

УДК 372.800.2

**С. В. Журавлев**

**S. V. Zhuravlev**

Журавлев Сергей Владимирович, преподаватель, КГПИ  
ФГБОУ ВО «КемГУ», г. Новокузнецк, Россия.

Zhuravlev Sergei Vladimirovich, lecturer, Kuzbass  
Humanitarian Pedagogical Institute of Kemerovo State  
University, Novokuznetsk, Russia.

## **ИСПОЛЬЗОВАНИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ПРИ ИЗУЧЕНИИ ТЕМЫ «ГРАФИКА» В ПРОЦЕССЕ ОБУЧЕНИЯ ШКОЛЬНИКОВ ПРОГРАММИРОВАНИЮ**

## **THE USE OF ARTIFICIAL INTELLIGENCE IN STUDYING THE TOPIC «GRAPHICS» IN THE PROCESS OF TEACHING PROGRAMMING TO PUPILS**

**Аннотация.** В статье описываются особенности применения нейронных сетей при обучении школьников программированию на примере Chat GPT. Приводится пример изучения темы «Графика».

**Annotation.** The article describes the features of the introduction of neural network in teaching programming to pupils on the example of Chat GPT. An example of studying the topic «Graphics» is given.

**Ключевые слова:** *искусственный интеллект, нейронная сеть, обучение школьников программированию, генерация кода программы, графика.*

**Keywords:** *artificial intelligence, neural network, teaching programming to pupils, program code generation, graphics.*

В настоящее время продолжается активное внедрение искусственного интеллекта в сферу образования с целью повышения качества обучения. Также продолжает развиваться и проектное обучение.

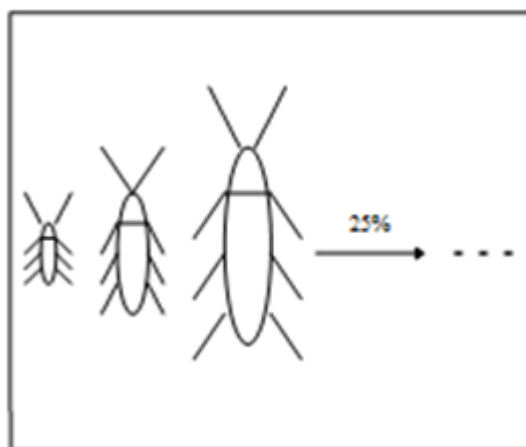
В предыдущей статье мы уже рассматривали применение нейронных сетей при обучении школьников программированию в среде Lazarus на примере Replit [1]. В этой статье речь пойдет о программировании на Python и другой нейросети, которая способна решать более сложные задачи, не только сформулированные в виде простого текста, но и представленные в других форматах, в том числе из загруженных файлов.

Chat GPT – искусственный интеллект, созданный компанией OpenAI. Он работает на базе языковых моделей серии Generative Pre-trained Transformer, способных понимать человеческий язык, анализировать информацию, а после выдавать осмысленные ответы. Изначально нейросеть справлялась лишь с простыми задачами, но на сегодняшний день она может не только ответить на вопрос, но и предсказать следующие шаги пользователя, предложить идеи, подсказать, как решить сложную задачу поэтапно [3]. В частности, Chat GPT умеет переводить с языка на язык, генерировать компьютерные программы, SQL-запросы и сложные формулы для Excel и даже играть в шахматы [4].

Работать с Chat GPT удобнее всего в платформе с одноимённым названием версии 4.0 или выше. Эта платформа удобна тем, что её не нужно скачивать, с ней можно работать на любом устройстве с подключением к сети Интернет после регистрации. Необходимо учитывать, что уровень бесплатного пользования имеет ограничения на ресурсы: допускается не более 30 запросов в виде промптов и 1 запрос в виде файла (изображение, видео и т. п.) в сутки. Кроме того, в бесплатной версии нейросеть после ответа «забывает» его, так же как и вопрос, поэтому при необходимости уточнить какой-либо момент приходится заново полностью формулировать условие задачи. Чтобы запустить чат для подачи промпта, нужно нажать на кнопку «Войти» в правой верхней части экрана и войти в свой аккаунт.

Мы рассматриваем возможности использования нейронных сетей при изучении темы «Графика», которая содержит наиболее сложные задачи. Перед нами стояла задача проверить, сможет ли Chat GPT понять задачу, заданную в виде рисунка, и сгенерировать код для её решения.

В качестве примера мы сформулировали для нейронной сети запрос в виде файла, содержащего рисунок с пояснением вида: «Напиши программу решения задачи на языке Python с пошаговым объяснением. Вывести на экран следующую картинку (необходимо использовать цикл):  $n$  жуков, каждый из которых на 25 % больше предыдущего (рис. 1) [2]. Для рисования использовать модуль `tkinter`».



