>_
sys.pro
mmf.nsu

Системными программистами не рождаются

Иван Углянский
Excelsior@Huawei

О чем доклад

- Об университетском IT образовании (классика?)

О чем доклад

- Об университетском IT образовании (классика?)
- Для системных программистов
- О том, как сделать его своими руками

О чем доклад

- Об университетском IT образовании (классика?)
- Для системных программистов
- О том, как сделать его своими руками
- Можно смотреть на это, как на рассказ о связи IT индустрии с системой образования (а можно и не так)

Чего мы хотим от мидла?

Говорю именно про свою команду

Чего мы хотим от мидла—?

Говорю именно про свою команду

Чего мы хотим от мидла—?

Говорю именно про свою команду

Команда мне отвечает:

- Чтобы умел делать то, о чем договорились

Чего мы хотим от мидла—?

Говорю именно про свою команду

Команда мне отвечает:

- Чтобы умел делать то, о чем договорились
- Чтобы спрашивал, что непонятно, обсуждал
- Чтобы умел читать спецификацию
- Чтобы вникал в постановку задачи

Чего мы хотим от мидла—?

Говорю именно про свою команду

Команда мне отвечает:

- Чтобы умел делать то, о чем договорились
- Чтобы спрашивал, что непонятно, обсуждал
- Чтобы умел читать спецификацию
- Чтобы вникал в постановку задачи

Остальному научим!

Чего мы хотим от мидла—?

Говорю именно про свою команду

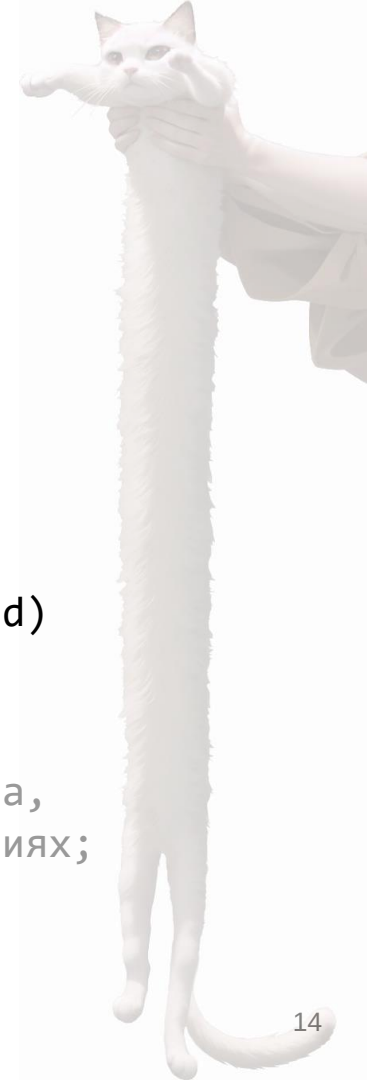
Ок, с софтами понятно. А все-таки, что нужно по хардам?

- Алгоритмы: классические структуры данных, оценки, трейдофы
- Парадигмы языков программирования: ООП vs ФП, их элементы
- Low-level: RISC-V and x86 assembly; ABI, Calling conventions; Processor pipeline, predictor;
- OS: устройство виртуальной памяти, планировщик, caches, MESI;
- Многопоточка: parallel/concurrent, data race, mutex, condvar, true/false sharing, aba, поддержка и реализация в языках;
- Реализация языков программирования (AOT/JIT/Interpreter/Tiered)
- ООП: и концепции и реализация - VMT/IMT, Окно Коэна, PIC;
- GC stuff: manual MM, алгоритмы выделения памяти, сборка мусора, refs counting vs tracing GC, алгоритмы GC, гипотезы о поколениях;
- Compiler stuff: IR, SSA, Sea of Nodes, GCM, GVN, regalloc.
- Tools, build systems, performance audit, etc.

- **Алгоритмы**: классические структуры данных, оценки, трейдофы
- **Парадигмы** языков программирования: ООП vs ФП, их элементы
- Low-level: RISC-V and x86 assembly; ABI, Calling conventions; Processor pipeline, predictor;
- OS: устройство виртуальной памяти, планировщик, caches, MESI;
- Многопоточка: parallel/concurrent, data race, mutex, condvar, true/false sharing, aba, поддержка и реализация в языках;
- Реализация языков программирования (AOT/JIT/Interpreter/Tiered)
- ООП: и концепции и реализация - VMT/IMT, Окно Коэна, PIC;
- GC stuff: manual MM, алгоритмы выделения памяти, сборка мусора, refs counting vs tracing GC, алгоритмы GC, гипотезы о поколениях;
- Compiler stuff: IR, SSA, Sea of Nodes, GCM, GVN, regalloc;
- Tools, build systems, performance audit, etc.

- **Алгоритмы**: классические структуры данных, оценки, трейдофы
- **Парадигмы** языков программирования: ООП vs ФП, их элементы
- **Low-level**: RISC-V and x86 assembly; ABI, Calling conventions; Processor pipeline, predictor;
- **OS**: устройство виртуальной памяти, планировщик, caches, MESI;
- **Многопоточка**: parallel/concurrent, data race, mutex, condvar, true/false sharing, aba, поддержка и реализация в языках;
- Реализация языков программирования (AOT/JIT/Interpreter/Tiered)
- ООП: и концепции и реализация - VMT/IMT, Окно Коэна, PIC;
- GC stuff: manual MM, алгоритмы выделения памяти, сборка мусора, refs counting vs tracing GC, алгоритмы GC, гипотезы о поколениях;
- Compiler stuff: IR, SSA, Sea of Nodes, GCM, GVN, regalloc;
- Tools, build systems, performance audit, etc.

- **Алгоритмы**: классические структуры данных, оценки, трейдофы
- **Парадигмы** языков программирования: ООП vs ФП, их элементы
- **Low-level**: RISC-V and x86 assembly; ABI, Calling conventions; Processor pipeline, predictor;
- **OS**: устройство виртуальной памяти, планировщик, caches, MESI;
- **Многопоточка**: parallel/concurrent, data race, mutex, condvar, true/false sharing, aba, поддержка и реализация в языках;
- **Реализация** языков программирования (AOT/JIT/Interpreter/Tiered)
- ООП: и концепции и **реализация** - VMT/IMT, Окно Коэна, PIC;
- GC stuff: manual MM, алгоритмы выделения памяти, сборка мусора, refs counting vs tracing GC, алгоритмы GC, гипотезы о поколениях;
- Compiler stuff: IR, SSA, Sea of Nodes, GCM, GVN, regalloc;
- Tools, build systems, performance audit, etc.



- **Алгоритмы**: классические структуры данных, оценки, трейдофы
- **Парадигмы** языков программирования: ООП vs ФП, их элементы
- **Low-level**: RISC-V and x86 assembly; ABI, Calling conventions; Processor pipeline, predictor;
- **OS**: устройство виртуальной памяти, планировщик, caches, MESI;
- **Многопоточка**: parallel/concurrent, data race, mutex, condvar, true/false sharing, aba, поддержка и реализация в языках;
- **Реализация** языков программирования (AOT/JIT/Interpreter/Tiered)
- ООП: и концепции и **реализация** - VMT/IMT, Окно Коэна, PIC;
- **GC stuff**: manual MM, алгоритмы выделения памяти, сборка мусора, refs counting vs tracing GC, алгоритмы GC, гипотезы о поколениях;
- **Compiler stuff**: IR, SSA, Sea of Nodes, GCM, GVN, regalloc;
- Tools, build systems, performance audit, etc.

- **Алгоритмы**: классические структуры данных, оценки, трейдофы
- **Парадигмы** языков программирования: ООП vs ФП, их элементы
- **Low-level**: RISC-V and x86 assembly; ABI, Calling conventions; Processor pipeline, predictor;
- **OS**: устройство виртуальной памяти, планировщик, caches, MESI;
- **Многопоточка**: parallel/concurrent, data race, mutex, condvar, true/false sharing, aba, поддержка и реализация в языках;
- **Реализация** языков программирования (AOT/JIT/Interpreter/Tiered)
- ООП: и концепции и **реализация** - VMT/IMT, Окно Коэна, PIC;
- **GC stuff**: manual MM, алгоритмы выделения памяти, сборка мусора, refs counting vs tracing GC, алгоритмы GC, гипотезы о поколениях;
- **Compiler stuff**: IR, SSA, Sea of Nodes, GCM, GVN, regalloc;
- Tools, build systems, performance audit, etc.

Пара вопросов

1. Вы нормальные вообще?

Пара вопросов

1. Вы нормальные вообще?
2. Такие джуны/мидлы вообще бывают?
3. А вам зачем???

Иван Углянский

JVM engineer at Excelsior

VM engineer at Excelsior@Huawei

@dbg_nsk

@ugliansky

Иван Углянский

Делали JVM с AOT компилятором

Полностью самобытная кодовая база, не базировалась на OpenJDK

JVM engineer at Excelsior

VM engineer at Excelsior@Huawei

@dbg_nsk

@ugliansky

Иван Углянский

Делали JVM с AOT компилятором

Полностью самобытная кодовая база, не базировалась на OpenJDK

JVM engineer at Excelsior

VM engineer at Excelsior@Huawei

Делаем компиляторы, VM и новые языки программирования

@dbg_nsk

@ugliansky

Пара вопросов

1. Вы нормальные вообще?
2. Такие джуны/мидлы вообще бывают?
3. А вам зачем??? → Чтобы делать компиляторы, рантаймы, виртуальные машины и новые языки программирования

Пара вопросов

Те, кто этим занимаются, – это подвид "системных программистов"

1. Вы нормальные вообще?
2. Такие джуны/мидлы вообще бывают?
3. А вам зачем??? → Чтобы делать компиляторы, рантаймы, виртуальные машины и новые языки программирования

https://www.youtube.com/watch?v=ywUvhPsd6Ew

Пара вопросов

Те, кто этим занимаются, – это подвид "системных программистов"

✓ 1. Вы нормальные вообще?

? 2. Такие джуны/мидлы вообще бывают?

✓ 3. А вам зачем??? → Чтобы делать компиляторы, рантаймы, виртуальные машины и новые языки программирования

Нужен волшебник!

Нужен волшебник!

OOP:

- Conceptions
- VMT/IMT
- Cohen Display
- PIC

Low level:

- ABI
- Intel/RISC-V
- OS stuff
- Linker

Paradigms

Algorithms

GC stuff: reference counting vs tracing GC, concurrent, generational, region based

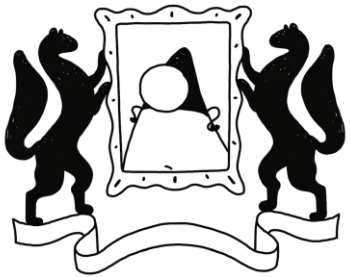
Compilers stuff: IR, SSA, Sea of Nodes, GCM, GVN

Проблемы

1. Людей с такой квалификацией вообще мало

Проблемы

1. Людей с такой квалификацией вообще мало
2. Ищем такого человека в Новосибирске.



