Муниципальное бюджетное общеобразовательное учреждение

«Менделеевская средняя общеобразовательная школа»

УТВЕРЖДАЮ

Директор МБОУ «МСОШ»

_____ Т.Б. Богданова

Приказ №337/02 от «25» августа 2023 г.

РАБОЧАЯ ПРОГРАММА

на 2023 – 2024 учебный год

К ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ

(ОБЩЕРАЗВИВАЮЩЕЙ) ПРОГРАММЕ

Профессиональная проба

«Профессия – программист»

«Графика в Python при помощи модуля Turtle»

(14 – 17 лет)

Автор-составитель: Нагоева Эльвира Германовна,

Педагог дополнительного образования

п. Менделеево, 2023

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Актуальность профессиональной пробы связана с проблемой подготовки обучающегося к жизненному и профессиональному самоопределению в современных социально-экономических условиях. Складывающийся в России рынок труда высветил серьезные затруднения в ее решении.

Профессиональная проба является средством актуализации профессионального самоопределения и активизации творческого потенциала личности школьников. Такой подход ориентирован на расширение границ возможностей традиционного трудового обучения в приобретении учащимися опыта профессиональной деятельности.

В этой связи особенностями профессиональной пробы являются:

1. Диагностический характер пробы, т. е. на каждом этапе профессиональной пробы осуществляется диагностика общих и специальных профессионально важных качеств.
2. Результатом каждого этапа и итога профессиональной пробы является получение завершенного продукта деятельности — изделия, узла, выполнение функциональных обязанностей профессионала.
3. Процесс выполнения пробы направлен на формирование у школьников целостного представления о конкретной профессии, группе родственных профессий, сферы, их включающей.
4. Развивающий характер профессиональной пробы, направленный на интересы, склонности, способности личности школьника, достигаемый за счет постепенного усложнения выполнения практических заданий профессиональной пробы в соответствии с уровнем подготовленности школьников к ее выполнению, внесения в содержание пробы элементов творчества и самостоятельности.
5. Профессиональная проба выступает как системообразующий фактор формирования готовности школьников к выбору профессии. Она интегрирует знания школьника о мире профессии данной сферы, психологических особенностях деятельности профессионала и практическую проверку собственных индивидуально-психологических качеств, отношения к сфере профессиональной деятельности.

Профессиональная область. Профессия относится к типу профессий «человек – техническая система».

Родственными профессиями являются программирование, информационно-коммуникационные технологии, компьютерные сети. Следовательно, профессиональная проба поможет определить именно ту профессию, которая соответствует способностям обучающихся, их возможностям и интересам.

Цель профессиональной пробы – оказание помощи учащимся в профессиональном самоопределении, приобретение опыта в области информационно-коммуникационных технологий

Задачи профессиональной пробы:

- выявление склонностей и способностей обучающихся к выполнению работ, связанных с обработкой цифровой информации и оформлением документации с использованием электронно-вычислительной техники;
- характеристика специальности «Программирование в компьютерных системах»
- содействие профессиональному самоопределению обучающихся.

Образовательная область	Информатика
Наименование профессионального направления /отрасль	Прикладная информатика, информационные технологии
Профессия	Программист
Профессиональные компетенции	Область «Информатика» - способен к выполнению работ, связанных с обработкой цифровой информации и оформлением документации с использованием электронно-вычислительной техники
Уровень сложности профессиональной пробы	базовый
Продолжительность:	4 ч
Темы пробы	1. Специфика и содержание профессиональной деятельности техника -

	программиста 2. Выбор языка программирования, обоснование. Особенности работы в модуле **turtle**. 3. Прямое программирование в среде программирования Python. Часть 1. 4. Прямое программирование в среде программирования Python Часть 2. Презентация программных продуктов.
Возрастная категория	14-17лет

Результатом участия в профессиональной пробе будет самостоятельное создание каждым учащимся программного решения.

СОДЕРЖАНИЕ ПРОГРАММЫ

Постановка задачи (5 мин / 10 мин)

1. *Постановка цели и задачи в рамках пробы.* Необходимо создать графическое приложение, соответствующее приведенному макету. Для создания приложения будем использовать одну из самых популярных сред разработки PyCharm.

2. *Демонстрация итогового результата, продукта.* Ваше приложение в итоге должно успешно запускается, давать ожидаемые результаты и иметь внешний вид, соответствующий макету с незначительными отклонениями (цвета, размер и расположение элементов, шрифты)

Содержание профессиональной пробы

1. Содержательная часть (1 час)

Программист — это специалист, создающий исходный код для программы. Такой программой может быть операционная система компьютера, видеоигра, web или мобильное приложение и даже алгоритм работы микроволновки. Программный код пишется на специальном языке программирования. Он состоит из обычных слов и некоторых специальных символов. Сегодня насчитывается несколько сотен языков программирования, но самые распространенные из них –Java (джава), Python (питон), C#(сишарп), JavaScript, C++, Swift. Какой язык программирования выбрать, программист решает сам в зависимости от конкретной задачи (сделать игру, приложение для web или программу для сервера) и собственных знаний. Квалифицированный программист уверенно использует 2-4 языка.

Специализация программистов

Если вы думаете, что программист = компьютерщик, то это далеко не так. Самое удивительное, что программист не обязательно работает на компьютере. Он может писать код программы хоть на салфетке, а компьютер ему нужен, чтобы применить этот код и протестировать. Не стоит также приравнивать программиста к "айтишнику". Под общим названием "IT-специалист" скрывается более 50 разных профессий и должностей.

Например, UIX-дизайнер (разработчик интерфейсов),

account-менеджер (специалист техподдержки),

системный администратор,

devops-инженер ("инженер по автоматизации IT-процессов"),

менеджер IT-проекта – это "айтишники", но не программисты.

Однако стоит помнить, что в любую из IT-профессий можно попасть через образование программиста.

https://www.youtube.com/watch?v=WN43apU5Q4s (продолжительность видео 14.30)
По специализации программисты делятся на системных, прикладных и веб-программистов. Прикладные программисты занимаются написанием различных программ и приложений: игры (например, популярная игра Angry birds), офисные приложения (например, Microsoft Office Word) и многое другое. Системные программисты занимаются созданием операционных систем (например, Android или iOS). Веб-программисты разрабатывают интернет-сайты и программы, которые поддерживают работу этих сайтов.
На сайтах вакансий вы найдете разделение программистов по тем языкам, с которыми они работают: программист C++, программист джава, PHP-программист. А также по типам задач, которые они выполняют: программист баз данных, фронтенд / бекенд разработчик, мобильный / десктоп разработчик, программист 1С, программист Битрикс, геймдев, архитектор ПО, системный инженер и т.д.

