

*School of
Computer
Science*

ЗНАКОМСТВО

ПРОГРАММИРОВАНИЕ НА PYTHON

Лекции для IT-школы

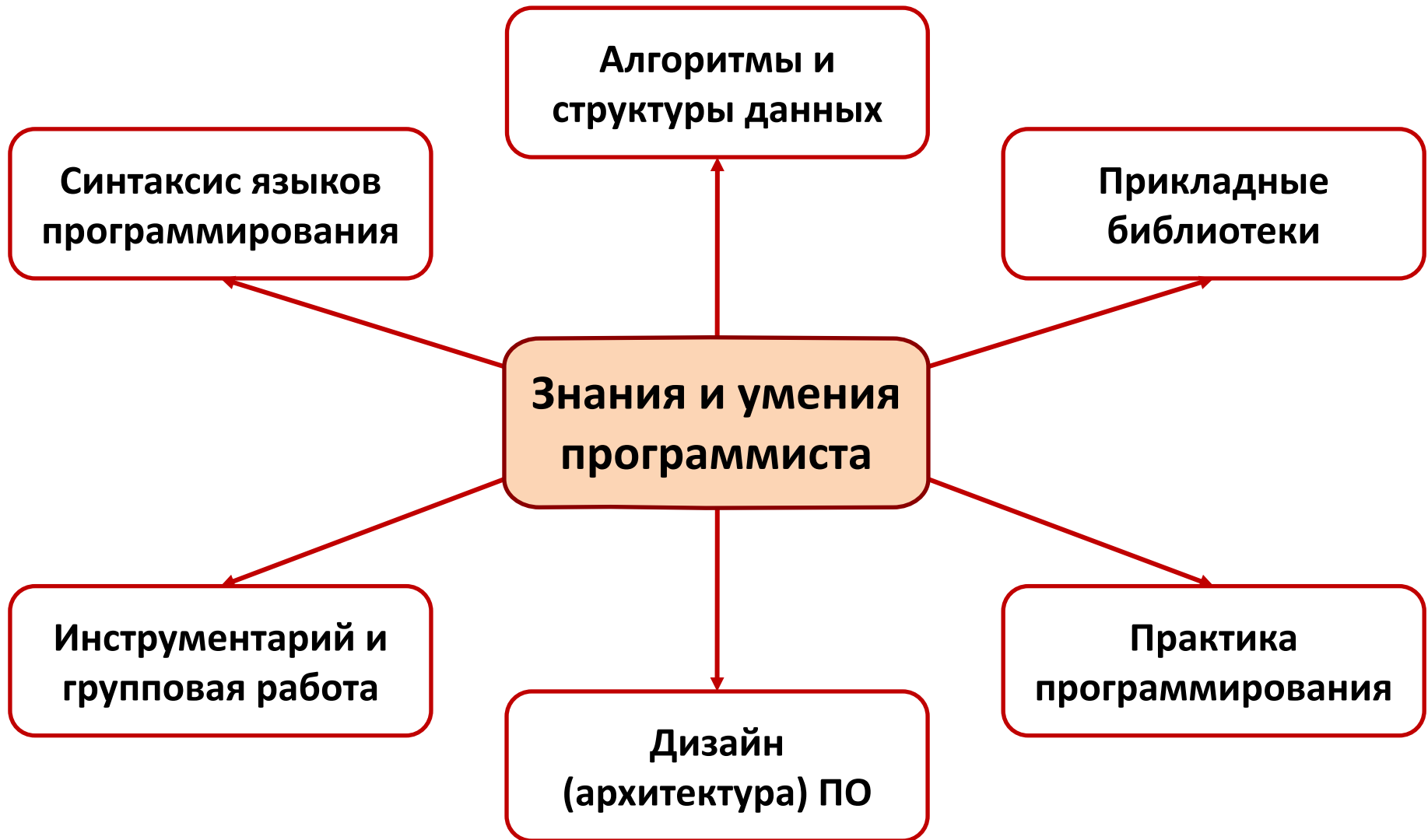


- **Алгоритм:**
 - Последовательность действий, приводящих к желаемому результату
- **Компьютерная программа:**
 - Комбинация компьютерных инструкций и данных, позволяющая компьютеру решать практические задачи
- **Программист:**
 - Специалист, занимающийся написанием и корректировкой программ для компьютеров

Кто знает в какой день отмечается
профессиональный праздник программистов?