Проблемы

1. Людей с такой квалификацией вообще мало
2. Ищем такого человека в Новосибирске. А здесь много команд системных программистов => большая конкуренция за них!

Проблемы

1. Людей с такой квалификацией вообще мало
2. Ищем такого человека в Новосибирске. А здесь много команд системных программистов => большая конкуренция за них!

Решение

- Нужно самим учить людей и давать такую квалификацию!

Проблемы

1. Людей с такой квалификацией вообще мало
2. Ищем такого человека в Новосибирске. А здесь много команд системных программистов => большая конкуренция за них!

Решение

- Нужно самим учить людей и давать такую квалификацию!
- Студенты кажутся для этого идеальными кандидатами.

Проблемы

1. Людей с такой квалификацией вообще мало
1. Ищем такого человека в Новосибирске. А здесь много команд системных программистов => большая конкуренция за них!

Решение

- Нужно самим учить людей и давать такую квалификацию!
- Студенты кажутся для этого идеальными кандидатами.
- Весь Excelsior состоит из бывших студентов (и их преподавателей)

Как мы делали стажировки

1. Талантливые студенты (2-3 курс) подаются на стажировку

Как мы делали стажировки

1. Талантливые студенты (2-3 курс) подаются на стажировку

2. Первый этап: разбор научных статей по теме + мелкие практические задачи

Как мы делали стажировки

1. Талантливые студенты (2-3 курс) подаются на стажировку

2. Первый этап: разбор научных статей по теме + мелкие практические задачи

3. К 4-ому курсу начинается диплом.

Научрук: практикующий VM инженер
Задача: R&D в рамках настоящей VM
Результат: (зачастую) часть продукта

Как мы делали стажировки

1. Талантливые студенты (2-3 курс)
   подаются на стажировку

2. Первый этап: разбор научных статей по теме +
   мелкие практические задачи

3. К 4-ому курсу начинается диплом.

   Научрук: практикующий VM инженер
   Задача: R&D в рамках настоящей VM
   Результат: (зачастую) часть продукта

4. Продолжение в магистратуре (уже не стажировка)

Как мы делали стажировки

1. Талантливые студенты (2-3 курс) подаются на стажировку
2. Первый этап: разбор научных статей по теме + мелкие практические задачи

0.5-2 года 😲

3. К 4-ому курсу начинается диплом.

   Научрук: практикующий VM инженер
   Задача: R&D в рамках настоящей VM
   Результат: (зачастую) часть продукта

1 год 😲

4. Продолжение в магистратуре (уже не стажировка)

Как мы делали стажировки: проблемы

1. Поиск талантов (где найти замотивированных?)
2. Путь от студента до инженера занимает **годы.**

Как мы делали стажировки: проблемы

1. Поиск талантов (где найти замотивированных?)

   Решение:
   - преподавать в университете (качественно)
   - выделять талантливых студентов
   - рассказывать им про компиляторы 
   - PROFIT!

2. Путь от студента до инженера занимает **годы.**

Как мы делали стажировки: проблемы

1. Поиск талантов (где найти замотивированных?)

   Решение:
   - преподавать в университете (качественно)
   - выделять талантливых студентов
   - рассказывать им про компиляторы 
   - PROFIT!

2. Путь от студента до инженера занимает **годы.**

   Решение:
   - А что, если часть своего опыта передавать уже сразу в университете?

Как мы делали стажировки: проблемы

Решение:

- А что, если часть своего опыта передавать уже сразу в университете?

Как мы делали стажировки: проблемы

Решение:

- А что, если часть своего опыта передавать уже сразу в университете?
- Тогда люди будут приходить на стажировку уже подготовленными

Как мы делали стажировки: проблемы

Решение:

- А что, если часть своего опыта передавать уже сразу в университете?
- Тогда люди будут приходить на стажировку уже подготовленными
- Моральное удовлетворение: IT образование станет лучше!

Нужен волшебник?

Нужен волшебник => построй Хогвартс

Как построить Хогвартс?

Как построить Хогвартс?

- Нужен замок

Как построить Хогвартс?

- Нужен замок => Новосибирский Государственный Университет

Как построить Хогвартс?

- Нужен замок ✓

Как построить Хогвартс?

- Нужен замок ✓
- Команда волшебников-преподавателей,

Преподаватели

- 6 сотрудников Excelsior, читающих курсы в НГУ

Преподаватели

- 6 сотрудников Excelsior, читающих курсы в НГУ

Преподаватели

- 6 сотрудников Excelsior, читающих курсы в НГУ

  Исторически были на ММФ
  (Механико-математический факультет, не IT факультет)

Преподаватели

- 6 сотрудников Excelsior, читающих курсы в НГУ

  Исторически были на ММФ
  Считаем это правильным местом (матан важен)

Преподаватели

- 6 сотрудников Excelsior, читающих курсы в НГУ
- Курсы:

  Базовый курс по C — один семестр
  Курс по ООП на C++ — один семестр

Преподаватели

- 6 сотрудников Excelsior, читающих курсы в НГУ
- Курсы:

  Базовый курс по C — один семестр
  Курс по ООП на C++ — один семестр

  Теория программирования — один семестр → читается на 4 курсе

  Спецкурс по компиляторам — два семестра → спецкурс для уже заинтересованных

Преподаватели

- 6 сотрудников Excelsior, читающих курсы в НГУ
- Курсы:

Базовый курс по C — один семестр
Курс по ООП на C++ — один семестр

} т.е. 2 семестра на все.

Теория программирования — один семестр → читается на 4 курсе

Спецкурс по компиляторам — два семестра → спецкурс для уже заинтересованных

Преподаватели

- 6 сотрудников Excelsior, читающих курсы в НГУ

Преподаватели

Нужно больше разнообразных курсов по нашей теме!

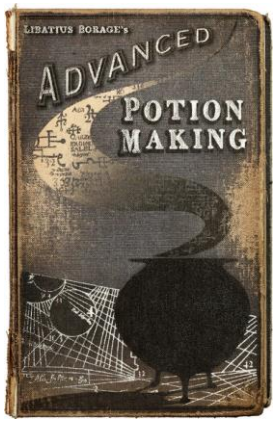
- 6 сотрудников Excelsior, читающих курсы в НГУ

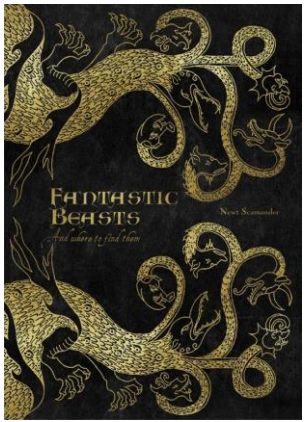
Как построить Хогвартс?

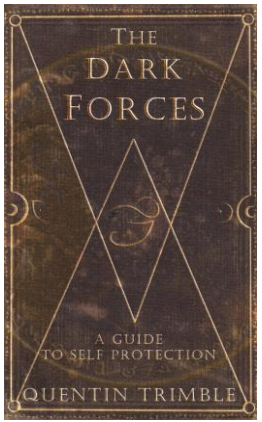
- Нужен замок ✓
- Команда волшебников-преподавателей, ✓

Как построить Хогвартс?

- Нужен замок ✓
- Команда волшебников-преподавателей, ✓
- Магическая учебная программа,







Составляем программу

– Простите, а можно нам не 2 семестра, а 3? Совершенно не помещаемся, нужно больше времени!

Составляем программу

- Простите, а можно нам не 2 семестра, а 3? Совершенно не помещаемся, нужно больше времени!

- Нет, так не пойдет. Куда же нам его вставить?

Составляем программу

– Простите, а можно нам не 2 семестра, а 3? Совершенно не помещаемся, нужно больше времени!

– Нет, так не пойдет. Куда же нам его вставить?

…

– А давайте мы тогда сделаем свой профиль, где будет не 2 наших курса, а 16?

Составляем программу

– Простите, а можно нам не 2 семестра, а 3? Совершенно не помещаемся, нужно больше времени!

– Нет, так не пойдет. Куда же нам его вставить?

…

– А давайте мы тогда сделаем свой профиль, где будет не 2 наших курса, а 16?

– А давайте ☺

Делаем свой профиль

Что вообще такое профиль?

Делаем свой профиль

Что вообще такое профиль?

- Механизм ранней специализации в рамках факультета. Обычно специализация на 3 курсе, а на профиле - сразу при поступлении.

Делаем свой профиль

Что вообще такое профиль?

- Механизм ранней специализации в рамках факультета. Обычно специализация на 3 курсе, а на профиле - сразу при поступлении.
- Изменение учебного плана с первого дня.
- Дополнительные испытания на входе.

Делаем свой профиль

Что вообще такое профиль?

- Механизм ранней специализации в рамках факультета. Обычно специализация на 3 курсе, а на профиле - сразу при поступлении.
- Изменение учебного плана с первого дня.
- Дополнительные испытания на входе.
- На ММФ НГУ есть целый набор профилей:

Делаем свой профиль

В нашем случае:

- Договорились набирать на 1 курс ММФ
  одну специальная группа (16 человек) каждый год

Делаем свой профиль

В нашем случае:

- Договорились набирать на 1 курс ММФ
  одну специальная группа (16 человек) каждый год

- Меняем программу с первого же семестра

Делаем свой профиль

В нашем случае:

- Договорились набирать на 1 курс ММФ
  одну специальная группа (16 человек) каждый год

- Меняем программу с первого же семестра

  ✓ Остается базовая математика: матан, вышка, ангеом, дискретка, диффуры, теорвер и т.д.