Плюсы профессии программист:

? Высокая зарплата. Средняя зарплата высококвалифицированного программиста 70-100 тыс. рублей.

Востребованность. Рынок труда для программистов растет за счет цифровизации всех сфер общественной жизни. Онлайн-образование, телемедицина, интернет вещей (IoT), даже сельское хозяйство не обходится теперь без IT-специалистов.

? Интересные задачи и простор для творчества. Что делает программист на работе? Пишет код, тестирует, проверяет на уязвимости, переписывает код. Тем не менее, конечный результат кодирования – это рабочая программа с разными целями. Одни программисты создают игры. Другие программируют роботов. Третьи делают полезное приложение под Android. Четвертые создают агрегатор, который соберет все билеты мировых авиакомпаний в один сервис для бронирования авиабилетов. Разнообразно, не правда ли? А когда продукт начинает исправно работать, программист испытывает радость, как персонаж папа Карло, который из деревянного полена выстругал живого мальчика.

? Возможность удаленной работы и свободный график. Программисты - это каста людей, к которым не применяют строгих офисных правил. Для поддержания креативности и работоспособности им разрешают работать из дома, приходить позже на работу. И, если вы еще не знаете, лучшие офисы принадлежат именно IT-компаниям. Загляните в Google, Microsoft, Яндекс и Mail.Ru.

Карьерный рост в России и за рубежом. Для программирования нет географических границ. Во-первых, многие российский специалисты работают удаленно в иностранных проектах. Во-вторых, переехать в "Силиконовую" долину (более точно ее называть "Кремниевой") для IT-профессионала с отличным знанием английского гораздо проще, чем, например, бухгалтеру, юристу или инженеру-строителю, которым придется изучать специфику профессии в конкретной стране.

Минусы:

? Сидячий образ жизни. Без физических нагрузок в нерабочее время программист рискует большую часть зарплаты тратить на врачей. Особенно страдают зрение и опорно-двигательный аппарат.

? Быстрое устаревание знаний. Программисту необходимо постоянно учиться. Программирование развивается стремительно, поэтому программист обязательно должен читать профильные журналы и интернет-ресурсы, получать новые знания и сертификаты на курсах. Иначе его ценность как специалиста на рынке труда упадет.

1. Где учиться?

Чтобы стать программистом, можно пойти тремя путями:

1. После 11 класса в вуз по направлению 09.00.00 "Информатика и вычислительная техника" или 10.00.00 "Информационная безопасность".

ПГНИУ

Программы бакалавриата и программы специалитета на 2021/22 уч. год

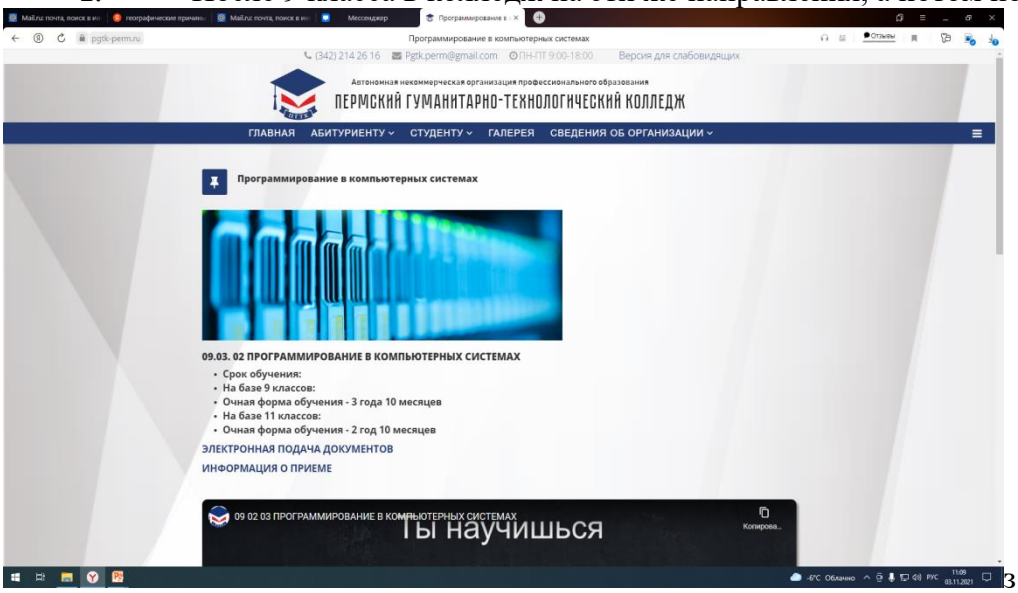
Код	Направление подготовки	Конкурсная группа	Наименование профилей/ специализаций, реализуемых в рамках направления	Вступительные испытания и минимальные баллы, подтверждающие успешное прохождение вступительных испытаний (в порядке приоритетности): *- на базе СОО-ЕГЭ; - на базе ПО-внутренний экзамен ПНИПУ*	Реализуемые формы обучения		
					Очная	Очно-заочная	Заочная
Программы подготовки бакалавриата					Срок обучения		
09.03.01	**Информатика и вычислительная техника**	ИВТ	Автоматизированные системы обработки информации и управления Компьютерные системы	1. Математика (профильный уровень)-39, 2. **Физика-39 или Информатика и ИКТ-44**, 3. Русский язык-40	4 года	нет	5 лет (3,5 года*)
09.03.04	**Программная инженерия**	РИС	Разработка программно-информационных систем	1. Математика (профильный уровень)-39, 2. **Физика-39 или Информатика и ИКТ-44**, 3. Русский язык-40	4 года	нет	5 лет (3,5 года*)

ПНИПУ

Программы бакалавриата на 2021/22 уч. год

Код	Направление подготовки	Конкурсная группа	Наименование профилей/ специализаций, реализуемых в рамках направления	Вступительные испытания и минимальные баллы, подтверждающие успешное прохождение вступительных испытаний (в порядке приоритетности): - на базе СОО-ЕГЭ; - на базе ПО- внутренний экзамен ПНИПУ	Реализуемые формы обучения		
					Очная	Очно-заочная	Заочная
Программы подготовки бакалавриата					Срок обучения		
01.03.02	**Прикладная математика и информатика**	ММ	Математическое моделирование	1. Математика (профильный уровень)-39, 2. **Физика-39**, 3. Русский язык-40	4 года	нет	нет
01.03.02	**Прикладная математика и информатика**	МИЭ	Математическое и информационное обеспечение экономической деятельности	1. Математика (профильный уровень)-39, 2. **Физика-39 или Информатика и ИКТ-44**, 3. Русский язык-40	4 года	нет	нет
09.03.02	**Информационные системы и технологии**	ИСТ	Цифровые технологии и интеллектуальные системы управления	1. Математика (профильный уровень)-39, 2. **Физика-39 или Информатика и ИКТ-44**, 3. Русский язык-40	4 года	нет	5 лет (3,5 года*)

1. После 9 класса в колледж на эти же направления, а потом пойти или не пойти в ву



3.