ЗНАНИЯ И УМЕНИЯ ПРОГРАММИСТА





СПЕЦИАЛИЗАЦИЯ ПРОГРАММИСТА



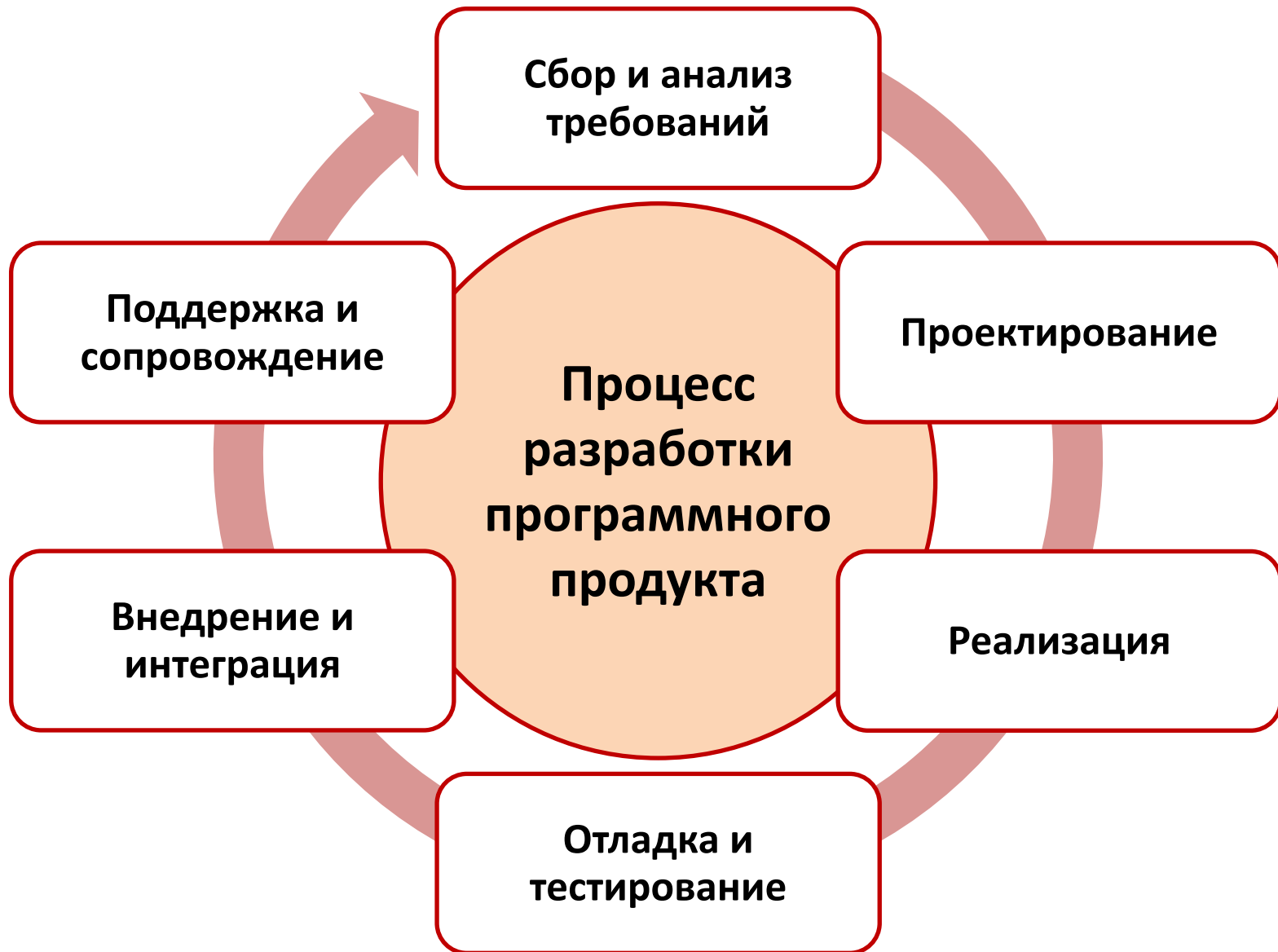


ЧЕМ ЗАНИМАЮТСЯ ПРОГРАММИСТЫ?