Делаем свой профиль

В нашем случае:

- Договорились набирать на 1 курс ММФ
  одну специальная группа (16 человек) каждый год

- Меняем программу с первого же семестра

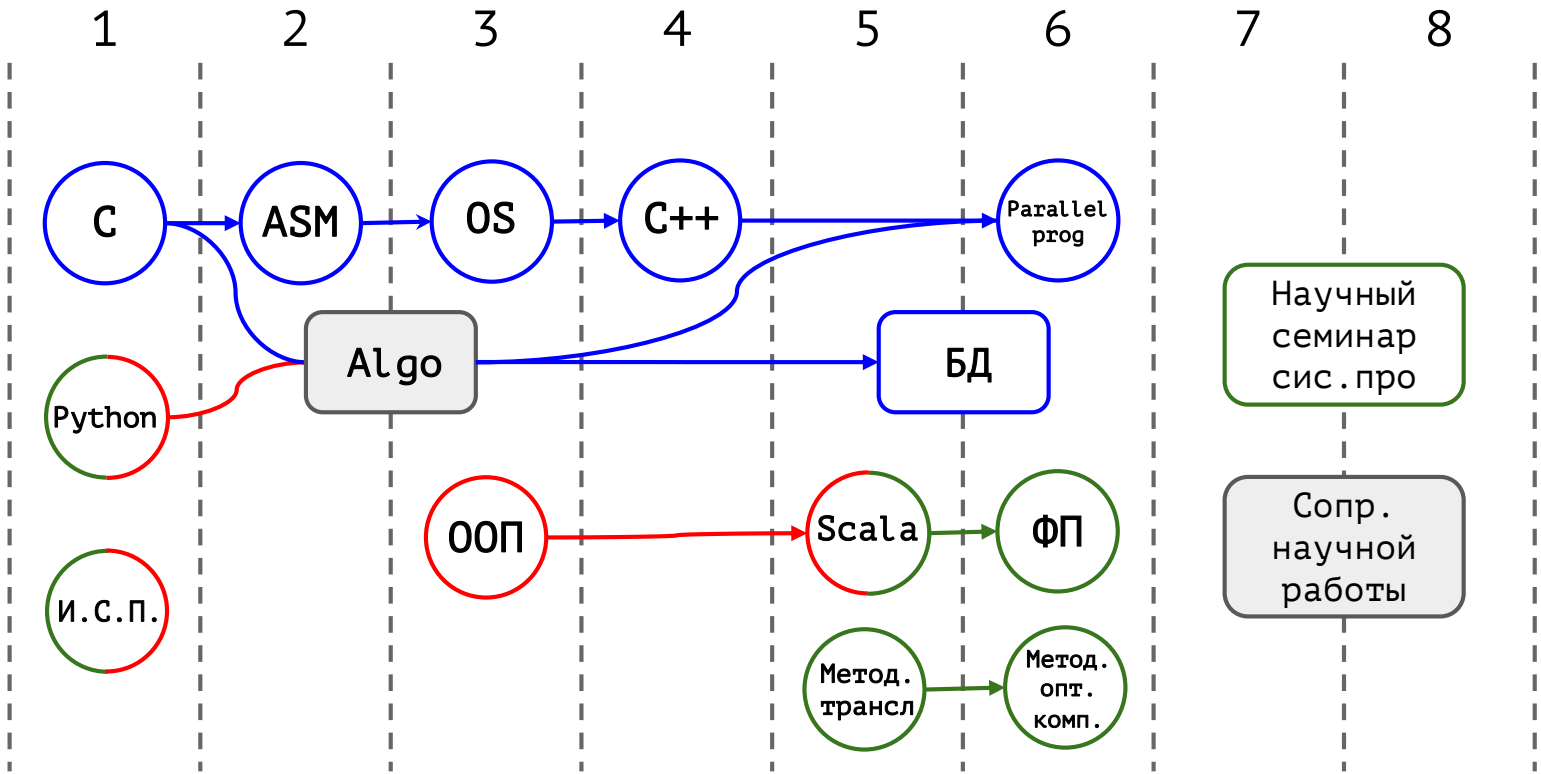
  ✓ Остается базовая математика: матан, вышка, ангеом, дискретка, диффуры, теорвер и т.д.

  ✓ Убираем: выч. мат. и физику

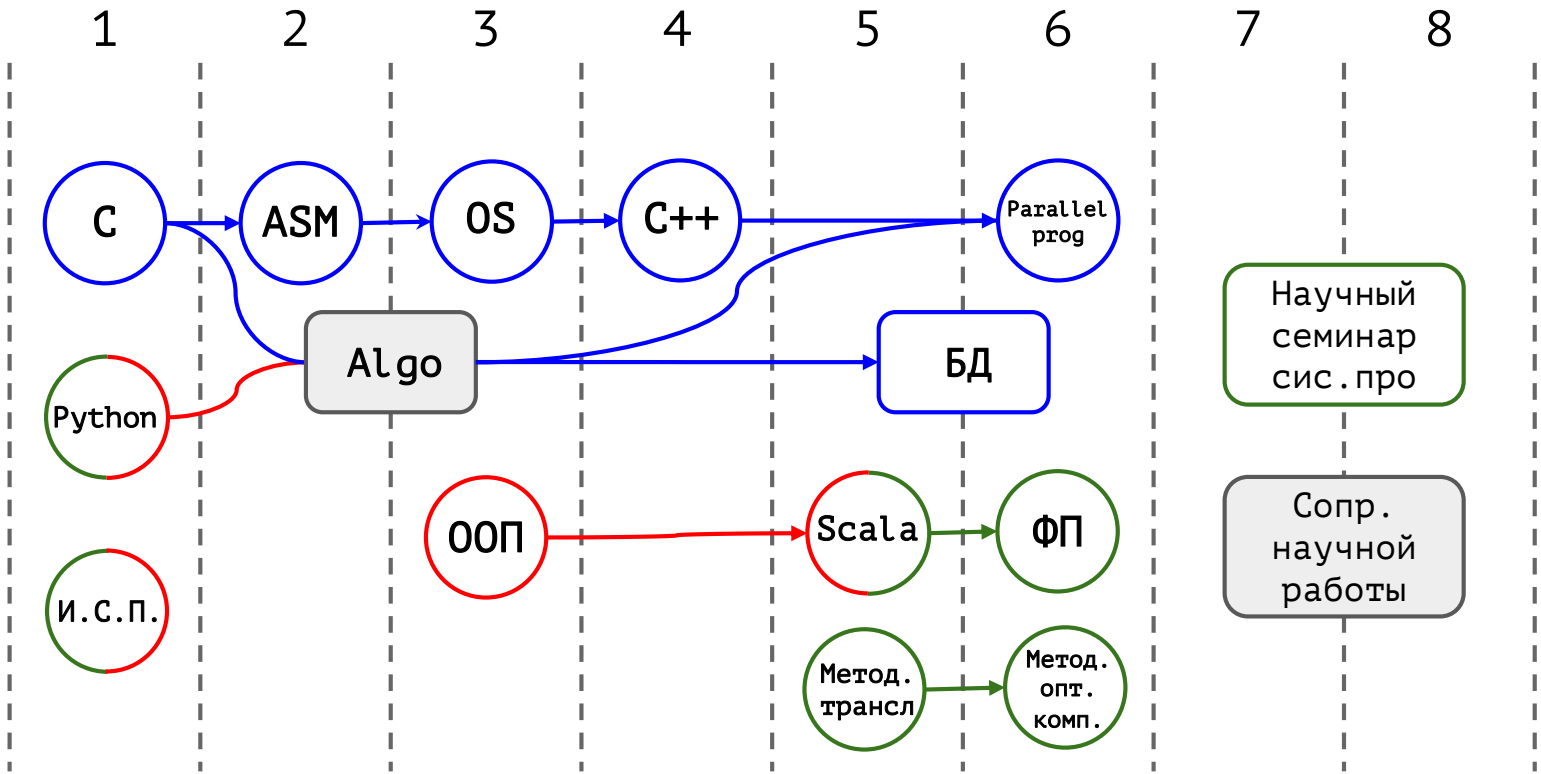
  ✓ Добавляем свои курсы

Так и чему учить то будем?