## Рисунок 1. Условие задачи

Следующая часть текста содержит ответ нейросети.

*Давайте пошагово напишем программу, рисующую жуков в стиле как на картинке, каждый следующий на 25% больше предыдущего, с помощью `tkinter`.*

*ШАГ 1: Импорт и инициализация окна.*

```
import tkinter as tk  
  
root = tk.Tk()  
  
canvas = tk.Canvas(root, width=400, height=250, bg='white')  
  
canvas.pack()
```

*ШАГ 2: Функция рисования одного жука. Будем рисовать жука в виде овала с линиями-лапками, усиками и т.д. Функция будет принимать координаты центра и масштаб.*

```
def draw_beetle(canvas, x, y, scale):  
  
    body_len = 40 * scale  
  
    body_wid = 16 * scale  
  
    head_r = 6 * scale  
  
    # Тело  
  
    canvas.create_oval(x - body_wid, y - body_len/2, x +  
body_wid, y + body_len/2)  
  
    # Голова  
  
    canvas.create_oval(x - head_r, y - body_len/2 - head_r*2, x +  
head_r, y - body_len/2)  
  
    # Лапки (3 слева, 3 справа)
```

```
offset = body_len * 0.2
```

```
for i in range(-1, 2):
```

```
    y_leg = y - offset*i
```

```
    # слева
```

```
        canvas.create_line(x - body_wid, y_leg, x - body_wid -  
18*scale, y_leg - 12*scale)
```

```
    # справа
```

```
        canvas.create_line(x + body_wid, y_leg, x + body_wid +  
18*scale, y_leg - 12*scale)
```

```
    # Усики
```

```
        canvas.create_line(x, y - body_len/2 - head_r*2, x - 12*scale, y  
- body_len/2 - head_r*3)
```

```
        canvas.create_line(x, y - body_len/2 - head_r*2, x + 12*scale,  
y - body_len/2 - head_r*3)
```

*ШАГ 3: Цикл по жукам. Объявим количество жуков `n`,  
первый размер `scale = 1`, и будем увеличивать каждый  
следующий на 25%.*

```
n = 3
```

```
x0 = 40 # стартовая позиция x
```

```
y0 = 130
```

```
scale = 0.5
```

```
dx = 60
```

```
for i in range(n):
```

```
    draw_beetle(canvas, x0 + i*dx*scale, y0, scale)
```

```
    scale *= 1.25
```

$dx *= 1.25$  # расстояние между жуками тоже увеличивается

ШАГ 4: Надписи и стрелка. Добавим текст и стрелку как на картинке:

```
canvas.create_line(200, 130, 270, 130, arrow=tk.LAST)
```

```
canvas.create_text(235, 120, text="25%")
```

```
canvas.create_line(270, 130, 320, 130, dash=(3,2))
```

ШАГ 5: Запуск окна.

```
root.mainloop()
```

Если нужно другое количество жуков — просто поменяйте значение `n`!

Запустив программу, получаем результат (рис. 2).

