

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский Авиационный Институт»
(Национальный Исследовательский Университет)

Институт: №8 «Информационные технологии и прикладная
математика»
Кафедра: 806 «Вычислительная математика и программирование»

Руферат
по курсу "Фундаментальная
информатика"
I семестр
Тема:
"Язык программирования Python на примере разработки программы-
игры Flappy Bird"

Группа	М8О-109Б-22
Студент	Яшин В.А.
Преподаватель	Сысоев М.А.
Оценка	
Дата	

СОДЕРЖАНИЕ

1. История Python и сферы применения.....	
2. Основные особенности языка.....	
3. Игра Flappy Bird.....	
4. Геймплей.....	
5. Работа с pygame.....	
6. Список литературы.....	

1.История Python и сферы применения

Язык программирования Python является, пожалуй, одним из самых популярных. Используется в веб-разработке, DataScience, системах автоматизации, приложениях – сферы его применения поистине безграничны. Но главное преимущество Питона – это его низкий порог входа. Иными словами, обучиться программированию на этом языке может практически каждый.

Python обладает простым и понятным синтаксисом. Даже далекому от программирования человеку будет несложно разобраться в строках простенького кода. Какими еще преимуществами обладает этот язык, почему его выбирает весь мир, как и где лучше изучать Python, вы узнаете из нашего материала.

Краткая история создания:

Python сегодня входит в число наиболее популярных языков программирования.

Неудивительно, что столько начинающих кодеров задаются вопросом, трудно ли учить язык программирования Python и есть ли вообще в этом смысл.

Python входит в самые различные информационные экосистемы, но, по большей части, он используется в бэкенде. Такие гиганты разработки, как Google, Facebook*, Stripe, Spotify, Netflix, применяют его в своих продуктах. Поэтому можно не сомневаться, что язык программирования Python способен стать неплохим стартом карьеры для программиста.

Python является языком программирования высокого уровня общего назначения. Сфера его применения весьма широка. В разработке приложений и веб-сайтов он тоже активно используется. Этот интерпретируемый объектно-ориентированный язык имеет открытый исходный код и динамическую семантику. Сам Python создавался посредством языка “Си”.

В 1989 г. Гвидо Ван Россум создал новый язык программирования под названием Python, а в 1991 г выпустил его. Главная цель, которую ставил перед собой автор — это упрощение процесса программирования. Чтобы писать код было проще, он должен стать более читабельным и понятным для человека. У Python открытый исходный код. Одно из преимуществ этого языка — возможность запускать программы на нём как на ОС Windows, так и на macOS и Linux.

Код, написанный на “питоне”, действительно легко читать и разбираться в его структурных элементах. Это делает язык программирования Python подходящим для начинающих. Но его возможности не сводятся лишь к базовым: на Python написаны и эффективно поддерживаются сложнейшие веб-платформы и высоконагруженные приложения.

Основные сферы применения:

web-разработка:

Для разработки веба подключают специальные фреймворки, наиболее

популярными являются Django и Flask. Благодаря фреймворкам, работа по написанию веб-приложений значительно упрощается. К примеру, фреймворки помогают реализовывать нужные бизнес-процессы, облегчают рефакторинг старых сайтов, упрощают применение Ajax и так далее.

Два слова о Django и Flask:

Flask. Гибок, прост в эксплуатации, минималистичен;

Django. «Все включено»: административная панель, ORM, интерфейс базы данных, структура каталогов для готовых проектов и программных приложений.

Big Data и Machine learning (машинное обучение, работа с данными: анализ, визуализация и т. д.):

Машинное обучение сегодня более чем популярно и у всех на слуху. При использовании Machine learning (ML) задействуются методы работы с искусственным интеллектом. Существует много алгоритмов машинного обучения, которые применяются в системах распознавания лиц, голосов, изображений и прочих системах. Всего не перечесать. Если интересуют подробности, основные алгоритмы машинного обучения и примеры использования ML для решения реальных проблем, вам сюда. Также может представлять определенный интерес подборка Open Source-проектов по машинному обучению.

Таким образом, именно Python в силу своих возможностей чаще всего используется в Data Science. С его помощью пишут алгоритмы ML-программ и аналитические приложения. Среди популярных фреймворков и библиотек для ML можно выделить scikit-learn и TensorFlow.

Нельзя отдельно не упомянуть анализ и визуализацию данных, где тоже

задействуется Python. Одна из наиболее популярных библиотек для визуализации — Matplotlib.

Также «Питон» подходит для написания парсеров, позволяющих выполнять сбор информации в глобальной сети. А еще с его помощью обслуживают облачные сервисы, хранилища данных и т. п.

скриптинг (написание скриптов автоматизации):

Под скриптингом понимают написание скриптов, позволяющих автоматизировать простые задачи. Это могут быть дополнительные модули, плагины к уже готовому программному обеспечению и т. д.

Широко применяется «Питон» и при автоматизации тестирования ПО. Таким образом, если вы освоите этот язык, вы сможете развивать свою карьеру по целому ряду IT-направлений:

— программист (бэкенд-разработчик);

— дата-инженер, data scientist, специалист по машинному обучению;

— QA-инженер (тестировщик) с уклоном в автоматизацию тестирования.

Причины популярности:

Интерпретируемость — одно из главных преимуществ “питона”. Напомним, что интерпретируемым называется тот язык, код на котором не нужно компилировать, а можно сразу запускать.

Запуск кода на Python возможен на любом ПК, интерпретаторы для него есть

езде. А это значит, что разработчик может сразу видеть результат. Однако, будучи интерпретируемым языком, Python уступает по скорости компилируемым языкам (поскольку не работает непосредственно на компьютере).