Составляем программу

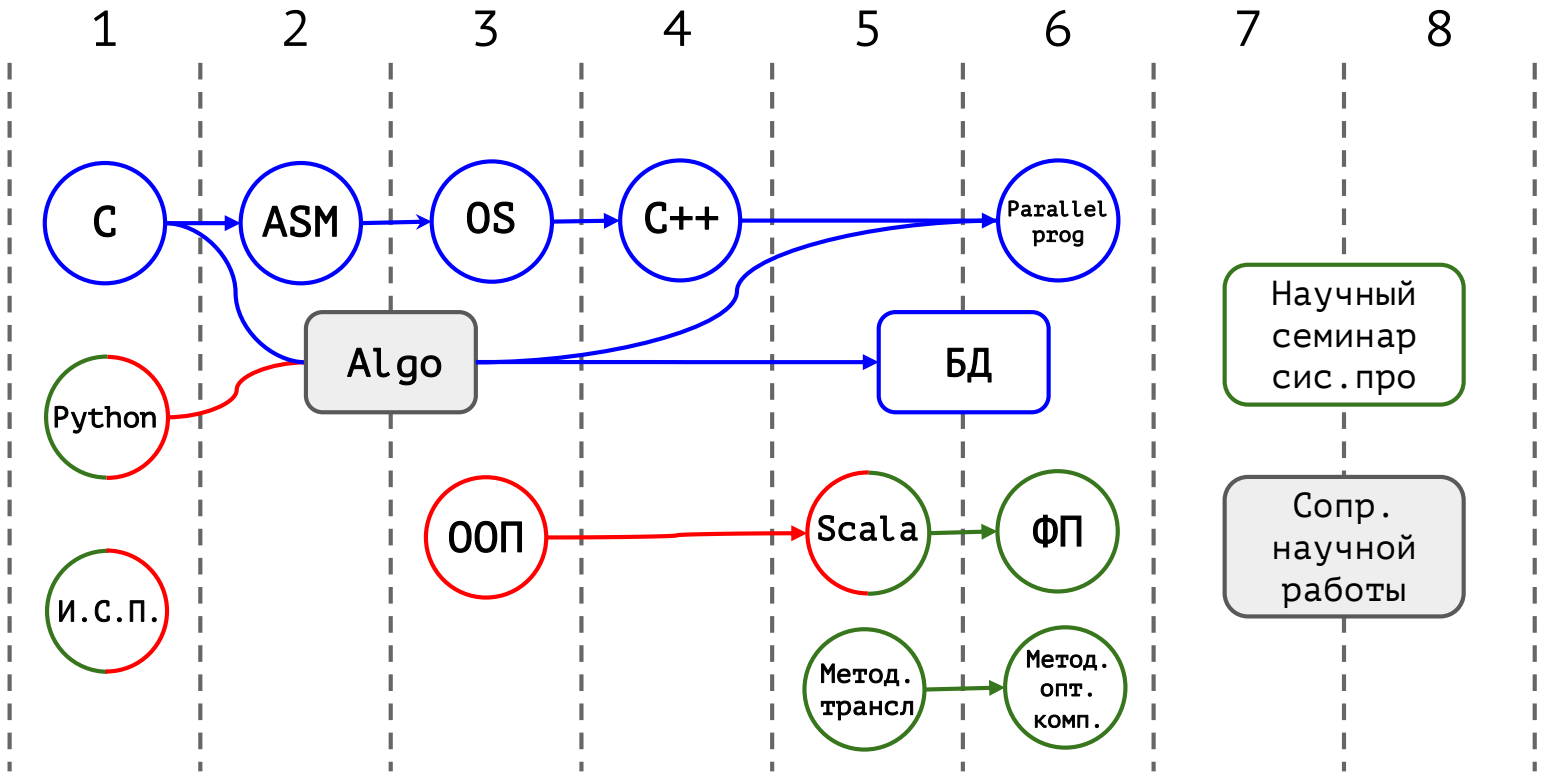


- — "системщина",
  низкий уровень



■ — "системщина", низкий уровень

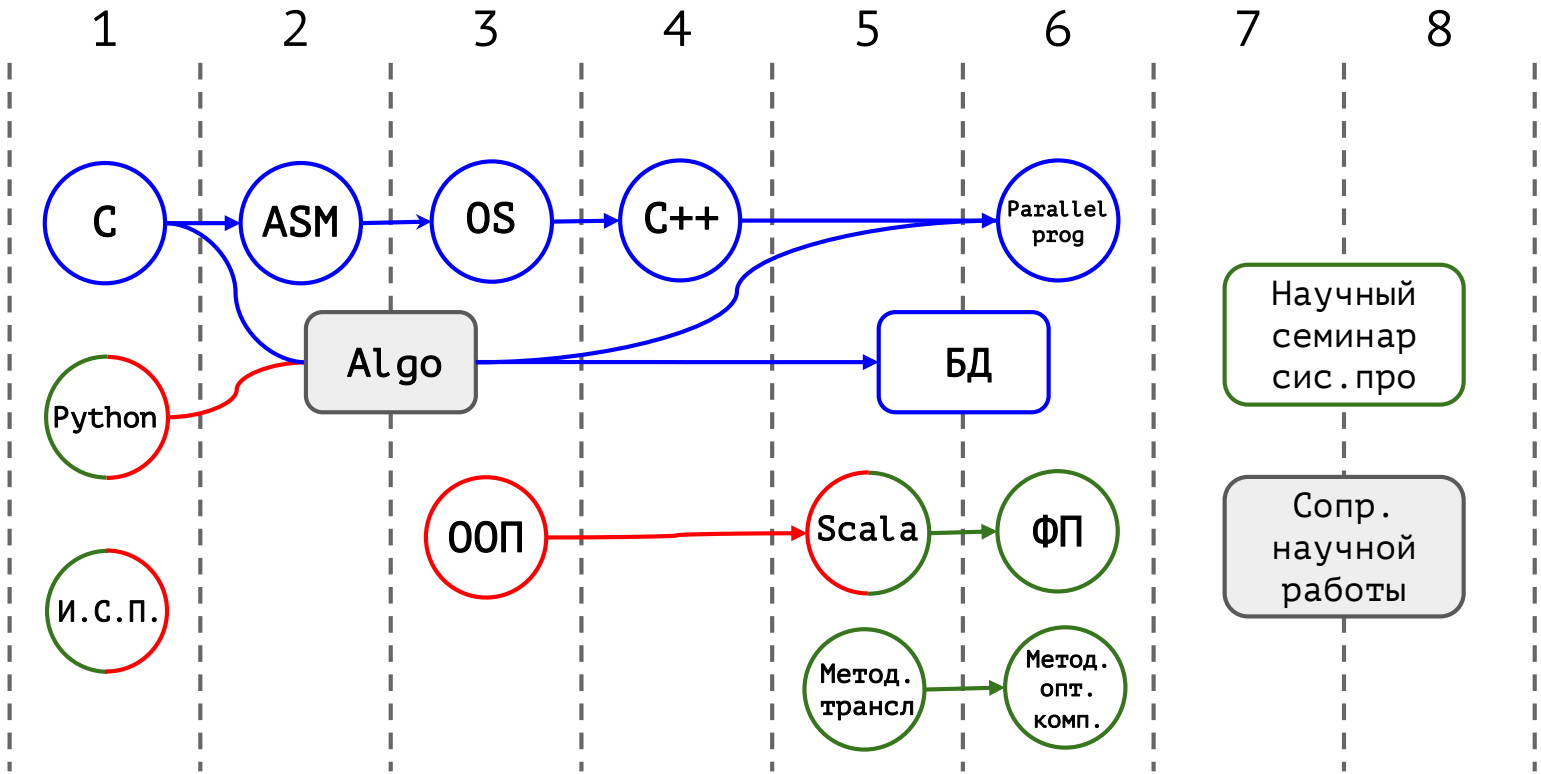
■ — "программистский кругозор", какие языки, как реализуют



■ — "системщина", низкий уровень

■ — "программистский кругозор", какие языки, как реализуют

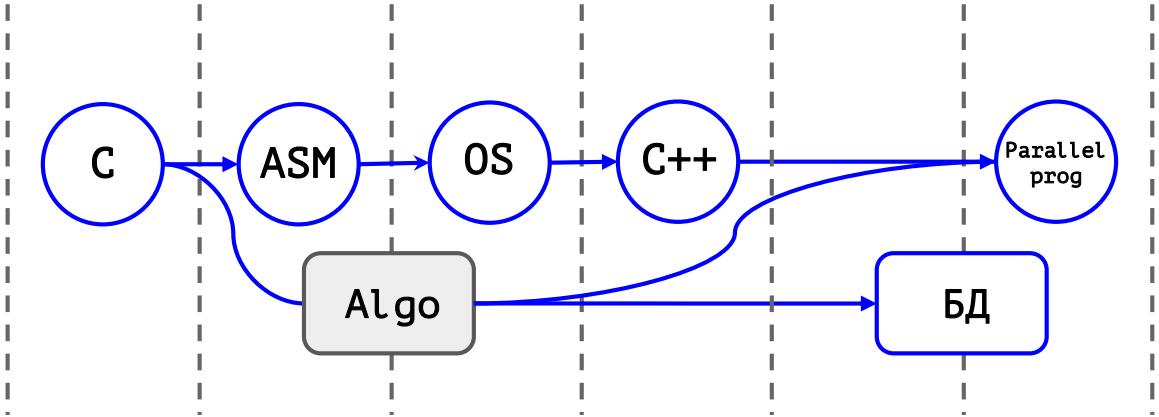
■ — "прикладное" программирование



- **Алгоритмы**: классические структуры данных, оценки, трейдофы
- **Парадигмы** языков программирования: ООП vs ФП, их элементы
- **Low-level**: RISC-V and x86 assembly; ABI, Calling conventions; Processor pipeline, predictor;
- **OS**: устройство виртуальной памяти, планировщик, caches, MESI;
- **Многопоточка**: parallel/concurrent, data race, mutex, condvar, true/false sharing, aba, поддержка и реализация в языках;
- **Реализация** языков программирования (AOT/JIT/Interpreter/Tiered)
- ООП: и концепции и **реализация** - VMT/IMT, Окно Коэна, PIC;
- **GC stuff**: manual MM, алгоритмы выделения памяти, сборка мусора, refs counting vs tracing GC, алгоритмы GC, гипотезы о поколениях;
- **Compiler stuff**: IR, SSA, Sea of Nodes, GCM, GVN, regalloc;
- Tools, build systems, performance audit, etc.

Составляем программу

- С низкоуровневыми предметами - понятно, фокус на реализации и на связи с машиной



Составляем программу

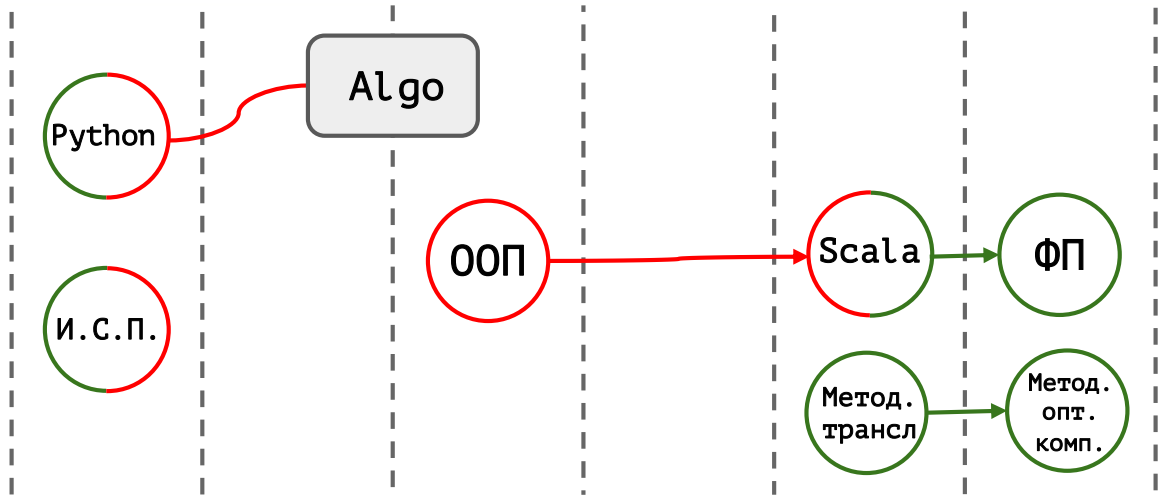
- С низкоуровневыми предметами - понятно, фокус на реализации и на связи с машиной

  Отличительные особенности:

  ✓ На С говорим про ABI
  ✓ Ассемблер и Intel x86 и RISC-V
  ✓ OS - пишем свою с нуля
  ✓ C++ - меньше фокус на ООП, больше на реализацию

Составляем программу

- С низкоуровневыми предметами - понятно, фокус на реализации и на связи с машиной
- Зачем нужны красные (прикладные) курсы, и что на них?



Составляем программу

Пример: все абитуриенты знают Python (спасибо, ЕГЭ).

Составляем программу

Пример: все абитуриенты знают Python (спасибо, ЕГЭ).

Что напечатает?

```
x = 10

def foo():
    print(x)

foo()
```

Составляем программу

Пример: все абитуриенты знают Python (спасибо, ЕГЭ).

Что напечатает?

```
x = 10

def foo():
    print(x)

foo() // 10
```

Составляем программу

Пример: все абитуриенты знают Python (спасибо, ЕГЭ).

Что напечатает?

```
x = 10

def foo():
    print(x)
    x = 30

foo()
foo()
```

Составляем программу

Пример: все абитуриенты знают Python (спасибо, ЕГЭ).

Что напечатает?

```
x = 10

def foo():
    print(x)
    x = 30

foo() // 10 ?
foo() // 30 ?
```

Составляем программу

Пример: все абитуриенты знают Python (спасибо, ЕГЭ).

Что напечатает?

```
x = 10

def foo():
    print(x)
    x = 30

foo()
foo()
```

```
UnboundLocalError: local variable 'x' referenced before assignment
```

Составляем программу

Пример: все абитуриенты знают Python (спасибо, ЕГЭ).

Что напечатает? => Почему?

```
x = 10

def foo():
    print(x)
    x = 30

foo()
foo()
```

```
UnboundLocalError: local variable 'x' referenced before assignment
```

Составляем программу

Пример: все абитуриенты знают Python (спасибо, ЕГЭ).

Что напечатает? => Почему? => Надо написать global

```
x = 10

def foo():
    global x
    print(x)
    x = 30

foo()
foo()
```

Составляем программу

Пример: все абитуриенты знают Python (спасибо, ЕГЭ).

Что напечатает? => Почему? => Надо написать global
=> Но почему?

```
x = 10

def foo():
    global x
    print(x)
    x = 30

foo() // 10
foo() // 30
```

Составляем программу

Пример: все абитуриенты знают Python (спасибо, ЕГЭ).