- Сбор требований от представителя заказчика
- Анализ и постановка задачи
- Разработка и согласование документов, включая:
 - Описание решения
 - Написание технических спецификаций
- Проектирование программы, включая:
 - Разработка структур данных
 - Разработка алгоритмов
 - Построение интерфейсов
- Отладка и тестирование ПО
- Рецензирование исходного кода своих коллег
- Документирование программного продукта
- Настройка программного продукта и окружения
- Поддержка и сопровождение
- ...



ЭТАПЫ РАЗРАБОТКИ ПРОГРАММНОГО ПРОДУКТА





ЯЗЫКИ ПРОГРАММИРОВАНИЯ. КЛАССИФИКАЦИЯ

- От «низкого» к «высокому»:
 - Машинный код
 - Ассемблеры
 - Высокоуровневые языки
- По способу решения задач:
 - Декларативные (SQL, HTML, ...)
 - Императивные (C, C++, ...)
- По области применения:
 - Языки для системного и прикладного программирования
 - Языки для WEB/Desktop/мобильной разработки
- По парадигме программирования:
 - Процедурные и структурные (Pascal, Basic, ...)
 - Объектные (C++, Java, ...)
 - Функциональные и логические (Lisp, Prolog, ...)
- Список общеизвестных языков программирования:
 - https://en.wikipedia.org/wiki/List_of_programming_languages



ТРАНСЛЯТОРЫ

- Трансляция – превращения исходного кода программы в код на другом языке
- Варианты трансляции:
 - *Компиляция* – выдает бинарный код для исполнения на компьютере
 - *Интерпретация* – последовательное исполнение исходного кода программы при ее выполнении
- Достоинства интерпретации:
 - Быстрая разработка
 - Легкая переносимость между платформами
- Недостатки интерпретации:
 - Относительное снижение скорости исполнения



ХАРАКТЕРИСТИКИ PYTHON

- Универсальный
- Высокоуровневый
- Интерпретируемый
- Портируемый (кроссплатформенный)
- Мульти-парадигменный
- Динамический
- Язык быстрой разработки
- Понятный и краткий в изложении
- Встраиваемый и расширяемый другими языками
- С большим количеством библиотек
- И широкой поддержкой Open Source сообщества
- Хорошо подходит для обучения



ОБЛАСТИ ПРИМЕНЕНИЯ PYTHON

- Обучение
- Научная сфера
- Системное администрирование
- Исследование данных (big data)
- Тестирование
- Web-приложения
- Сервисы
- Разработка игр

<https://www.mvoronin.pro/en/blog/post-75>





ИСТОРИЯ PYTHON

- Выпущен впервые в 1991 году
- Создатель Гвидо Ван Россум, «Великодушный пожизненный диктатор»
- Назван в честь британского телешоу «Летающий цирк Монти Пайтона»
- Создавался как универсальный язык для обучения
- Вобрал в себя множество возможностей из других языков



У всех установлен Python версии 3.x?

Скачивать здесь:

<https://www.python.org/downloads/>





ВЕРСИИ ЯЗЫКА

v.2.7

Поддержка прекращается
в 2020 году

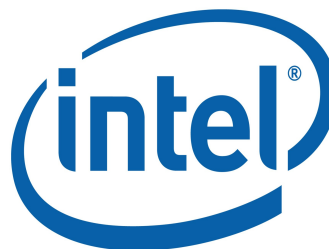
v.3.7

Python версии 3.x
развивается с 2009 года





КТО ИСПОЛЬЗУЕТ PYTHON





ФИЛОСОФИЯ (ДЗЕН) ПИТОНА

Выводится на английском по команде `import this` в Python Shell

- Красивое лучше, чем уродливое
- Явное лучше, чем неявное
- Простое лучше, чем сложное
- Сложное лучше, чем запутанное
- Плоское лучше, чем вложенное
- Разреженное лучше, чем плотное
- Читаемость имеет значение
- Особые случаи не настолько особые, чтобы нарушать правила
- При этом практичность важнее безупречности