Интерпретируемость сильно упрощает тестирование и перемещение с платформы на платформу небольших блоков кода. Совместимость с большинством существующих ОС делает Python универсальным языком программирования. Он прекрасно подходит для тех, кто только начинает программировать. Данный высокоуровневый язык позволяет разработчику сосредоточиться на сути и целях, а не конкретных шагах и процедурах. По этой причине писать скрипты на Python оказывается быстрее, чем на многих других языках.

Его сходство с естественным языком (английским) сильно облегчает его изучение, если сравнивать Python с другими языками программирования. Разработчику легко запомнить синтаксис “питона” и начать читать код.

Помимо небольших прикладных скриптов, на Python можно писать и крупные коммерческие приложения, думая только о том, как обеспечить им необходимый функционал (все мелкие задачи решаются “под капотом”, и программисту не нужно беспокоиться о них). Надёжность делает язык программирования Python очень популярным.

Это четыре причины столь трепетной привязанности к “питону” со стороны программистов, хакеров и дата-сайентистов.

Ключевым фактором при его выборе является гибкость этого языка программирования и его соответствие ООП-парадигме. Поэтому его активно применяют в Data Science и машинном обучении.

2.Основные особенности языка

Высокоуровневый

Языки программирования делятся на высокоуровневые и низкоуровневые.

Низкоуровневые языки — языки, близкие к машинному коду или его конструкциям (например, байт-кодам). Классикой таких языков являются C, Assembler, Forth.

Высокоуровневые языки — соответственно разрабатываются для удобства использования и скорости написания программы. В них применяются определённые абстракции — структуры данных, набор вспомогательных функций и так далее. Это такие языки как Python, JS, PHP, Go.

Интерпретируемый

Языки делятся на интерпретируемые (Python, JS, PHP, R, Ruby) и компилируемые (C, C++, Pascal). В первом случае программа выполняется специальной программой — интерпретатором, во втором программа сначала преобразуется в понятные компьютеру исполняемые файлы.

Объектно-ориентированный

Все языки также разделяются на процедурные, функциональные и объектно-ориентированные в зависимости от того, с помощью каких конструкций создаётся программа и как происходит её выполнение.

В объектно-ориентированных языках основа это классы и экземпляры классов это равносильно типу и объекту этого типа. Выполнение условных задач или же просто работа программы строится на взаимодействии различных классов.

Python хоть и является объектно-ориентированным языком но также поддерживает и процедурное программирование это значит, что программу можно написать без единого класса.

В основе функциональных языков лежит отличная от предыдущих вычислительная система, называемая лямбда-исчисление, которая, тем не менее, эквивалентна машине Тьюринга, о чём доказана соответствующая теорема

(спасибо моему преподавателю за этот абзац).

3.Игра Flappy Bird

Суть игры заключается в том чтобы как можно дольше "пролететь" на птичке ну убившись об столб

есть несколько основных частей

начало

когда птичка просто стоит на месте и ждет когда вы нажмете в первый раз для начала движения

полет

после первого клика начинается движение

конец

коснувшись столба птичка "погибает" и игра начинается с начала

по сути это весь геймплей игры

программа будет основанна на pygame

4.Геймплей

начало - в программе помечается как "start"

if state == "start":

 if click and timer == 0 and len (pipes)== 0:

 state = "play"

 py += (HEIGHT // 2 - py)*0.1

 player.y = py

что в этом этапе:

сигнал к началу в виде клика мышкой

таймер что не позволяет играть сразу после "смерти"

условие на то что мы можем начать играть только когда пропадут столбы
здесь начало 2 этапа
начальное положение птички

полет - в программе помечается как "play"

```
elif state == "play":
```

```
    if click :
```

```
        ay = -2
```

```
    else:
```

```
        ay = 0
```

```
    py+=sy
```

```
    sy = (sy+ay+1)*0.98
```

```
    player.y = py
```

```
if len(pipes)==0 or pipes[len(pipes)-1].x<WIDTH-200:
```

```
    pipes.append(pygame.Rect(WIDTH, 0, 50, 200))
```

```
    pipes.append(pygame.Rect(WIDTH, 400, 50, 200))
```

```
if player.top < 0 or player.bottom > HEIGHT:
```

```
    state = "fall"
```

```
for pipe in pipes:
```

```
    if player.colliderect(pipe):
```

```
        state = "fall"
```

что в этом этапе:

движение птички вверх при нажатии

скорость движения

появление столбов

условие что касание границ экрана (потолок и пол) это смерть птички

условие что касание столбов это смерть птички

конец - в программе помечается как "fall"

```
elif state == "fall":
```

```
    sy, ay = 0, 0
```

```
    state = "start"
```

```
    timer = 10
```

что в этом этапе:

обнуляются скорость и ускорение

восстанавливается таймер

переходим к этапу начало

4.Работа с pygame

были использованны следующие функции

pygame.display Доступ к дисплею

pygame.draw Рисует фигуры, линии и точки

pygame.event Управление внешними событиями

pygame.mouse Управляет мышью

pygame.rect Управляет прямоугольными областями

pygame.time модуль pygame для управления временем и частотой кадров

pygame.key Считывает нажатия клавиш с клавиатуры

pygame.event.get получает события из очереди

pygame.QUIT передает значение False в результате чего игровой цикл

заканчивается.

`event.type` используется для ссылки на определённое событие
`Colliderect(Rect)` – проверка пересечения двух прямоугольников
`pygame.Color()` присвоивание цвета

5.Список литературы

<https://pg1.readthedocs.io/en/latest/ref/pygame.html>

<https://mrtsepa.gitbooks.io/pygame-tutorial/content/reference/pygame/events.html>

<https://waksoft.susu.ru/2019/04/24/pygame-shpargalka-dlja-ispolzovanija/>

<https://gb.ru/blog/yazyk-programmirovaniya-python/>