Что напечатает? => Почему? => Надо написать global
=> Но почему?
=> Потому, что Python так работает
=> Но почему?

```
x = 10

def foo():
    global x
    print(x)
    x = 30

foo() // 10
foo() // 30
```

Составляем программу

Пример: все абитуриенты знают Python (спасибо, ЕГЭ).

```
x = 10

def foo():
    print(x)
    x = 30

foo()
```

```
8    0  LOAD_GLOBAL       0 (print)
     2  LOAD_FAST_CHECK   0 (x)
     4  CALL_FUNCTION     1
     6  POP_TOP

9    8  LOAD_CONST        1 (30)
    10  STORE_FAST        0 (x)
    12  LOAD_CONST        0 (None)
    14  RETURN_VALUE
```

Составляем программу

Пример: все абитуриенты знают Python (спасибо, ЕГЭ).

```
x = 10

def foo():
    print(x)
    x = 30

foo()
```

```
8     0  LOAD_GLOBAL       0 (print)
      2  LOAD_FAST_CHECK   0 (x)
      4  CALL_FUNCTION     1
      6  POP_TOP

9     8  LOAD_CONST        1 (30)
     10  STORE_FAST        0 (x)
     12  LOAD_CONST        0 (None)
     14  RETURN_VALUE
```

При трансляции в байткод статически вычисляются все локалы, и работа с ними идет через *_fast_* инструкции.

Составляем программу

Пример: все абитуриенты знают Python (спасибо, ЕГЭ).

```
x = 10

def foo():
    print(x)
    x = 30

foo()
```

```
8     0  LOAD_GLOBAL       0 (print)
      2  LOAD_FAST_CHECK   0 (x)
      4  CALL_FUNCTION     1
      6  POP_TOP

9     8  LOAD_CONST        1 (30)
     10  STORE_FAST        0 (x)
     12  LOAD_CONST        0 (None)
     14  RETURN_VALUE
```

При трансляции в байткод статически вычисляются все локалы, и работа с ними идет через *_fast_* инструкции.

Это позволяет ускорить доступ к локалам (использовать массив)

Составляем программу

Пример: все абитуриенты знают Python (спасибо, ЕГЭ).

```
x = 10

def foo():
    print(x)
    x = 30

foo()
```

```
8     0  LOAD_GLOBAL       0 (print)
      2  LOAD_FAST_CHECK   0 (x)
      4  CALL_FUNCTION     1
      6  POP_TOP

9     8  LOAD_CONST        1 (30)
     10  STORE_FAST        0 (x)
     12  LOAD_CONST        0 (None)
     14  RETURN_VALUE
```

При трансляции в байткод статически вычисляются все локалы, и работа с ними идет через *_fast_* инструкции.

Это позволяет ускорить доступ к локалам (использовать массив)

А еще эффективно реализовать замыкание!

Составляем программу

Пример #2: ООП читается на Java

Пример #2: ООП читается на Java. 8 строчек кода:

```
public static void main(String[] args) {
    final int size = 512*1024*1024 / Integer.BYTES;
    int[] arr1 = new int[size];
    System.out.println("arr1 (" + arr1 + ") allocated");
    System.gc();
    int[] arr2 = new int[size];
    System.out.println("arr2 (" + arr2 + ") allocated");
}
```

512mb

Пример #2: ООП читается на Java. 8 строчек кода:

```
public static void main(String[] args) {
    final int size = 512*1024*1024 / Integer.BYTES;
    int[] arr1 = new int[size];                                  512mb
    System.out.println("arr1 (" + arr1 + ") allocated");
    System.gc();
    int[] arr2 = new int[size];                                  512mb
    System.out.println("arr2 (" + arr2 + ") allocated");
}
```

Пример #2: ООП читается на Java. 8 строчек кода:

```
public static void main(String[] args) {
    final int size = 512*1024*1024 / Integer.BYTES;
    int[] arr1 = new int[size];                                  512mb
    System.out.println("arr1 (" + arr1 + ") allocated");
    System.gc();
    int[] arr2 = new int[size];                                  512mb
    System.out.println("arr2 (" + arr2 + ") allocated");
}
```

java -Xmx768m Test →

```
arr1 ([I@4aa8f0b4) allocated
Exception in thread "main"
java.lang.OutOfMemoryError: ...
```

Пример #2: ООП читается на Java. 8 строчек кода:

```
public static void main(String[] args) {
    final int size = 512*1024*1024 / Integer.BYTES;
    int[] arr1 = new int[size];                                  512mb
    System.out.println("arr1 (" + arr1 + ") allocated");
    System.gc();
    int[] arr2 = new int[size];                                  512mb
    System.out.println("arr2 (" + arr2 + ") allocated");
}
```

```
java -Xcomp -Xmx768m Test
```

Пример #2: ООП читается на Java. 8 строчек кода:

```
public static void main(String[] args) {
    final int size = 512*1024*1024 / Integer.BYTES;
    int[] arr1 = new int[size];                                  512mb
    System.out.println("arr1 (" + arr1 + ") allocated");
    System.gc();
    int[] arr2 = new int[size];                                  512mb
    System.out.println("arr2 (" + arr2 + ") allocated");
}
```

java -Xcomp -Xmx768m Test →

```
arr1 ([I@4aa8f0b4) allocated
arr2 ([I@9e89d68) allocated
```

Пример #2: ООП читается на Java. 8 строчек кода:

```
public static void main(String[] args) {
    final int size = 512*1024*1024 / Integer.BYTES;
    int[] arr1 = new int[size];
    System.out.println("arr1 (" + arr1 + ") allocated");
    System.gc();
    int[] arr2 = new int[size];
    System.out.println("arr2 (" + arr2 + ") allocated");
}
```

Почему отличается для интерпретации и компиляции?

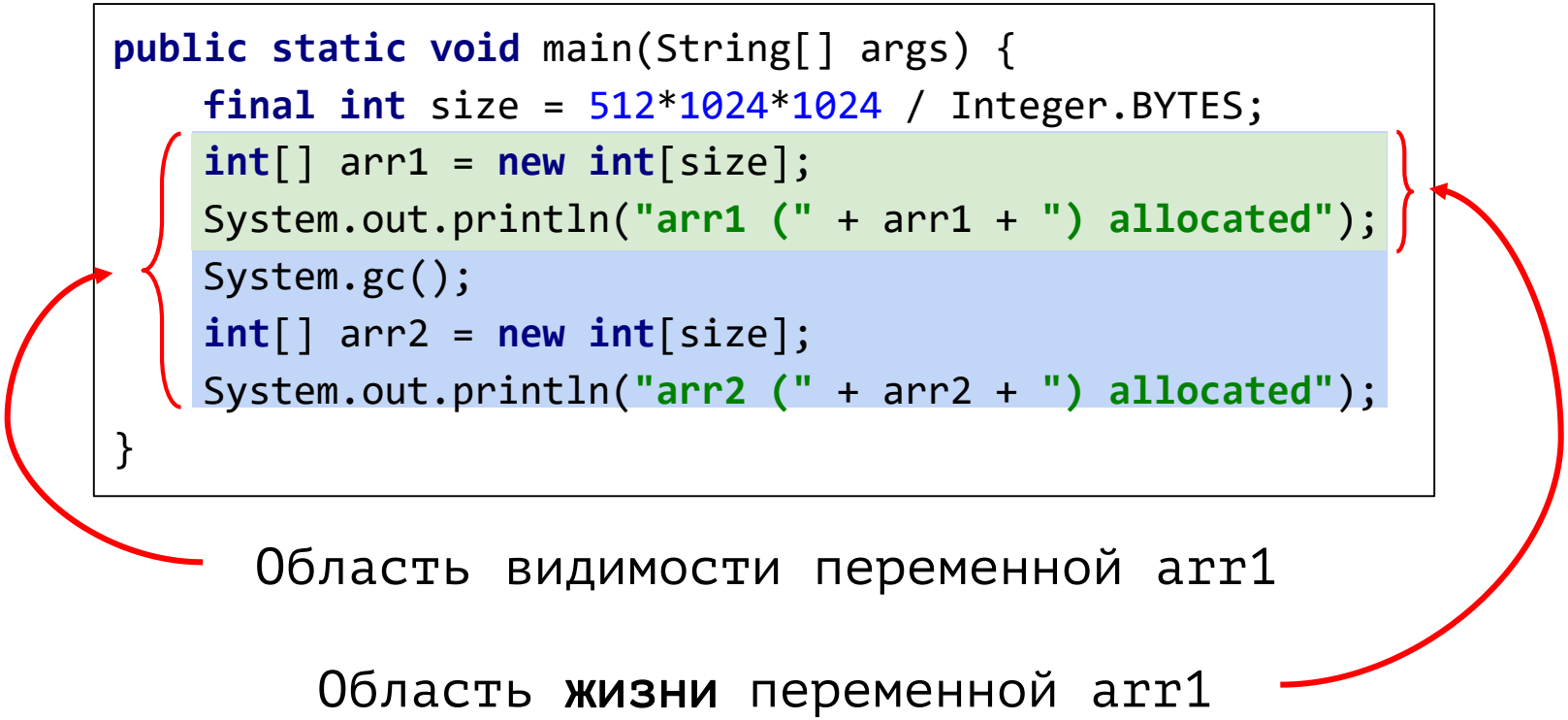
Как это связано со сборкой мусора?

Пример #2: ООП читается на Java. 8 строчек кода:

```
public static void main(String[] args) {
    final int size = 512*1024*1024 / Integer.BYTES;
    int[] arr1 = new int[size];
    System.out.println("arr1 (" + arr1 + ") allocated");
    System.gc();
    int[] arr2 = new int[size];
    System.out.println("arr2 (" + arr2 + ") allocated");
}
```

Область видимости переменной arr1

Пример #2: ООП читается на Java. 8 строчек кода:



```
public static void main(String[] args) {
    final int size = 512*1024*1024 / Integer.BYTES;
    int[] arr1 = new int[size];
    System.out.println("arr1 (" + arr1 + ") allocated");
    System.gc();
    int[] arr2 = new int[size];
    System.out.println("arr2 (" + arr2 + ") allocated");
}
```

Пример #2: ООП читается на Java. 8 строчек кода:

```
public static void main(String[] args) {
    final int size = 512*1024*1024 / Integer.BYTES;
    int[] arr1 = new int[size];
    System.out.println("arr1 (" + arr1 + ") allocated");
    System.gc();
    int[] arr2 = new int[size];
    System.out.println("arr2 (" + arr2 + ") allocated");
}
```

Здесь GC имеет право
собрать первый массив

Пример #2: ООП читается на Java. 8 строчек кода:

```
public static void main(String[] args) {
    final int size = 512*1024*1024 / Integer.BYTES;
    int[] arr1 = new int[size];                                  512mb
    System.out.println("arr1 (" + arr1 + ") allocated");
    System.gc();
    int[] arr2 = new int[size];                                  512mb
    System.out.println("arr2 (" + arr2 + ") allocated");
}
```

java -Xmx768m Test →

```
arr1 ([I@4aa8f0b4) allocated
Exception in thread "main"
java.lang.OutOfMemoryError: ...
```

Пример #2: ООП читается на Java. 8 строчек кода:

```
public static void main(String[] args) {
    final int size = 512*1024*1024 / Integer.BYTES;
    int[] arr1 = new int[size];                                  512mb
    System.out.println("arr1 (" + arr1 + ") allocated");
    System.gc();
    int[] arr2 = new int[size];                                  512mb
    System.out.println("arr2 (" + arr2 + ") allocated");
}
```

Интерпретатор не вычисляет время жизни!