ГБПОУ "Пермский краевой колледж "ОНИКС" > Специальности > ... > 09.02.07

Информационные системы и программирование

Специальность: 09.02.07 Информационные системы и программирование

Базовое образование: основное общее образование (9 классов)

Срок обучения: 3 года 10 месяцев

Форма обучения: очная

Вступительные испытания: нет

Квалификация выпускников: разработчик веб и мультимедийных приложений

Область профессиональной деятельности, в которой выпускники, освоившие образовательную программу, могут осуществлять профессиональную деятельность:

06 Связь, информационные и коммуникационные технологии

Основные виды деятельности:

Проектирование и разработка информационных систем
Разработка дизайна веб-приложений
Проектирование, разработка и оптимизация веб-приложений

3. Учиться на курсах программирования и самому.

Графика в Python при помощи модуля Turtle (2-4 часы)

Python — это перспективный язык программирования с яркой экосистемой библиотек и фреймворков для решения ряда задач. В этой статье мы сосредоточимся на библиотеке Turtle, которая предоставляет средства для создания графики и анимации. Turtle является одной из самых простых библиотек для начинающих разработчиков, позволяя им сконцентрироваться на основных принципах программирования и создании графики.

Краткий обзор библиотеки Turtle

Библиотека Turtle представляет собой доступный и простой инструмент, который включает в себя функциональные возможности для рисования линий и фигур с помощью командной строки. Данная библиотека была впервые представлена в версии 2.6 языка Python и включается в стандартную библиотеку Python начиная с версии 3.0.

Как и большинство библиотек Python, Turtle выгодно отличается от своих коллег на других языках программирования более простым и чистым синтаксисом, а также гибкостью в использовании. Turtle предоставляет пользователям возможность создавать креативную графику и анимацию, где пользователь может описать все движения карандаша черепахи.

Как начать работу с библиотекой Turtle?

Для выполнения любой работы на Python, необходимо импортировать нужные модули. Для импортирования библиотеки Turtle не требуется устанавливать какие-то дополнительные пакеты.

Писать программе будем в PyCharm.

Импорт черепашки

Как и с любым модулем, для работы с **turtle** её нужно импортировать. Делаем это вот так:

```
1 import turtle
```

Далее создадим объект библиотеки **turtle**, который назван **Turtle**. Значение запишем в переменную **turtlePen**.

```
1 turtlePen = turtle.Turtle()
```

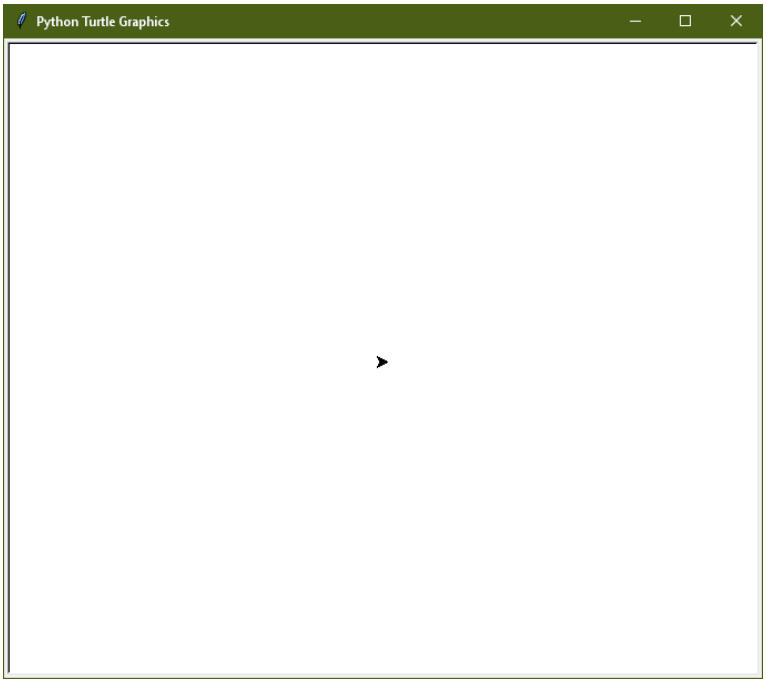
Попробуем запустить? Если все выполнено правильно, то вы должны увидеть белое окно, которое тут же исчезнет, так как программа завершена.

Добавим переменную **window**, которая будет содержать объект **Screen** (Не забывайте о **скобках ()**!). С его помощью можно изменить параметры этого самого окна, вроде цвета фона или возможность сохранить окно даже после всех операций отрисовки.

Последнее нам и нужно, для этого добавляем в самый конец такую команду:

```
1 import turtle
2
3 turtlePen = turtle.Turtle()
4 window = turtle.Screen()
5
6 window.mainloop()   # &lt;- не даст закрыть окно
```

Запускаем и видим:

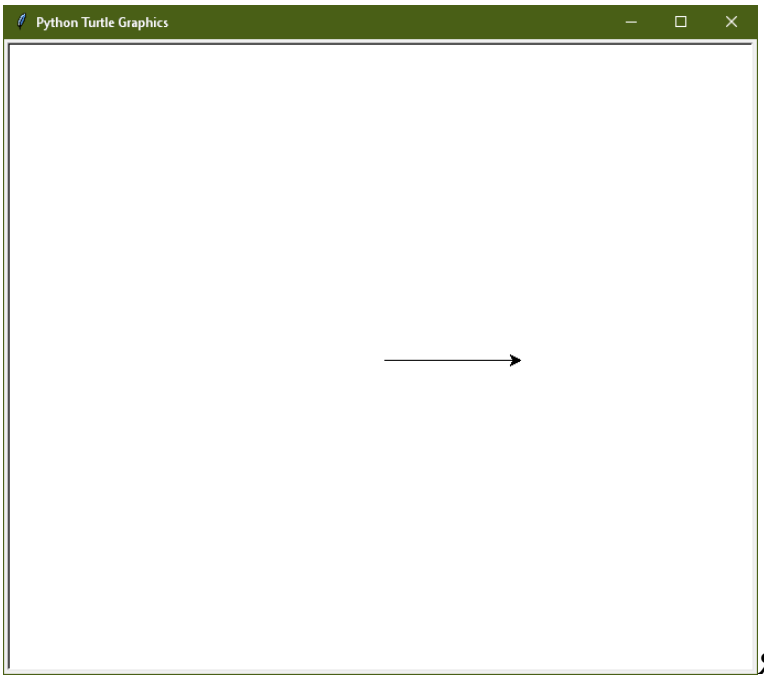


Рисуем при помощи Turtle!

