Разрабатывался для параллельных и распределенных вычислений
- Сопрограммы: легковесные зеленые потоки (green threads)
- Возможность определять дополнительные типы не уступающие в скорости и удобстве встроенным
- Элегантные и расширяемые преобразования для числовых и других типов
- Поддержка Юникода, включающая, но не ограничиваемая только UTF-8

## Примеры кода:

```
function mandel(z)
    c = z
    maxiter = 80
    for n = 1:maxiter
        if abs(z) > 2
            return n-1
        end
        z = z^2 + c
    end
    return maxiter
end

function randmatstat(t)
    n = 5
    v = zeros(t)
    w = zeros(t)
    for i = 1:t
        a = randn(n,n)
        b = randn(n,n)
        c = randn(n,n)
        d = randn(n,n)
        P = [a b c d]
        Q = [a b; c d]
        v[i] = trace((P.'*P)^4)
        w[i] = trace((Q.'*Q)^4)
    end
    std(v)/mean(v), std(w)/mean(w)
end
```

```
nheads = @parallel (+) for i=1:100000000
    randbit()
end
```

## Инструменты:

**Julia Studio** — это интегрированная среда разработки для Julia. В настоящее время её заменила среда Juno, являющаяся надстройкой над Atom.