Пример #2: ООП читается на Java. 8 строчек кода:

```
public static void main(String[] args) {
    final int size = 512*1024*1024 / Integer.BYTES;
    int[] arr1 = new int[size];
    System.out.println("arr1 (" + arr1 + ") allocated");
    System.gc();
    int[] arr2 = new int[size];
    System.out.println("arr2 (" + arr2 + ") allocated");
}
```

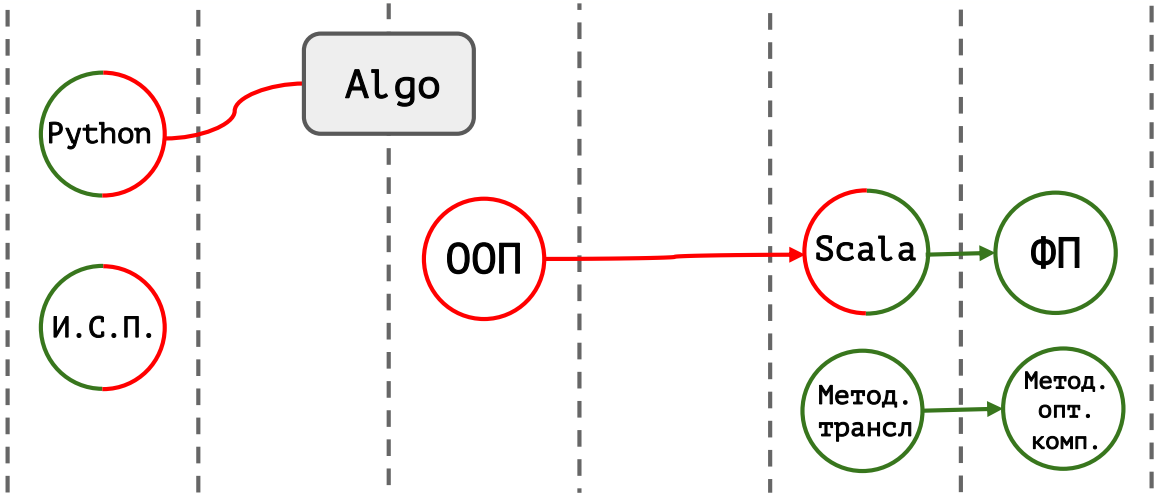
512mb

512mb

Интерпретатор не вычисляет время жизни!
(предполагает самый худший случай)

Как мы преподаем: содержание

- С низкоуровневыми предметами - понятно, фокус на реализации и на связи с машиной
- Зачем нужны красные (прикладные) курсы, и что на них?



Как мы преподаем: содержание

- С низкоуровневыми предметами - понятно, фокус на реализации и на связи с машиной

- Зачем нужны красные (прикладные) курсы, и что на них?

  Прикладные курсы - это еще и повод поговорить про детали реализации соответствующих языков, а в этом и весь смысл

Как мы преподаем: содержание

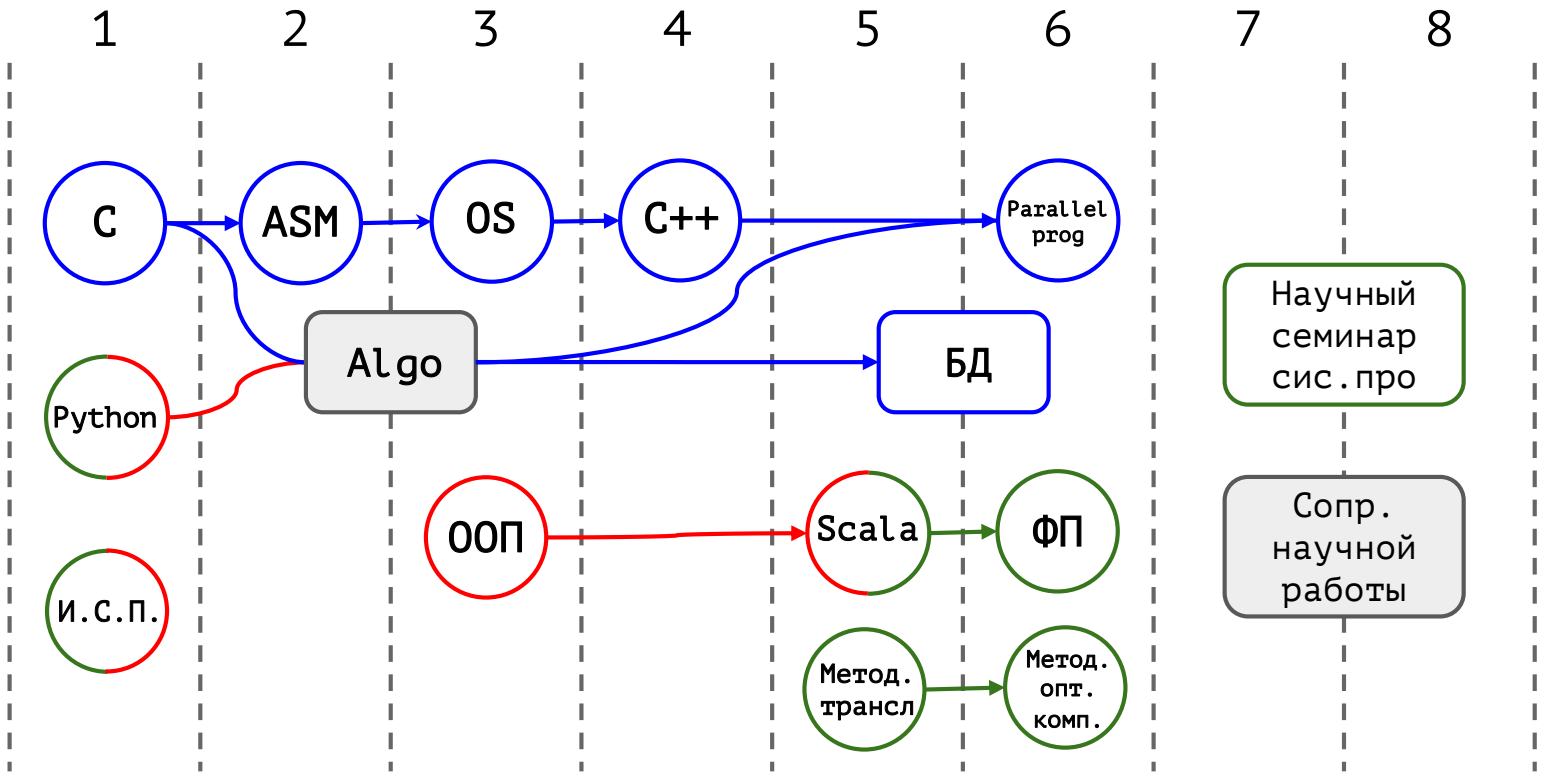
- С низкоуровневыми предметами - понятно, фокус на реализации и на связи с машиной
- Зачем нужны красные (прикладные) курсы, и что на них?

  Прикладные курсы - это еще и повод поговорить про детали реализации соответствующих языков, а в этом и весь смысл
- Зеленые курсы - это, зачастую, уже открытый разговор про компиляторы и рантаймы

■ — "системщина", низкий уровень

■ — "программистский кругозор", какие языки, как реализуют

■ — "прикладное" программирование



Как мы преподаем: содержание

- С низкоуровневыми предметами - понятно, фокус на реализации и на связи с машиной

- Зачем нужны красные (прикладные) курсы, и что на них?

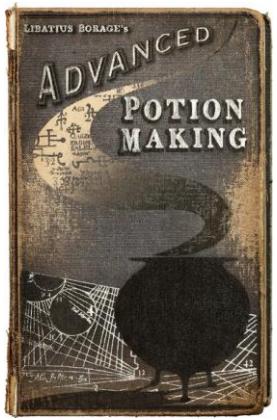
  Прикладные курсы - это еще и повод поговорить про детали реализации соответствующих языков, а в этом и весь смысл

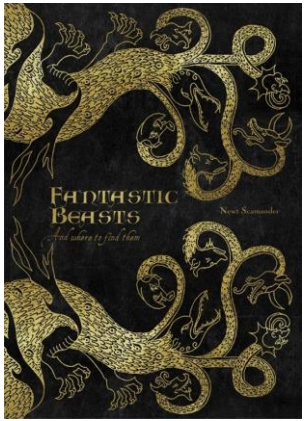
- Зеленые курсы - это, зачастую, уже открытый разговор про компиляторы и рантаймы

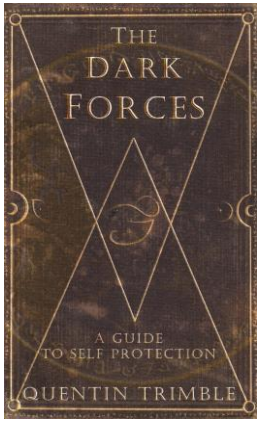
  ✓ Даже в курсе по ФП говорим в том числе про реализацию GHC

Как построить Хогвартс?

- Нужен замок ✓
- Команда волшебников-преподавателей, ✓
- Магическая учебная программа, ✓







Как построить Хогвартс?

- Нужен замок ✓
- Команда волшебников-преподавателей, ✓
- Магическая учебная программа, ✓
- Броское название и символика!


N*
Новосибирский
государственный
университет
*НАСТОЯЩАЯ НАУКА

Назвать решили "системное программирование" на ММФ НГУ

((N)nullptr)

*The real system programming

MMF NSU

Назвать решили "системное программирование" на ММФ НГУ

Как построить Хогвартс?

- Нужен замок ✓
- Команда волшебников-преподавателей, ✓
- Магическая учебная программа, ✓
- Броское название и символика! ✓

Как построить Хогвартс?

- Нужен замок ✓
- Команда волшебников-преподавателей, ✓
- Магическая учебная программа, ✓
- Броское название и символика, ✓
- Отбор способных к магии студентов

Как отбирать абитуриентов?

Как отбирать абитуриентов?

- Как выделить заинтересованных? Кому интересна реализация и дизайн языков программирования?

Как отбирать абитуриентов?

- Как выделить заинтересованных? Кому интересна реализация и дизайн языков программирования?

  Собеседования!

Как отбирать абитуриентов?