Уже можно рисовать? Да!

Для этого управляем нашим **turtlePen**. Допустим, мы можем указать ему идти вперед на определенное расстояние.

```
turtlePen.forward(123)
```



Я художник!

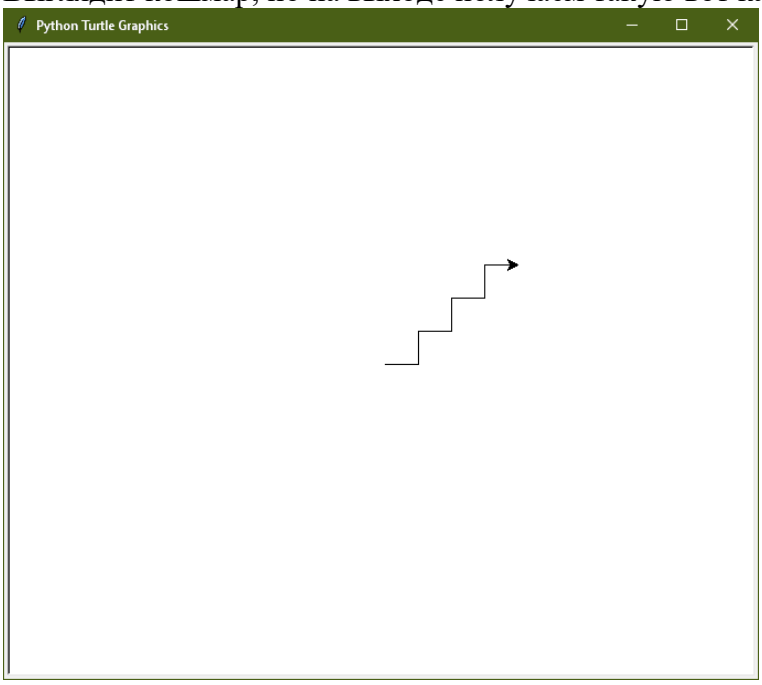
Вперед — это в правую сторону из центра? Любая черепашка начинает с координаты **(0;0)**, т.е. в центре экрана и смотрит в правую сторону.

Чтобы изменить направление движения используем команды **left** или **right**. Для примера нарисуем лесенку при помощи уже известных команд.

```
import turtle

turtlePen = turtle.Turtle()
window = turtle.Screen()

turtlePen.forward(30)
turtlePen.left(90)
turtlePen.forward(30)
turtlePen.right(90)
```

```
10      turtlePen.forward(30)
11      turtlePen.left(90)
12      turtlePen.forward(30)
13      turtlePen.right(90)
14      turtlePen.forward(30)
15      turtlePen.left(90)
16      turtlePen.forward(30)
17      turtlePen.right(90)
18      turtlePen.forward(30)
19
20      window.mainloop()
```

Выглядит кошмар, но на выходе получаем такую вот картинку:

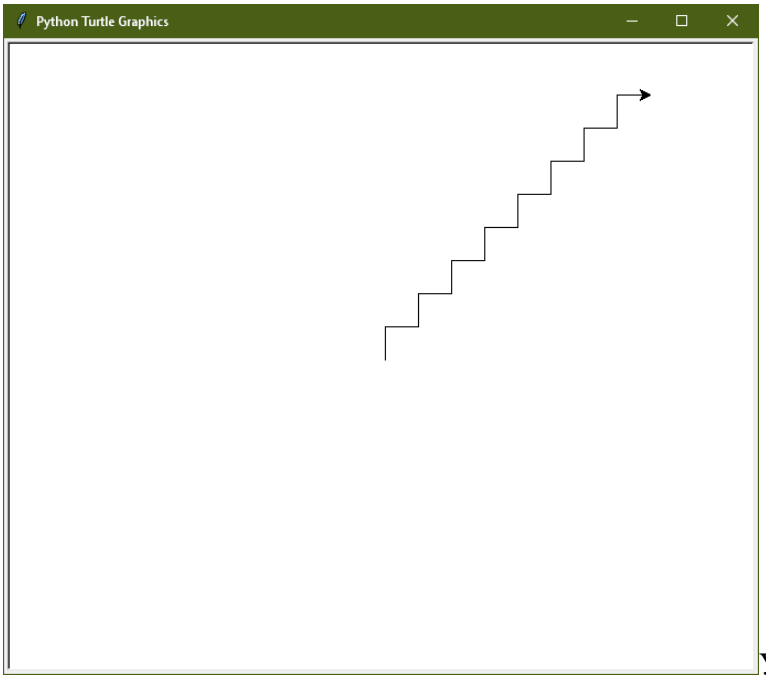


Можно увидеть, что у нас получилось очень много одинаковых команд, а потому для выполнения таких заданий разумно использовать циклы. А потому перепишем этот пример более корректно.

Обсудим скорость движения черепашки. При помощи команды **speed** мы можем изменять скорость рисования. Чем больше значение, тем быстрее рисует, но если укажем 0, то рисунок будет нарисован мгновенно.

Итак, версия ступени версия 2.0:

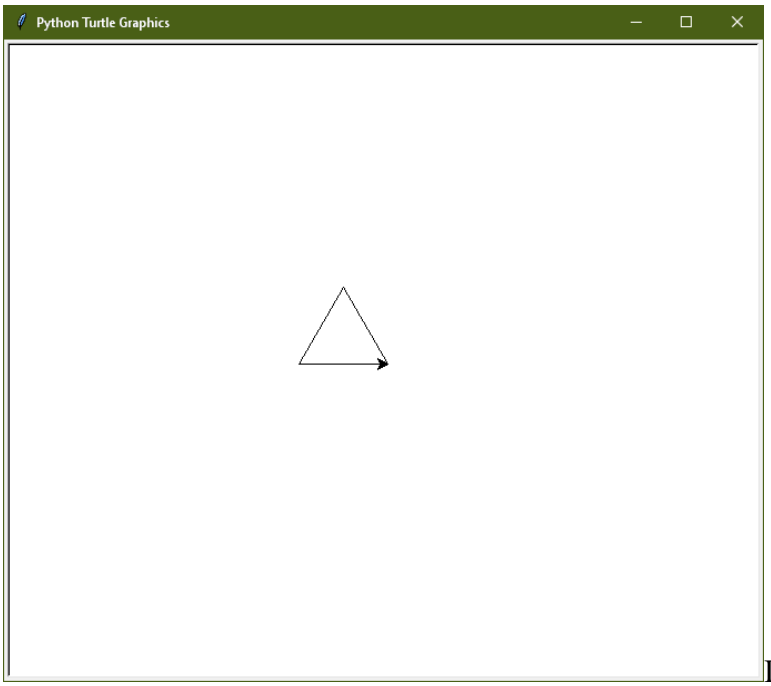
```
1       import turtle
2
3       turtlePen = turtle.Turtle()
4       window = turtle.Screen()
5
6
7       def draw_stairs(n, size=30):     # &lt;- создали функцию, для рисования
8           for i in range(0, n):        # &lt;- цикл нарисует указанное количество ступеней
9               turtlePen.left(90)
10              turtlePen.forward(size)
11              turtlePen.right(90)
12              turtlePen.forward(size)
13
14      turtlePen.speed(10)
15
16      draw_stairs(8)   # &lt;- вызов функции
17
18      window.mainloop()
```



Уже лучше

Что насчет других геометрических фигур? Без проблем! Вот треугольник!

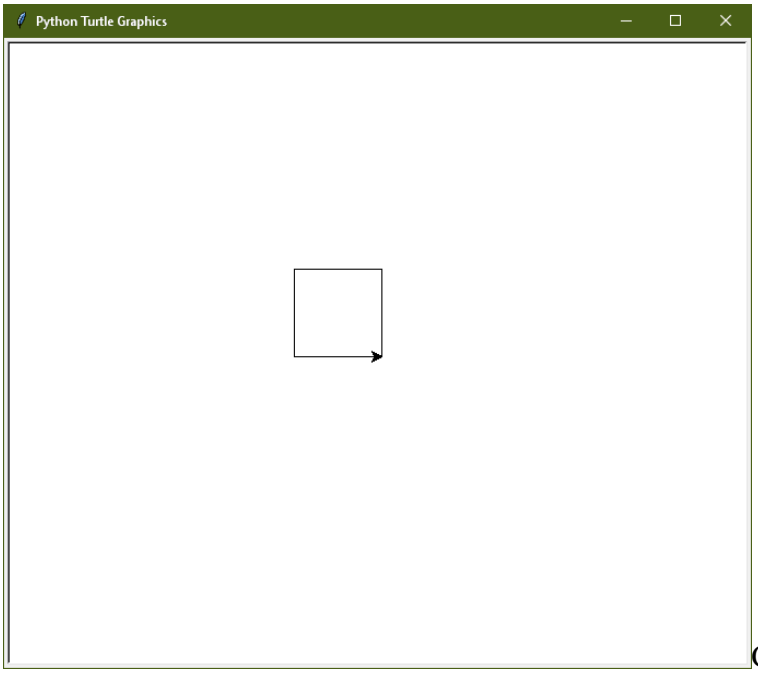
```
turtlePen.left(120)
turtlePen.forward(80)
turtlePen.left(120)
turtlePen.forward(80)
turtlePen.left(120)
turtlePen.forward(80)
```



Похож ведь?

А это квардат:

```
turtlePen.left(90)
turtlePen.forward(80)
turtlePen.left(90)
turtlePen.forward(80)
turtlePen.left(90)
turtlePen.forward(80)
turtlePen.left(90)
turtlePen.forward(80)
```



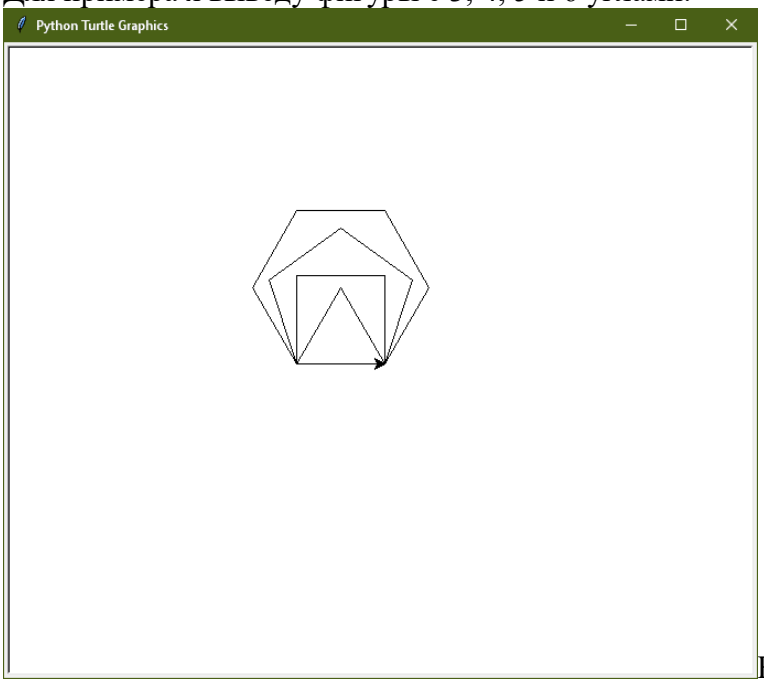
Стороны равны, гарантирую

Внимательные могли заметить, что и для **правильного** треугольника, и для квадрата, мы в сумме получаем поворот на 360 градусов, а значит можно создать функцию рисования правильных многоугольников!

Вот она:

```
def polygon(n, size=80):
    if n > 2:                    # &lt;- многоугольников меньше 3 углов я не знаю :)
        angle = 360/n            # &lt;- получаем угол поворота в зависимости от количества углов

        for n in range(0, n):        # &lt;- рисуем стороны
            turtlePen.left(angle)
            turtlePen.forward(size)
# рисуем разные фигуры
polygon(3)
polygon(4)
polygon(5)
polygon(6)
```