ДЗЕН ПИТОНА. ПРОДОЛЖЕНИЕ

- Ошибки никогда не должны замалчиваться
- Если не замалчиваются явно
- Встретив двусмысленность отбрось искушение угадать
- Должен существовать один – и, желательно, только один – очевидный способ сделать это
- Хотя он поначалу может быть и не очевиден, если вы не голландец
- Сейчас лучше, чем никогда
- Хотя никогда зачастую лучше, чем прямо сейчас
- Если реализацию сложно объяснить – идея плоха
- Если реализацию легко объяснить – идея, возможно, хороша
- Пространства имён – отличная штука! Будем делать их побольше!



ССЫЛКИ НА УЧЕБНЫЕ МАТЕРИАЛЫ

- Визуальная среда разработки Python в WEB :
 - <http://www.pythontutor.com/live.html#mode=edit>
- Standard Documentation (ENG):
 - <https://docs.python.org/3.7/tutorial>
 - <https://docs.python.org/3/library/index.html>
- Problem Solving with Algorithms and Data Structures:
 - ENG:
<http://interactivepython.org/runestone/static/pythonds/Introduction/GettingStartedwithData.html>
 - RUS: <https://aliev.me/runestone> – перевод учебника
- Подборка информации по Python в VK:
 - https://vk.com/page-54530371_48786262
- Самоучитель Python:
 - <https://pythonworld.ru/samouchitel-python>





ЗНАКОМСТВО СО СРЕДОЙ ИСПОЛНЕНИЯ

- Start / All Programs / Python 3.x / IDLE
- `exit()` для выхода
- `help()` для получения справки:
 - в режиме помощи
 - по конкретной функции или оператору
- `import this` – Дзен Питона
- Перемещение по истории ввода:
 - Alt+P – перейти к предыдущему
 - Alt+N – перейти к следующему вводу
- Пробуем математические операторы `+` `-` `*` `/` `//` `%` `**`
- Используем `_`:
 - для получения предыдущего значения (работает только в режиме Shell)
 - для разделения длинных чисел на разряды



ОСВОЕНИЕ СРЕДЫ РАЗРАБОТКИ IDLE

- Строки (в кавычках) можно сливать (конкатенировать) с помощью оператора `'+'`
- Переменные – именованные данные, которыми оперирует программа
- Оператор присваивания `'='` нужен для активации переменных
- `input('Приглашение: ')` – ввод строки, результат ввода можно присвоить переменной
- `print(...)` – вывод информации на консоль



ПРАКТИЧЕСКОЕ ЗАДАНИЕ №1

– В скрипте `hello_man.py` текст программы

– Необходимо:

1. Выполнить и отладить эту программу
2. Запросить текущий возраст пользователя
3. Рассчитать и вывести на экран через сколько лет пользователю будет 40 лет



– Используйте `input()` и `print()`

– Сообщайте об ошибках – будем исправлять



ВОЗМОЖНЫЕ ОШИБКИ В ЗАДАНИИ №1

1. Синтаксическая (опечатка)
2. Несовместимые типы данных при вычитании
3. Неверный тип данных при вводе пользователя

Как реагировать:

1. Исправить опечатку
2. Преобразовать строку в число – `int()`
3. Предупредить пользователя и ожидать правильный формат вводимых данных и/или
Фильтровать ввод данных от пользователя



С ВАМИ БУДУТ РАБОТАТЬ

- Лукашенко Сергей Николаевич:
 - Начальник Управления перспективных разработок в Компас Плюс
 - s.lukashenko@compassplus.com
- Ковалёва Александра Дмитриевна:
 - Инженер-программист Бюро мобильных платформ в Компас Плюс
 - Создает мобильные приложения под iOS
 - a.kovaleva@compassplus.com
 - Telegram: @AlexandraKovaleva
- Материалы лекций ищите здесь:
 - gitlab.com/slukashenko75/python-course-2019-2020

СПАСИБО ЗА ВНИМАНИЕ !
ВОПРОСЫ ?



*School of
Computer
Science*