- Как выделить заинтересованных? Кому интересна реализация и дизайн языков программирования?

  Собеседования!

  Лето, июль, ~60 собеседований абитуриентов.

Пример вопроса на собеседовании:

Пример вопроса на собеседовании:

Q: Как проверить число на четность?

Пример вопроса на собеседовании:

```
Q: Как проверить число на четность?
A: *Настороженно* Взять остаток от деления на 2?
```

Пример вопроса на собеседовании:

Q: Как проверить число на четность?
A: *Настороженно* Взять остаток от деления на 2?

Q: Отлично! А теперь, допустим, в вашем языке нет оператора взятия остатка от деления, как тогда?

Пример вопроса на собеседовании:

```
Q: Как проверить число на четность?
A: *Настороженно* Взять остаток от деления на 2?

Q: Отлично! А теперь, допустим, в вашем языке нет
оператора взятия остатка от деления, как тогда?
…
A: Привести к строке и посмотреть на последнюю цифру?
```

Пример вопроса на собеседовании:

```
Q: Как проверить число на четность?
A: *Настороженно* Взять остаток от деления на 2?

Q: Отлично! А теперь, допустим, в вашем языке нет
оператора взятия остатка от деления, как тогда?
…
A: Привести к строке и посмотреть на последнюю цифру?

Q: Неплохо, но ведь кто-то написал это приведение к строке?
Как же он сделал это без %?
```

Пример вопроса на собеседовании:

Q: Как проверить число на четность?
A: *Настороженно* Взять остаток от деления на 2?

Q: Отлично! А теперь, допустим, в вашем языке нет оператора взятия остатка от деления, как тогда?
…
A: Привести к строке и посмотреть на последнюю цифру?

Q: Неплохо, но ведь кто-то написал это приведение к строке? Как же он сделал это без %?

…

A: может тогда вычитать по 2, пока не дойдем до 0 или 1?

Q: Отлично! А теперь в вашем языке нет - и +, как тогда?
A: …

Пример вопроса на собеседовании:

```
Q: Как проверить число на четность?
A: *Настороженно* Взять остаток от деления на 2?
-------------------------------------------------
Q: Отлично! А теперь, допустим, в вашем языке нет
оператора взятия остатка от деления, как тогда?
…
A: Привести к строке и посмотреть на последнюю цифру?
-------------------------------------------------
Q: Неплохо, но ведь кто-то написал это приведение к строке?
Как же он сделал это без %?


…
-------------------------------------------------
A: может тогда вычитать по 2, пока не дойдем до 0 или 1?

Q: Отлично! А теперь в вашем языке нет - и +, как тогда?
A: …
```

Пример вопроса на собеседовании:

```
Q: Как проверить число на четность?
A: *Настороженно* Взять остаток от деления на 2?
```

Раунд 0

```
Q: Отлично! А теперь, допустим, в вашем языке нет
оператора взятия остатка от деления, как тогда?
…
A: Привести к строке и посмотреть на последнюю цифру?
```

Раунд 1

```
Q: Неплохо, но ведь кто-то написал это приведение к строке?
Как же он сделал это без %?


…
```

Раунд 2

```
A: может тогда вычитать по 2, пока не дойдем до 0 или 1?

Q: Отлично! А теперь в вашем языке нет - и +, как тогда?
A: …
```

Раунд 3

Как отбирать абитуриентов?

- Как выделить заинтересованных?

  Ряд вопросов, подводящих к сравнению разных языков, пониманию уровня абитуриента (он должен быть высоким), и заинтересованности в вопросах реализации.

Как отбирать абитуриентов?

- Выделяем подходящих нам абитуриентов

Как отбирать абитуриентов?

- Выделяем подходящих нам абитуриентов

- Как выиграть конкуренцию за них?

Как отбирать абитуриентов?

- Выделяем подходящих нам абитуриентов 
- Как выиграть конкуренцию за них? 
  - Конкуренцию с другими ВУЗ-ами (МФТИ, СПбГУ, ИТМО)

Как отбирать абитуриентов?

- Выделяем подходящих нам абитуриентов 
- Как выиграть конкуренцию за них? 
  - — Конкуренцию с другими ВУЗ-ами (МФТИ, СПбГУ, ИТМО)
  - — Конкуренцию с другими факультетами НГУ (с Факультетом Информационных Технологий, например)

Как отбирать абитуриентов?

- Выделяем подходящих нам абитуриентов 
- Как выиграть конкуренцию за них? 
  - Конкуренцию с другими ВУЗ-ами (МФТИ, СПбГУ, ИТМО)
  - Конкуренцию с другими факультетами НГУ (с Факультетом Информационных Технологий, например)
  - И даже с другими профилями на ММФ!

Как отбирать абитуриентов?

- Выделяем подходящих нам абитуриентов

- Как выиграть конкуренцию за них?

— Тяжело.

Как отбирать абитуриентов?

- Выделяем подходящих нам абитуриентов

- Как выиграть конкуренцию за них?

— Тяжело. Особенно с учетом глобальной неразберихи, возникающей ближе к зачислению во всех ВУЗ-ах

Как отбирать абитуриентов?

- Выделяем подходящих нам абитуриентов

- Как выиграть конкуренцию за них?

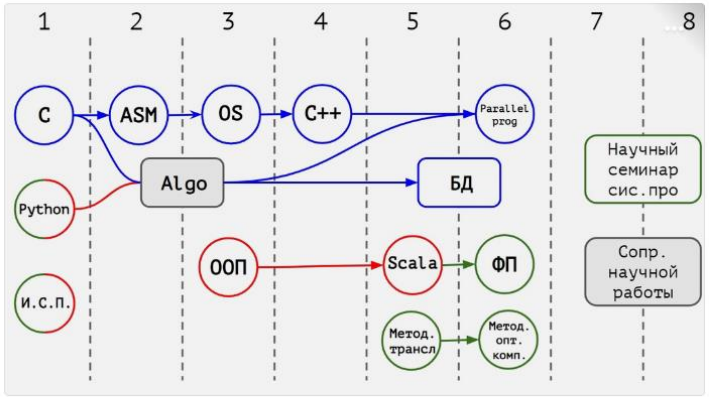
  — Тяжело.

  — Помогает собеседование (особенно в офисе компании).

Как отбирать абитуриентов?

- Выделяем подходящих нам абитуриентов 
- Как выиграть конкуренцию за них? 
  - Тяжело.
  - Помогает собеседование (особенно в офисе компании).
  - Еще помогает четкое описание профиля, предметов на нем, их взаимосвязей и мотивации.

Как отбирать абитуриентов?



Чему мы учим?

Обучение в бакалавриате состоит из 8 семестров – по 2 семестра на каждый учебный год. После каждого семестра вас ждет сессия, на которой..

1 курс, 2 семестр

Архитектура ЭВМ и низкоуровневое программирование (**Assembly**) – курс, продолжающий знакомство студентов с низким уровнем. На этот раз мы погружаемся на уровень команд процессора и изучаем язык ассемблера, а также низкоуровневую архитектуру: регистры процессора, регистры флагов, способы адресации и так далее.

Алгоритмы и структуры данных

Назовете ли вы структуру данных с этой картинки?

Одну и ту же задачу можно решить многими способами, но какой из них лучше? И что вообще такое *лучше* в контексте алгоритмов? В этом курсе вы узнаете ответы на эти вопросы, познакомитесь со многими классическими (и не только!) алгоритмами, попрактикуетесь в реализации интересных структур данных. Здесь мы не привязываемся к какому-то конкретному языку программирования, давая вам свободу выбора инструмента для решения практических задач.

Как к нам поступить?

Профиль «Системное программирование» – особое место: дополнительная нагрузка в...

«Системное программирование». Максимально коротко.

Прочитать подробное описание нашего профиля можно здесь, а посмотреть его...

Как отбирать абитуриентов?

- Выделяем подходящих нам абитуриентов

- Как выиграть конкуренцию за них?

  - Тяжело.
  - Помогает собеседование (особенно в офисе компании).
  - Еще помогает четкое описание профиля, предметов на нем, их взаимосвязей и мотивации.
  - Ходить в школы, рассказывать старшеклассникам о sys.pro по несколько раз.

Через несколько месяцев набрали первую группу

Через несколько месяцев набрали первую группу, через год еще одну,

Через несколько месяцев набрали первую группу, через год еще одну,
и сейчас идет третий набор

Как построить Хогвартс?

- Нужен замок ✓
- Команда волшебников-преподавателей, ✓
- Магическая учебная программа, ✓
- Броское название и символика, ✓
- Отбор способных к магии студентов ✓

Как построить Хогвартс?

- Нужен замок ✓
- Команда волшебников-преподавателей, ✓
- Магическая учебная программа, ✓
- Броское название и символика, ✓
- Отбор способных к магии студентов, ✓
- Правила, дисциплина и оценки

Как именно мы преподаем?

Как именно мы преподаем?

- Долой разделение на лекции и семинары, зачем оно? (группа то одна)

Как именно мы преподаем?

- Долой разделение на лекции и семинары, зачем оно? (группа то одна)
- Балльно-рейтинговая система => большая часть оценки всегда идет от практики

Как именно мы преподаем?

- Долой разделение на лекции и семинары, зачем оно? (группа то одна)
- Балльно-рейтинговая система => большая часть оценки всегда идет от практики
- Практика: прохождение автотестов и обязательное Code Review. Каждой задачи. Каждого студента.

Как именно мы преподаем?

- Долой разделение на лекции и семинары, зачем оно? (группа то одна)
- Балльно-рейтинговая система => большая часть оценки всегда идет от практики
- Практика: прохождение автотестов и обязательное Code Review. Каждой задачи. Каждого студента.
- Преподы на связи в телеграме 

Как именно мы преподаем?

- Долой разделение на лекции и семинары, зачем оно? (группа то одна)
- Балльно-рейтинговая система => большая часть оценки всегда идет от практики
- Практика: прохождение автотестов и обязательное Code Review. Каждой задачи. Каждого студента.
- Преподы на связи в телеграме

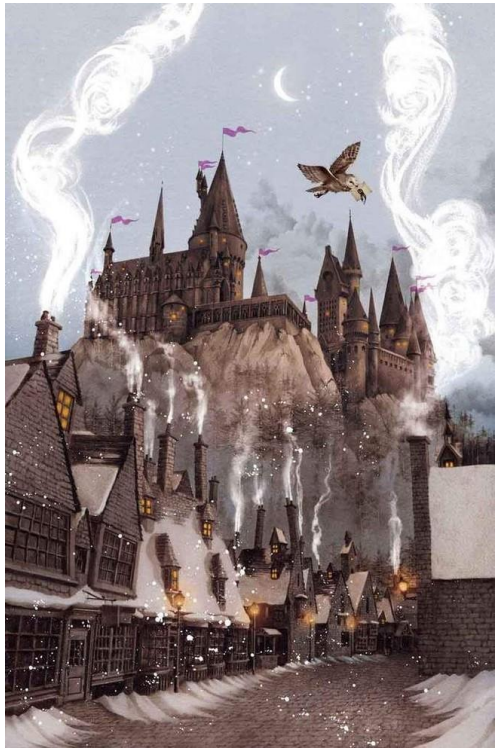
Единственность группы - очень важна для всего!

