

Муниципальное бюджетное образовательное учреждение
Заларинская средняя общеобразовательная школа № 1

Индивидуальный итоговый проект
«Игры на Python»

Автор: Савченко Татьяна
ученица 10 «а» класса,

Руководитель:
Кузеванова Яна Кирилловна,
учитель математики и информатики
МБОУ Заларинская СОШ №1

р.п.Залари
2023 год

Оглавление

Введение	3
Основная часть	4
1.1. История языка программирования Python.....	4
1.2. Основы языка программирования Python.....	4
1.3. Плюсы и минусы языка программирования Python.....	6
Практическая часть	8
2.1. Установка утилиты для языка Python	8
2.2. Создание файла для работы с Python	8
2.3. Разработка игры	8
Заключение	10
Список литературы	11

Введение

Все мы знаем, что в России на уроках информатики по программированию ученики учатся на языке Pascal. Но так ли это актуально сейчас? Наука и технологии сегодня развиваются столь стремительно, что возникает необходимость серьезно заниматься программированием, начиная со школы. Хотя этот язык и перестал развиваться – многие считают, что язык по-прежнему выполняет функцию обучения языку программирования. Большой минус языка **Pascal** – отсутствие мотивации к изучению этого языка школьниками.

Python – удобный и довольно модный сам по себе язык программирования, живой и перспективный, обеспечивающий легкий старт. Он, конечно, не обладает такой же строгостью Паскаля, что есть не всегда хорошо для развития алгоритмического мышления, формирование алгоритмической культуры.

Актуальность темы: В настоящее время все большую популярность приобретает язык Python. Одна из причин популярности Python – более простое и компактное оформление, чем в других языках. Это самый популярный язык общего назначения: он используется для машинного обучения, аналитике, разработке игр и в науке о данных.

Цель работы: разработать собственную игру, используя язык программирования Python.

Задачи:

1. Изучить имеющуюся литературу об основах языка программирования.
2. Определить и разработать тематику и сюжет будущей игры.
3. Создать игру, в соответствии с придуманным сюжетом и тематикой

Тип проекта: практико-ориентированный.

Основная часть

1.1. История языка программирования Python

История языка программирования Python началась в конце 1980-х. Гвидо ван Россум задумал Python в 1980-х годах, а приступил к его созданию в декабре 1989 года в центре математики и информатики в Нидерландах. Язык Python был задуман как потомок языка программирования ABC, способный к обработке исключений и взаимодействию с операционной системой Амёба. Ван Россум является основным автором Python и продолжал выполнять центральную роль в принятии решений относительно развития языка вплоть до 12 июля 2018 года.

Название языка произошло вовсе не от названия семейства пресмыкающихся. Автор назвал язык в честь популярного британского комедийного телешоу 1970-х «Летающий цирк Монти Пайтона».

Впрочем, всё равно название языка чаще связывают именно со змеей, нежели с передачей – даже эмблема на сайте `python.org` (до выхода версии 2.5) изображают змеиные головы. Важная цель разработчиков Python — создавать его забавным для использования. Это отражено в его названии, которое пришло из Монти Пайтона.

1.2. Основы языка программирования Python

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций. Язык обладает чётким и последовательным синтаксисом, продуманной модульностью и масштабируемостью, благодаря чему исходный код написанных на Python программ легко читаем.

При запуске Python появляется окно интерпретатора Python Shell. Оно переводит понятный человеку код в машинный язык, то есть в код, который может выполнить процессор устройства. Здесь находятся вкладки «Файл»,

«Редактировать», «Отлаживать», «Опции», «Окно», «Помощь». Для создания программ необходимо зайти во вкладку «Файл» и создать новый файл. Перед нами откроется окно, в котором мы будем писать код. После написания программы она будет исполняться в Python Shell.

Для создания программ часто необходимы дополнительные функции. Для этого существуют специальные библиотеки. Библиотеки могут использоваться для создания оконных приложений с кнопками, картинками и так далее. Существуют специальные библиотеки для создания игр. Некоторые из них встроены в Python, некоторые нужно скачивать отдельно.