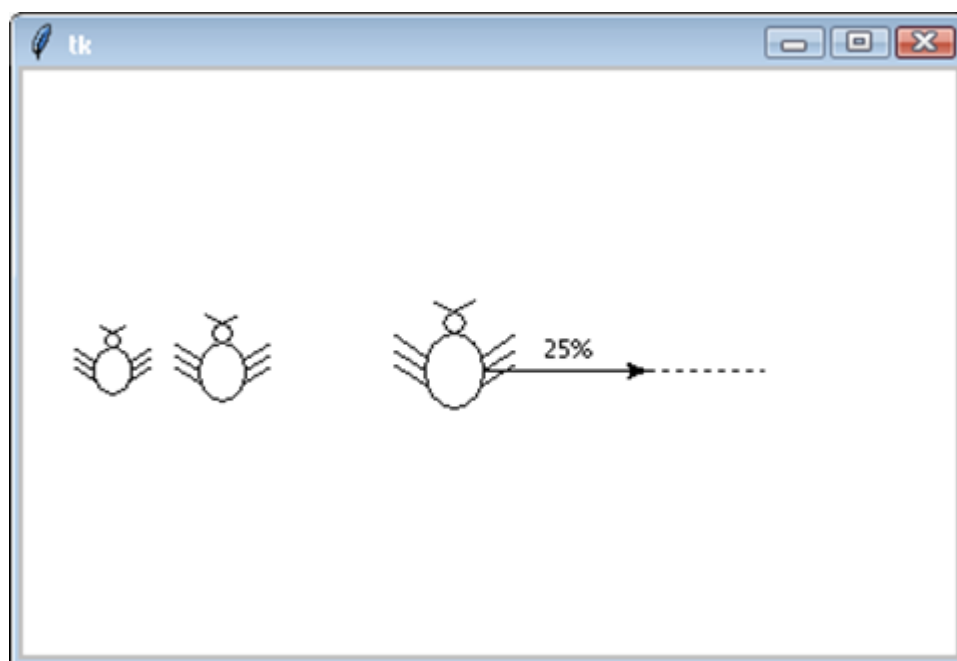


Рисунок 2. Результат работы программы

Как видим, жуки получились не совсем такие, как в задании, но это не так важно, функцию рисования жука легко можно изменить. Основная проблема состоит в том, что помимо жуков рисунок содержит также цифры, стрелку и пунктирную линию, которых не должно быть, а жуки не находятся на одинаковом расстоянии друг от друга. Приходим к выводу, что нужно переформулировать пояснение к рисунку следующим образом: «Напиши программу решения задачи на языке Python с пошаговым объяснением. Вывести на экран следующую картинку (необходимо использовать цикл):  $n$  жуков, каждый из которых на 25 % больше предыдущего. Текст, пунктирную линию и стрелку рисовать не нужно, а жуки должны располагаться по всей горизонтали на одинаковом расстоянии друг от друга. Для рисования использовать модуль `tkinter`».

Следующая часть текста содержит ответ нейросети.

*Давайте решим задачу шаг за шагом.*

*Алгоритм решения.*

1. *Импортируем модуль `tkinter`.*
2. *Создадим окно и холст (`Canvas`), где будут рисоваться жуки.*
3. *Опишем функцию рисования одного жука с параметрами масштаба и координатами центра.*
4. *Запросим у пользователя количество жуков `n`.*
5. *Рассчитаем параметры для размещения жуков по всей ширине окна, чтобы промежутки между ними были одинаковы.*
6. *В цикле нарисуем  $n$  жуков, каждый больше предыдущего на 25 %.*

```
import tkinter as tk
```

```
def draw_beetle(canvas, x, y, scale):
```

```
    body_len = 40 * scale
```

```
    body_wid = 16 * scale
```

```
head_r = 6 * scale
```

```
# Телосcanvas.create_oval(x - body_wid, y - body_len/2, x +  
body_wid, y + body_len/2)
```

```
# Голова
```

```
canvas.create_oval(x - head_r, y - body_len/2 - head_r*2, x +  
head_r, y - body_len/2)
```

```
# Лапки (3 слева, 3 справа)
```

```
offset = body_len * 0.2
```

```
for i in range(-1, 2):
```

```
    y_leg = y - offset*i
```

```
    # слева
```

```
    canvas.create_line(x - body_wid, y_leg, x - body_wid -  
18*scale, y_leg - 12*scale)
```

```
    # справа
```

```
    canvas.create_line(x + body_wid, y_leg, x + body_wid +  
18*scale, y_leg - 12*scale)
```

```
# Усики
```

```
canvas.create_line(x, y - body_len/2 - head_r*2, x - 12*scale, y  
- body_len/2 - head_r*3)
```

```
canvas.create_line(x, y - body_len/2 - head_r*2, x + 12*scale,  
y - body_len/2 - head_r*3)
```

```
# -----
```

```
# Параметры окна и задача
```

```
n = 5  # Количество жуков
```

```
canvas_w = 800
```

```
canvas_h = 400
```

```
min_size = 0.5 # Размер самого маленького жука
scale_factor = 1.25 # Каждый следующий на 25% больше

# Окно и холст
root = tk.Tk()

canvas = tk.Canvas(root, width=canvas_w, height=canvas_h,
bg="white")

canvas.pack()

# Расчёт расстояния между жуками
space = canvas_w // (n + 1)

center_y = canvas_h // 2

# Размер первого жука
size = min_size

# Цикл рисования жуков
for i in range(n):
    x = space * (i + 1)

    draw_beetle(canvas, x, center_y, size)

    size *= scale_factor

root.mainloop()
```

#### *Пояснения:*

- *draw\_beetle* – функция для рисования жука с учетом размера;
- количество жуков *n* и размер первого задаются в начале;
- между жуками одинаковое расстояние по горизонтали (*space*);
- каждый жук больше предыдущего в *scale\_factor* (1.25) раз;

- *для рисования простого жука используются линии и овалы на `Canvas`. Если хотите сделать красивее или добавить деталей, усложните функцию рисования жука!*

Запустив программу, убеждаемся, что на этот раз результат соответствует условию задачи (рис. 3).