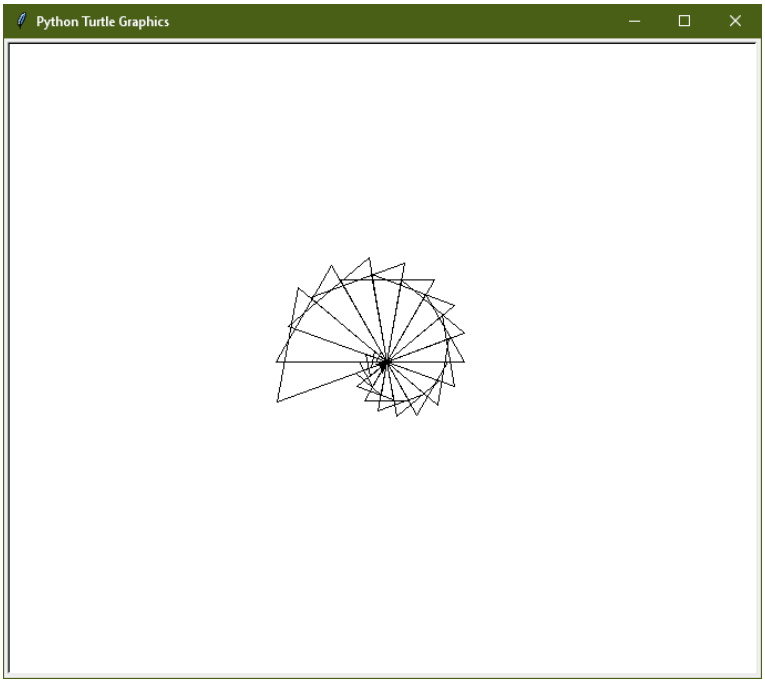
Для примера я выведу фигуры с 3, 4, 5 и 6 углами.



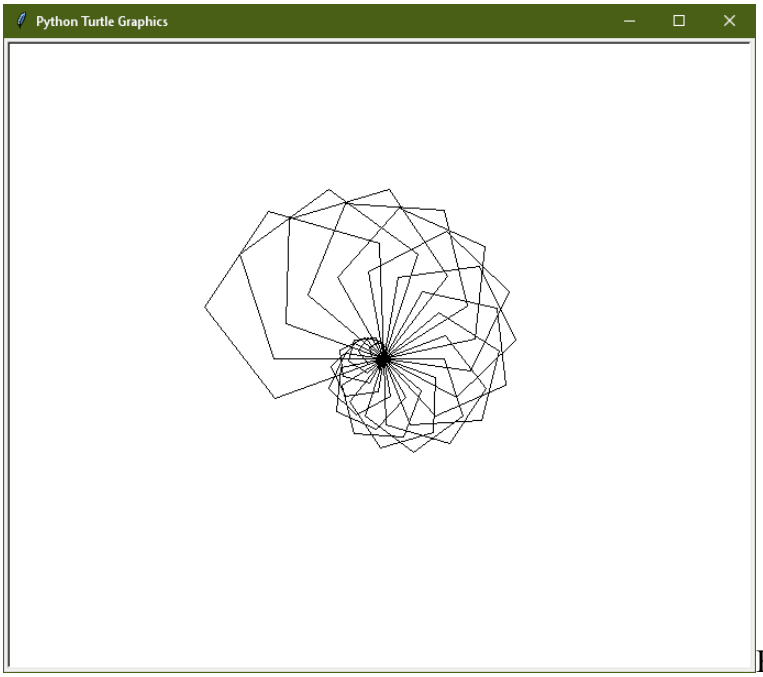
Все работает!

Если заметили, то я везде оставлял возможность менять длину стороны для каждой фигуры (2 параметр у каждой функции). Это вполне можно использовать для создания интересных узоров!
К примеру:

```

import turtle

turtlePen = turtle.Turtle()
window = turtle.Screen()

def polygon(n, size=80):
    if n > 2:
        angle = 360/n

        for n in range(0, n):
            turtlePen.left(angle)
            turtlePen.forward(size)

 turtlePen.speed(10)

for i in range(0, 100, 5):
    polygon(3, 10 + i)
    turtlePen.left(20)

window.mainloop()

```



Один из вариантов…
Изменим треугольники на пятиугольники, и получим:

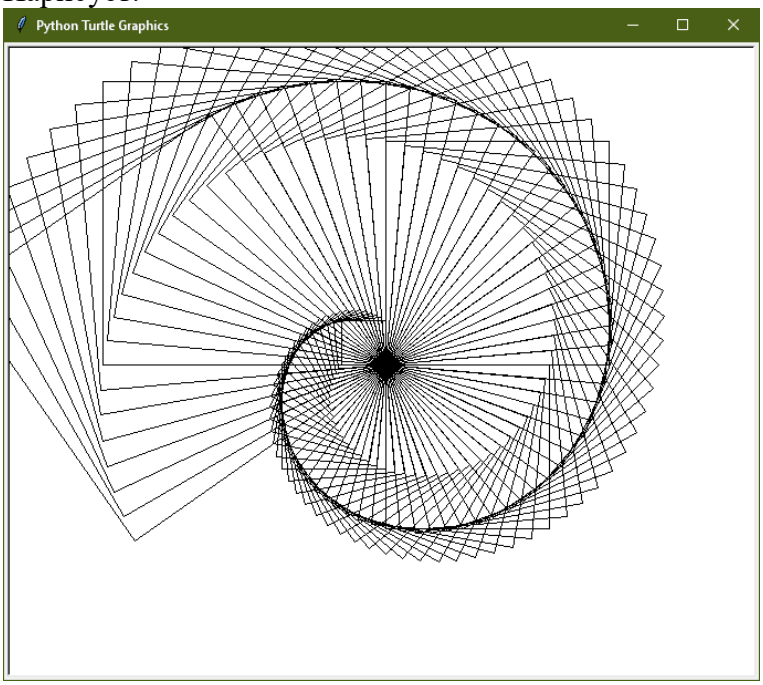


Все так же, но иначе

А этот цикл:

```
size = 40

for i in range(0, 80):
    polygon(4, size)
    turtlePen.left(5)
    size = size + 3
```

Нарисует:



Нужно будет переместить центр…

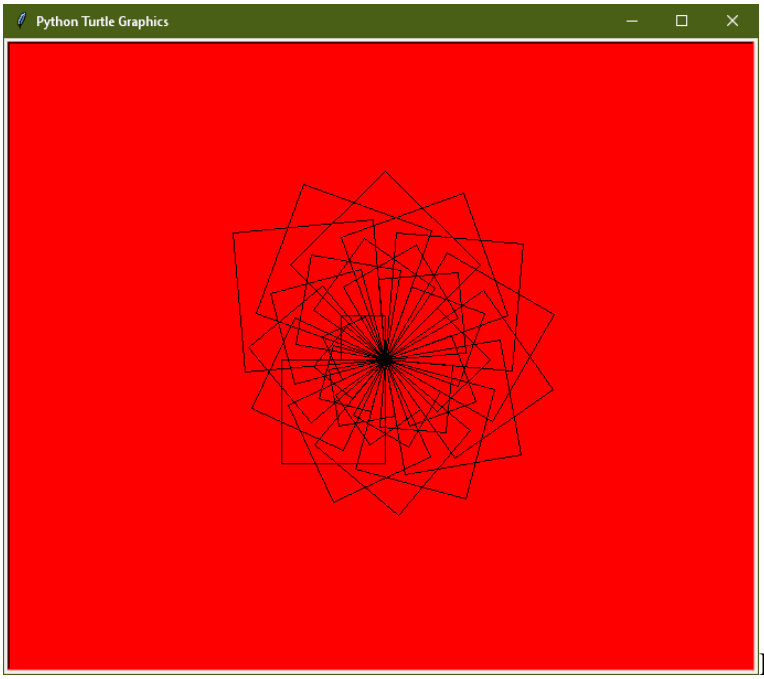
Иными словами — **экспериментируйте**! Так можно получить самые интересные изображения.