Основными функциями в Python являются:

- Print(‘’) – функция для печати аргументов, указанных в скобках
- Input(‘’) – функция для ввода данных с клавиатуры
- Int() – осуществляет переход к целочисленному типу данных
- str() - осуществляет переход к строковому типу данных
- range() – создает последовательность чисел в указанном диапазоне
- len() – функция для получения длины строки

Также необходимыми элементами при составлении кода являются циклы, которые позволяют сделать из сложных задач более простые. С помощью них можно настроить поток программы так, чтобы код повторялся определенное количество раз. Иными словами, не придется постоянно прописывать один и тот же код.

Основными циклами в Python являются:

- While (пока) - используется для неоднократного исполнения определённой инструкции до тех пор, пока заданное условие остаётся истинным.
- For(для) - позволяет выполнять повторяющиеся действия для каждого элемента в коллекции

Кроме того, для того, чтобы другой человек смог более детально разобраться в коде, используют комментарии. Комментарии — поясняющие

строки в программном коде, которые позволяют понять смысл написанного. Они пишутся для людей, но игнорируются компиляторами и интерпретаторами.

Компилятор Python – это инструмент, который преобразует исходный код на языке Python в машинный код, позволяя запускать программы на компьютере без необходимости наличия интерпретатора Python.

Интерпретатор – это специальная программа распознающая и исполняющая команды. Благодаря интерпретатору некоторые операторы по-разному работают с разными типами данных.

- Оператор + : значение для строк – объединение их, значение для чисел – их сумма.
- Оператор * : значение для строк – повторение строки несколько раз, значение для чисел – их произведение.

Ключевым элементом программирования в Python являются условные операторы, позволяющие коду принимать различные решения в зависимости от определенных условий. Для программирования условного оператора используются команды if(если) и else(иначе).

Их суть заключается в следующем: когда выражение после if оценивается как истинное (True), нужно выполнить следующий за ним код. Но если оно оценивается как ложное (False), нужно выполнить код, следующий за оператором else». Оператор else записывается на новой строке после последней строки кода с отступом и не может быть записан сам по себе. Оператор else является частью оператора if.

1.3. Плюсы и минусы языка программирования Python

Плюсы Python: Python является интерпретируемым языком программирования, который не компилируется. Таким образом, до запуска он представляет собой обычный текстовый файл. Соответственно, программировать можно почти на всех платформах, а сам язык логичен и хорошо спроектирован.

Кода в нём меньше, чем при использовании других языков программирования, поэтому разработка осуществляется быстрее. В том числе поэтому и считается, что Python хорошо подходит для новичков.

Минусы Python: Несмотря на множество плюсов (простота, лаконичность, универсальность), у языка есть и минусы:

1. Программы на Python считают недостаточно быстрыми. Для сравнения: софт для iOS, написанный на языке Swift, может работать в 8 раз быстрее, чем на Python. Также Python — не очень подходящее решение для задач, которые требуют большого объёма памяти, — такие задачи лучше решать с помощью C либо C++.

2. Высокий уровень зависимости от системных библиотек. В результате затрудняется перенос на другие системы.

Практическая часть

2.1. Установка утилиты для языка Python

Для создания программ и написания кода нам в первую очередь нужно установить на свой ПК такую программу как “PyCharm”. Именно эта программа нам отлично подойдёт для написания кода, так как в ней очень удобно работать, а также в ней работают практически все программисты, которые работают с языком программирования “Python”.

2.2. Создание файла для работы с Python

Написание кода начинается с создания файла, в котором вы будете работать. Для её создания нам придётся запустить программу PyCharm, затем нажать в верхнем левом углу на File, после чего нам нужно создать папку в которой мы будем работать, нажимаем на New Project. Когда у нас создалась папка нажимаем по ней правой кнопкой мыши, далее нажимаем на New и выбираем пункт Python File. Готово, ваш файл создан!

2.3. Разработка игры

Создание программы игры я начала прежде всего с того, что определилась каков будет сюжет игры и её оформление. Уверена, что каждый когда-нибудь играл в игру, сюжет которой заключается в том, что Игрок контролирует небольшую платформу-ракетку, которую можно передвигать горизонтально от одной стенки до другой, подставляя её под шарик, предотвращая его падение вниз. Удар шарика по кирпичу приводит к разрушению кирпича. Множество людей помнят эту игру из далекого детства, однако не многие задумываются о том, как она устроена, поэтому я взяла для разработки именно её.