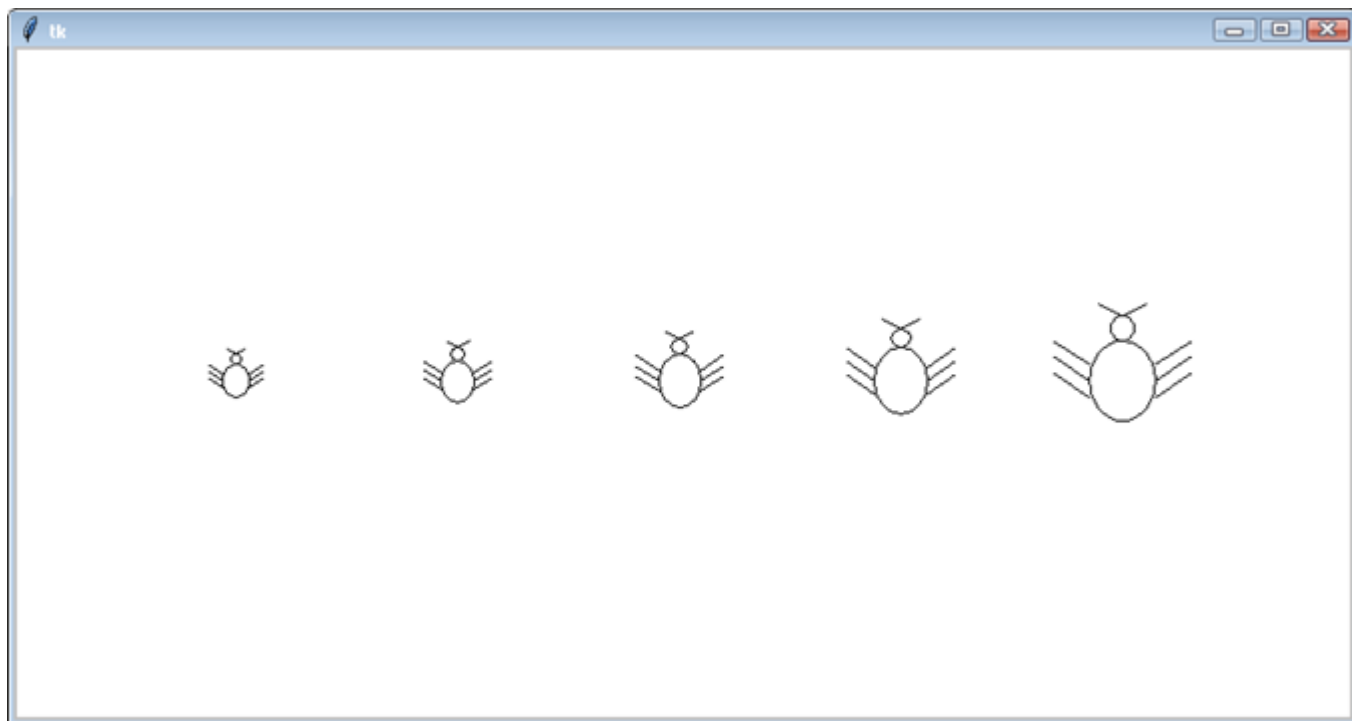


Рисунок 3. Результат работы программы после уточнения условия задачи

Итак, мы убедились, что нейронная сеть Chat GPT способна генерировать код программы, решающей графическую задачу по заданному рисунку, а значит, может быть полезна при изучении темы «Графика» в процессе обучения школьников программированию. Однако также мы приходим к выводу, что обучающиеся, для того чтобы получить результат, максимально соответствующий заданию, должны прописывать пояснение к рисунку с мельчайшими подробностями, чётко указывающими, что должно быть нарисовано, а что нет. Ну и, конечно, после получения результата школьникам необходимо внимательно анализировать код и при необходимости вносить изменения в него. Если возможностей бесплатной версии Chat GPT окажется недостаточно, можно будет оформить платную подписку.

## Список литературы

1. Журавлёв, С. В. Использование нейронных сетей при обучении школьников программированию / С. В. Журавлёв. – Текст: электронный // Информационно-коммуникационные технологии в педагогическом образовании: электронный научный журнал. – 2024. – № 06 (93) июль. – URL: <http://infed.ru/articles/10368/> (дата обращения: 19.05.2025).
2. Можаров, М. С. Введение в структурное программирование: Учебное пособие / М. С. Можаров, Г. Н. Бойченко. – 2-е изд., стереот. – Новокузнецк: Изд-во КузГПА, 2014. – 203 с. – Текст: непосредственный.
3. Полный гайд по ChatGPT: как использовать нейросети на максимум. / Я – зерокодер : [сайт]. – URL: <https://ya.zerocoder.ru/polnyj-gajd-po-chatgpt-kak-ispolzovat-nejroseti-na-maksimum/> (дата обращения: 19.05.2025). – Текст: электронный.
4. Цуриков, А. Н. ChatGPT – что это такое и как устроена нейросеть: возможности и примеры использования / А. Н. Цуриков. – Текст: электронный. – URL: <https://skillbox.ru/media/code/chatgpt-o-chyem-my-pogovorili-s-velikim-pritvorshchi-kom-ot-openai-i-kak-eto-bylo/> (дата обращения: 19.05.2025).

---

© Журавлев С. В., 2025