Как построить Хогвартс?

- Нужен замок ✓
- Команда волшебников-преподавателей, ✓
- Магическая учебная программа, ✓
- Броское название и символика, ✓
- Отбор способных к магии студентов, ✓
- Правила, дисциплина и оценки ✓

Как построить Хогвартс?

- Нужен замок ✓
- Команда волшебников-преподавателей, ✓
- Магическая учебная программа, ✓
- Броское название и символика, ✓
- Отбор способных к магии студентов, ✓
- Правила, дисциплина и оценки ✓
- Хогсмид!



Помогаем студентам выдержать нагрузку

Помогаем студентам выдержать нагрузку

1. Балансируем и ограничиваем нагрузку с учетом математики

Помогаем студентам выдержать нагрузку

1. Балансируем и ограничиваем нагрузку с учетом математики
2. Постоянно собираем со студентов фидбек
   и реагируем на него!

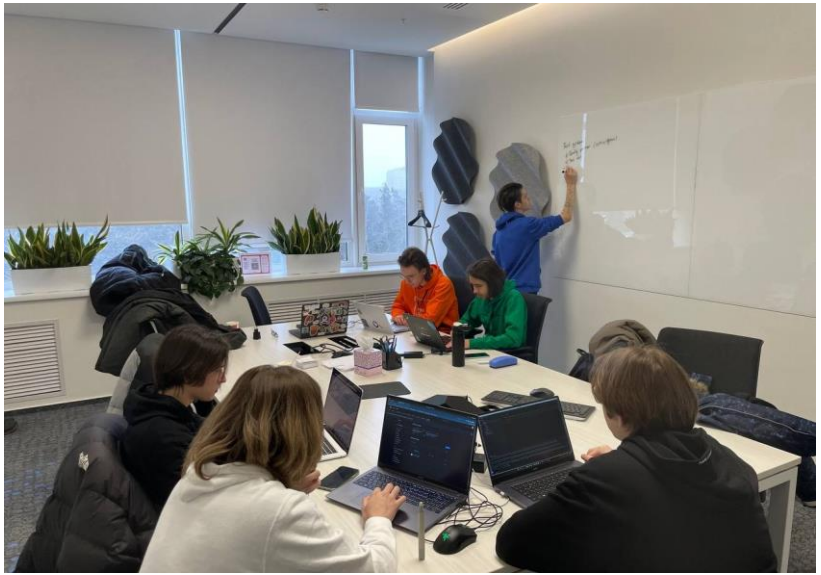
Помогаем студентам выдержать нагрузку

1. Балансируем и ограничиваем нагрузку с учетом математики
2. Постоянно собираем со студентов фидбек
   и реагируем на него!
3. Организуем внеклассные активности

Квесты

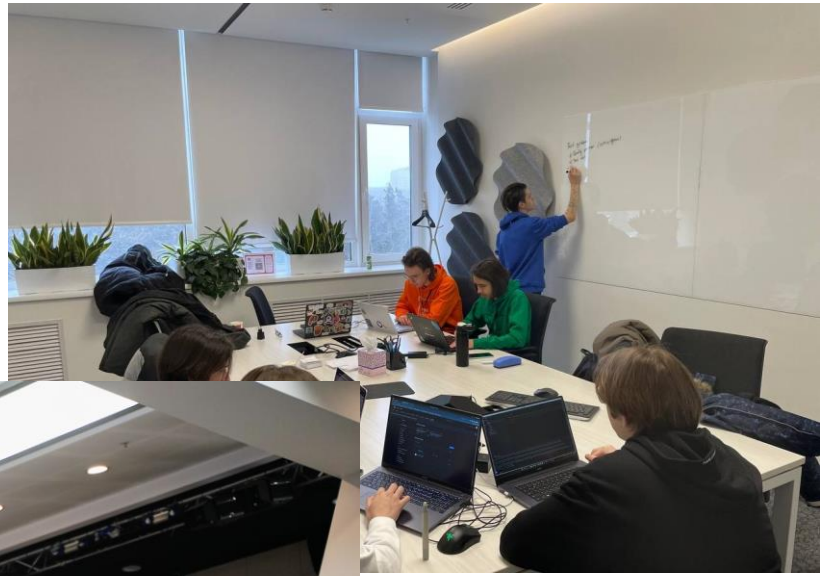
Квесты

Хакатоны



Квесты

Хакатоны



Конференции

Помогаем преподам выдержать нагрузку

Помогаем преподам выдержать нагрузку

1. Автоматизируем все, что можно автоматизировать

Помогаем преподам выдержать нагрузку

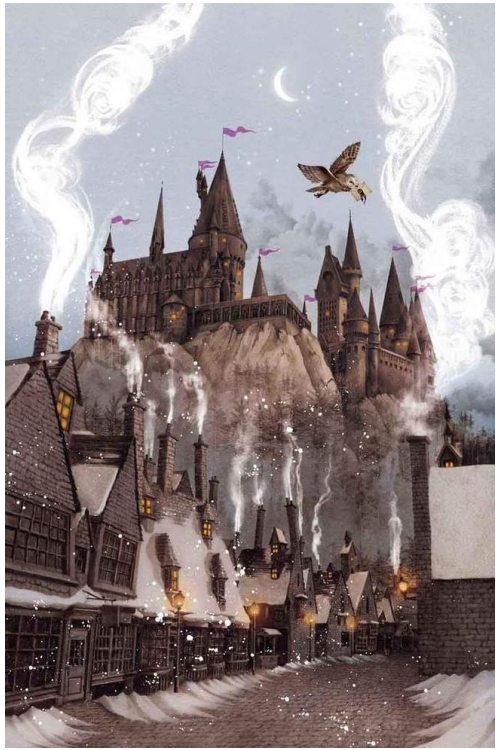
1. Автоматизируем все, что можно автоматизировать
2. Усиливаем преподавателей ассистентами

Помогаем преподам выдержать нагрузку

1. Автоматизируем все, что можно автоматизировать
2. Усиливаем преподавателей ассистентами
3. Программа финансовой поддержки от Huawei

Как построить Хогвартс?

- Нужен замок ✓
- Команда волшебников-преподавателей, ✓
- Магическая учебная программа, ✓
- Броское название и символика, ✓
- Отбор способных к магии студентов, ✓
- Правила, дисциплина и оценки ✓
- Хогсмид! ✓



Теперь у нас есть Хогвартс!

Теперь у нас есть Хогвартс!

И как теперь с волшебниками?

СисПро: промежуточные выводы

СисПро: промежуточные выводы

1. Выживаемость студентов:

```
2 курс: за первые 3 семестра двоих отчислили, одного добрали
```

СисПро: промежуточные выводы

1. Выживаемость студентов:

```
2 курс: за первые 3 семестра двоих отчислили, одного добрали
1 курс: за первый семестр четверых отчислили, троих  добрали
```

СисПро: промежуточные выводы

1. Выживаемость студентов:

```
2 курс: за первые 3 семестра двоих отчислили, одного добрали
1 курс: за первый семестр четверых отчислили, троих  добрали
```

2. Конкурс:

```
Растет: в первый год было 3 человека на места, во второй уже 4,
В этом надеемся на конкурс не меньше, абитуриенты уже спрашивают
```

СисПро: промежуточные выводы

1. Выживаемость студентов:

   2 курс: за первые 3 семестра двоих отчислили, одного добрали
   1 курс: за первый семестр четверых отчислили, троих добрали

2. Конкурс:

   Растет: в первый год было 3 человека на места, во второй уже 4,
   В этом надеемся на конкурс не меньше, абитуриенты уже спрашивают

3. Студенты и стажировки в Excelsior:

   2 подавались на стажировку,
   1 прошел,
   Еще 3-ое решают задачи в данный момент

СисПро: что дальше?

СисПро: что дальше?

1. Подготовить систему стажировок в разных компаниях для старшекурсников

СисПро: что дальше?

1. Подготовить систему стажировок в разных компаниях для старшекурсников

2. Продолжать вовлекать студентов в организацию всего профиля в качестве:
   - Кураторов
   - Ассистентов
   - Через какое-то и преподавателей

СисПро: что дальше?

1. Подготовить систему стажировок в разных компаниях для старшекурсников

2. Продолжать вовлекать студентов в организацию всего профиля в качестве:
   - Кураторов
   - Ассистентов
   - Через какое-то и преподавателей

3. Дорабатывать и улучшать наши курсы:
   - Новый курс лучше старых двух

СисПро: что дальше?

1. Подготовить систему стажировок в разных компаниях для старшекурсников

2. Продолжать вовлекать студентов в организацию всего профиля в качестве:
   - Кураторов
   - Ассистентов
   - Через какое-то и преподавателей

3. Дорабатывать и улучшать наши курсы

4. Никакого масштабирования! Только 1 группа, только бакалавриат.

Вам нужно в университет

1. Для некоторых областей IT университет — это отличное место для подготовки специалистов.

Вам нужно в университет

1. Для некоторых областей IT университет — это отличное место для подготовки специалистов.

2. Университет — не болото. Можно прийти и сделать хорошо.

Вам нужно в университет

1. Для некоторых областей IT университет — это отличное место для подготовки специалистов.

2. Университет — не болото. Можно прийти и сделать хорошо.

3. Студенты потрясающие 

Вам нужно в университет

1. Для некоторых областей IT университет — это отличное место для подготовки специалистов.

2. Университет — не болото. Можно прийти и сделать хорошо.

3. Студенты потрясающие ❤️‍🩹

   — Они очень умны, их контекст потрясает уже сейчас

Вам нужно в университет

1. Для некоторых областей IT университет — это отличное место для подготовки специалистов.

2. Университет — не болото. Можно прийти и сделать хорошо.

3. Студенты потрясающие 💔

   — Они очень умны, их контекст потрясает уже сейчас

   — Они дружны, sys.pro ощущается сформированной экосистемой

Вам нужно в университет

1. Для некоторых областей IT университет — это отличное место для подготовки специалистов.

2. Университет — не болото. Можно прийти и сделать хорошо.

3. Студенты потрясающие 💔

   — Они очень умны, их контекст потрясает уже сейчас

   — Они дружны, sys.pro ощущается сформированной экосистемой

   — Они очень помогают в создании sys.pro

Вам нужно в университет

1. Для некоторых областей IT университет — это отличное место для подготовки специалистов.
2. Университет — не болото. Можно прийти и сделать хорошо.
3. Студенты потрясающие ❤️‍🩹

То, что начиналось, как проект по оптимизации стажировок, превратилась во что-то намного большее лично для меня

Q & A

@dbg_nsk

@ugliansky

>_
sys.pro
mmf.nsu