А пока добавим всему этому красок!

Используем цвета

Фон окна можно изменить командой **bgcolor**, куда передаем имя текста в виде строки:

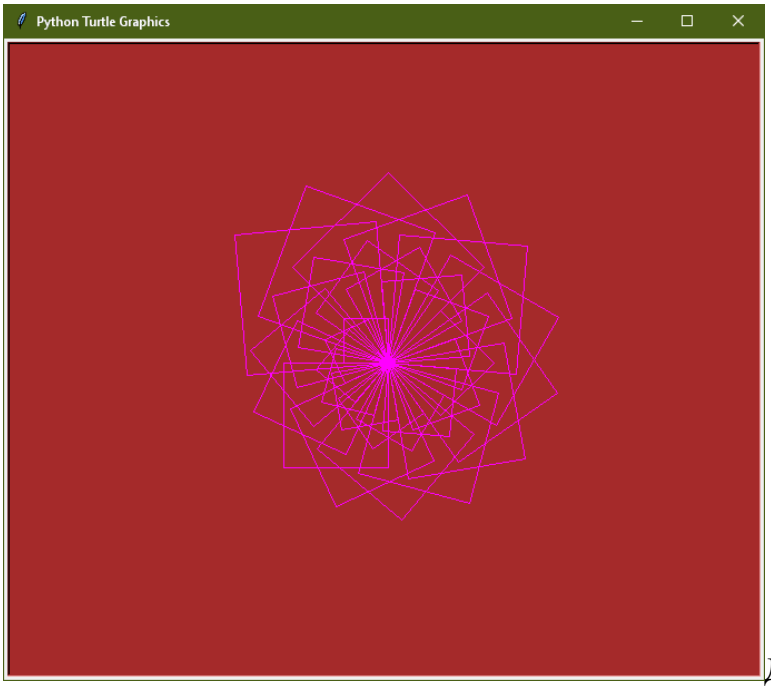
```
window.bgcolor("red")
```



Выбор явно не лучший, но для примера сойдет ^_^. Далее буду использовать **brown**

Аналогично при помощи команды **color** указываем цвет для рисования:

```
turtlePen.color("magenta")
```



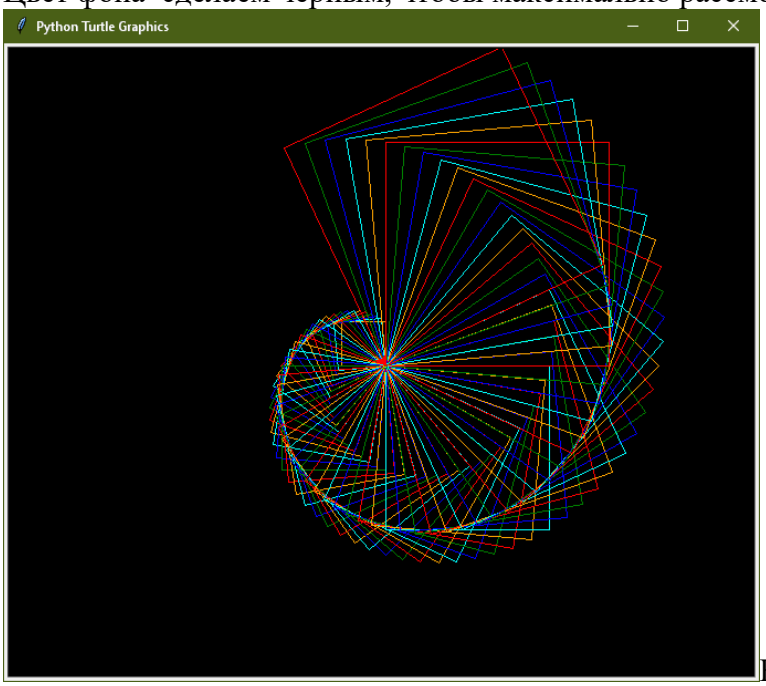
Думаю вполне неплохо

Цвет можно менять и во время рисования. Для этого создадим массив цветов и в зависимости от значения цикла (или иных параметров) будем менять цвет линий.

```
import turtle

turtlePen = turtle.Turtle()
window = turtle.Screen()

window.bgcolor("black")


def polygon(n, size=80):
    if n > 2:
        angle = 360 / n

        for n in range(0, n):
```

```
            turtlePen.left(angle)
            turtlePen.forward(size)


    turtlePen.speed(100)

    colors = ['orange', 'cyan', 'blue', 'green', 'red']

    size = 40

    for i in range(0, 60):
        turtlePen.color(colors[i % 5])
        polygon(4, size)
        turtlePen.left(5)
        size = size + 3

    window.mainloop()
```

Цвет фона сделаем черным, чтобы максимально рассмотреть это цветовое безумие.