При создании игры используются такие библиотеки как:

- tkinter, эта библиотека предустановлена на большинстве версиях Python и используется для создания самого оконного приложения.
- time, в нашем случае будет использоваться, что бы задать скорость мячу и платформе. Для установки нужно зайти в командную строку от имени

администратора и написать: `pip install time`, проверив, что `pip` на устройстве установлен.

- `random`, в нашем случае будет использоваться , что бы мяч отскакивал в разных направлениях. Эта библиотека так же предустановлена на всех версиях Python.

- `pygame`, используется для создания графического интерфейса игры. Эту библиотеку нужно скачать через командную строку, прописав: `pip install pygame`

Далее пишется сам код, который я представлю в приложении 1.

Заключение

Язык программирования Python - функциональный и понятный даже для детей. С помощью Python можно проводить весело время и заниматься серьёзными делами.

На основании всего вышеизложенного можно сделать вывод, что Python – активно развивающийся и самый перспективный язык программирования, который поддерживает почти все операционные системы. Он простой и удобный в использовании.

В процессе выполнения данного проекта я:

- изучила язык программирования Python;
- поняла сущность и выяснила историю языка Python;
- выяснила плюсы и минусы Python;
- создала свою игру, используя Python

Окончив свой проект, я могу сказать, что все из того, что было мной задумано, получилось. Я достигла своей цели, а также выполнила все задачи.

Список литературы

1. Рик Гаско «Простой Python. Просто с нуля» - 2023. – 256 с
2. Майк МакГрат: «Python. Программирование для начинающих» - 2023. – 192 с
3. Кольцов Д. М. : «Python. Полное руководство» - 2022. – 480 с
4. Википедия: «<https://ru.wikipedia.org/wiki/Python>»
5. Helloworld:
«<http://www.helloworld.ru/texts/comp/lang/python/python2/index.htm>»

```

    from tkinter import *
import time
import random
import pygame

class Ball():

    def __init__(self, canvas, platform, color):
        self.canvas = canvas
        self.platform = platform
        self.oval = canvas.create_oval(200, 200, 215, 215, fill=color)
        self.dir = [-3, -2, -1, 1, 2, 3]
        self.x = random.choice(self.dir)
        self.y = -1
        self.touch_bottom = False

    def touch_platform(self, ball_pos):
        platform_pos = self.canvas.coords(self.platform.rect)
        if ball_pos[2] >= platform_pos[0] and ball_pos[0] <= platform_pos[2]:
            if ball_pos[3] >= platform_pos[1] and ball_pos[3] <= platform_pos[3]:
                return True
        return False

    def draw(self):
        self.canvas.move(self.oval, self.x, self.y)
        pos = self.canvas.coords(self.oval)
        if pos[1] <= 0:
            self.y = 3
        if pos[3] >= 400:
            self.touch_bottom = True
        if self.touch_platform(pos) == True:
            self.y = -3
        if pos[0] <= 0:
            self.x = 3
        if pos[2] >= 500:
            self.x = -3

class Platform():

    def __init__(self, canvas, color):

```

```
self.canvas = canvas
self.rect = canvas.create_rectangle(230, 300, 330, 310, fill=color)
self.x = 0
self.canvas.bind_all('<KeyPress-Left>', self.left)
self.canvas.bind_all('<KeyPress-Right>', self.right)
```

```
def left(self, event):
    self.x = -2
```

```
def right(self, event):
    self.x = 2
```

```
def draw(self):
    self.canvas.move(self.rect, self.x, 0)
    pos=self.canvas.coords(self.rect)
    if pos[0] <= 0:
        self.x = 0
    if pos[2] >= 500:
        self.x = 0
```

```
window = Tk()
window.title("Аркада")
window.resizable(0, 0)
window.wm_attributes("-topmost", 1)
```

```
canvas = Canvas(window, width=500, height=400)
canvas.pack()
```

```
platform = Platform(canvas, 'green')
ball = Ball(canvas, platform, 'red')
```

```
while True:
    if ball.touch_bottom == False:
        ball.draw()
        platform.draw()
    else:
        break
```

```
window.update()
time.sleep(0.01)
```

`window.mainloop()`