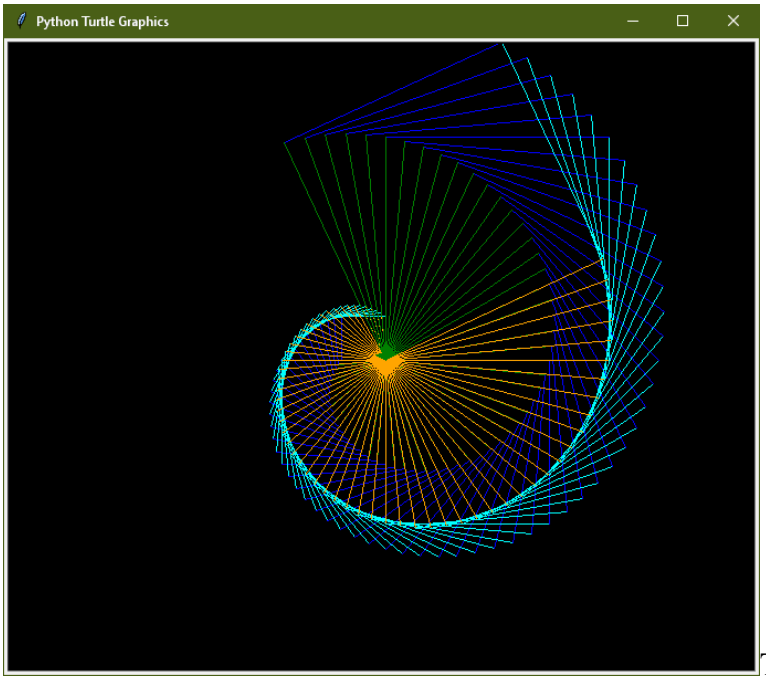
Все интереснее!

Если же вы хотите разрисовать конкретные стороны своих фигур, то цвет нужно указать в цикле отрисовки сторон (и убрать из цикла отрисовки фигур).

В итоге функция **polygon** должна выглядеть приблизительно так:

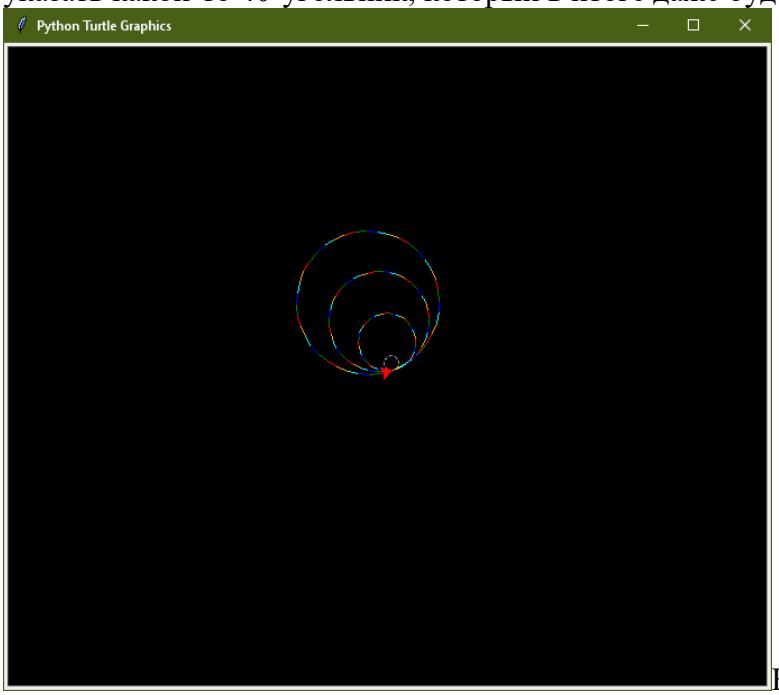
```
def polygon(n, size=80):
    if n > 2:
        angle = 360 / n

        for n in range(0, n):
            turtlePen.color(colors[n % 5])
            turtlePen.left(angle)
            turtlePen.forward(size)
```

А изображение получит совершенно новый вид:



Так даже круче!

Напоследок вспомним о кругах. Ведь наша программа рисует только многоугольники. И хотя можно указать какой-то 40-угольник, который в итоге даже будет похож на круг:

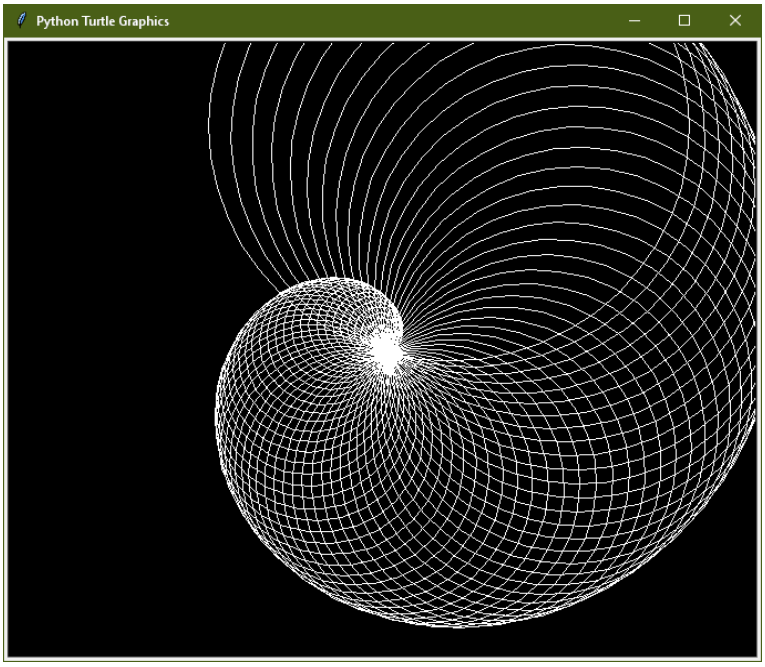


Ну почти. Зато красиво!

В программе есть для этого отдельная функция: **circle**. И заменив функцию **polygon** в цикле отрисовки:

```
for i in range(0, 70):
    turtlePen.color("white")
    turtlePen.circle(size)
    turtlePen.left(5)
    size = size + 3
```

Получаем очередной интересный узор, но только с кружками:



Лучше так, чем 40 угольник!

Демонстрация программных продуктов

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ:

Личностные

- имеет развитый интерес к информационно-техническим наукам, профессиям этой области; - сформированы ценности здорового и безопасного образа жизни
- умеет ориентироваться в мире современных профессий.

Метапредметные

- умеет определять понятия, устанавливать причинно-следственные связи, строить логическое рассуждение. Умозаключение.
- умеет создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;
- умеет оценивать правильность выполнения учебной задачи, собственные возможности её решения.

Универсальные учебные регулятивные действия:

- умеет самостоятельно планировать пути достижения целей, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;
- оценивает соответствие результата цели и условиям и при необходимости корректировать цель и процесс её достижения.

Универсальные коммуникативные действия:

- умеет осознанно использовать речевые средства в соответствии с задачей коммуникации для выражения своих чувств, мыслей и потребностей для планирования и регуляции своей деятельности.

Предметные

- умеет осуществлять математическую и информационную постановку задач по обработке информации;
- имеет базовые навыки в сфере информационных технологий; - использует инструментальные средства обработки информации.

ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

№	Тема	Электронный ресурс (активная ссылка на	Кол-во

		материал, платформа)	часов
1			
2			
3			