Ткаченко Кирилл Станиславович
Инженер 1-й категории
Федерального государственного автономного образовательного учреждения высшего образования «Севастопольского государственного университета».

Генерируемая ASCII-графика на школьном алгоритмическом языке

Рассматривается программная реализация генерации ASCII-графики на школьном алгоритмическом языке. Она может быть полезна для изучения работы с подпрограммами, циклами, ветвлениями.

Изучение любого современного языка программирования высокого уровня во многих случаях начинается с несложных примеров. Эти примеры включают в себя работу с основными языковыми конструкциями, операторами и операциями, стандартными подпрограммами. Такие примеры должны быть интересны. Поэтому, например, можно рассмотреть программное построение ASCII-графики [1].

Рис. 1. Символ

Целью настоящей работы является разработка программы на школьном алгоритмическом языке для построения ASCII-иллюстрации.

Подходящей ASCII-иллюстрацией для рассматриваемого случая является известный и распространенный символ «Инь и Ян» (рис. 1), для построения которого существуют программы на высокоуровневых языках[2].Поскольку начертание этого символа состоит из окружностей, то для проверки принадлежности точки окружности можно использовать известные соотношения [3]. Также стоит отметить, что на языке 1С имеется разработанная программа [4].

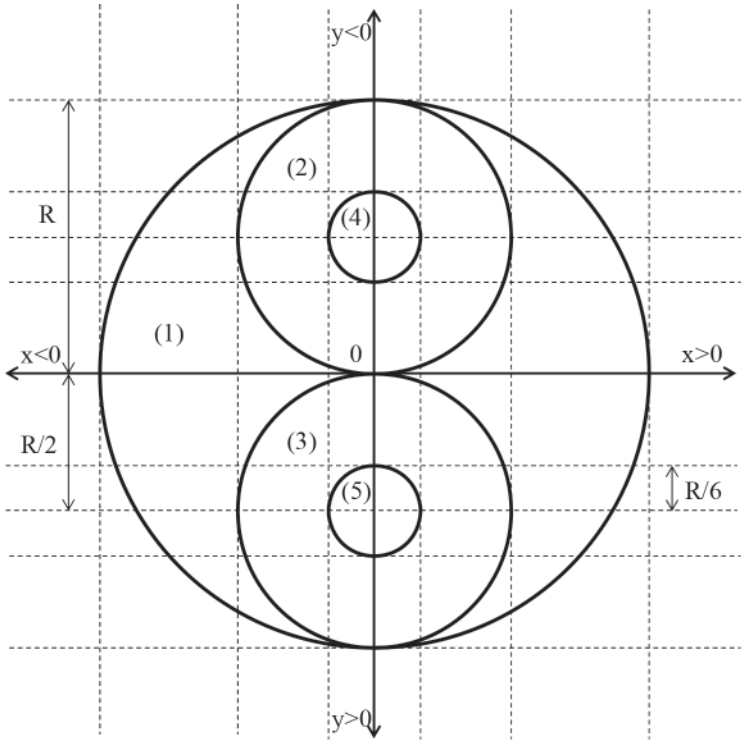

Рис. 2. Схема символа

Для дальнейшего изложения приводится иллюстрация символа в выбранной системе координат на рисунке 2 (соответствует рисунку 1 для языка 1С[4]).

На рисунке 1 обозначается R – радиус символа, (1) – большой круг, (2) – верхний средний круг, (3) – нижний средний круг, (4) – верхний малый круг, (5) – нижний малый круг. Для удобства обработки, координата уизменяется сверху вниз от меньших значений к большим.

Программа на школьном алгоритмическом языке начинается с головного алгоритма Потенциал. Этот алгоритм выводит на экран результаты работы функций для построения ASCII-изображения требуемого символа с различными радиусами:

```
алгПотенциал
нач
  вывод ИньЯн(8), нс;
  вывод ИньЯн(16), нс;
кон
```

Основной алгоритм ИньЯн (цел Радиус) возвращает строку, содержащую ASCII-изображение символа заданного радиуса Радиус. Вначале этого алгоритма определяются константы, содержащие символы для отображения цвета точек, ЦВЕТ_БЕЛЫЙ, ЦВЕТ_ЧЕРНЫЙ, и ЦВЕТ_ПУСТО, для точки, не входящий в рисунок символа:

```
алг лит ИньЯн(цел Радиус)
нач
  лит ЦВЕТ_БЕЛЫЙ = ".";
  лит ЦВЕТ_ЧЕРНЫЙ = "o";
  лит ЦВЕТ_ПУСТО = " ";
```

В целочисленных переменных x, y находятся параметры циклов, которые соответствуют точкам на изображении (координата xдля оси абсцисс для наглядности увеличивается в два раза), РадI, РадII – радиусы средних и малых окружностей, входящих в символ, yI, yII – ординаты координат центров средних и малых окружностей, x2 – значение координаты для оси абсцисс без масштабирования. Строковая переменнаяРез хранит результат:

```
  цел x, y;
  цел РадI, РадII;
  цел yI, yII;
  цел x2;
  лит Рез;
```

Радиус средних окружностей меньше радиуса символа в два раза, малых окружностей – в шесть раз:

```
  РадI := div(Радиус, 2);
  РадII := div(Радиус, 6);
```

Координаты центров по оси ординат средних и малых окружностей выбираются исходя из радиуса средних окружностей:

```
  yI := -РадI;
  yII := РадI;
```

Результат вначале пуст:

```
  Рез := "";
```

По всем допустимым значениям оси ординат, по всем допустимым значениям оси абсцисс с учетом масштабирования, рассчитывается значение координаты x без масштабирования и начинается определение цвета для точки с координатами (x2, y):

```
  нц для y от -Радиус до Радиус
    нц для x от -2 * Радиус до 2 * Радиус
      x2 := div(x, 2);
```

Рассматриваемая точка может принадлежать малому верхнему кругу, тогда ее цвет черный:

```
      если ПринадлежитКругу(x2, y, 0, yI, РадII)
        то
          Рез := Рез + ЦВЕТ_ЧЕРНЫЙ;
```

Рассматриваемая точка может принадлежать малому нижнему кругу, тогда ее цвет белый:

```
        иначе
          если ПринадлежитКругу(x2, y, 0, yII, РадII)
            то
              Рез := Рез + ЦВЕТ_БЕЛЫЙ;
```

Иначе проверяются случаи, когда точка принадлежит большому кругу:

```
            иначе
              если ПринадлежитКругу(x2, y, 0, 0, Радиус)
                то
```

Рассматриваемая точка может принадлежать верхнему среднему кругу, тогда ее цвет белый:

```
                  если ПринадлежитКругу(x2, y, 0, yI, РадI)
                    то
                      Рез := Рез + ЦВЕТ_БЕЛЫЙ;
```

Рассматриваемая точка может принадлежать нижнему среднему кругу, тогда ее цвет черный:

```
                  иначе
                    если ПринадлежитКругу(x2, y, 0, yII, РадI)
                      то
                        Рез := Рез + ЦВЕТ_ЧЕРНЫЙ;
```

В противном случае, если точка находится слева, то ее цвет белый:

```
                      иначе
                        если x < 0
                          то
                            Рез := Рез + ЦВЕТ_БЕЛЫЙ;
```

Либо, если точка находится справа, то ее цвет черный:

```
                          иначе
                            Рез := Рез + ЦВЕТ_ЧЕРНЫЙ;
                        все
                    все
                все
```

В любом другом случае, точка пустая, поскольку не относится к символу:

```
              иначе
                Рез := Рез + ЦВЕТ_ПУСТО;
            все
        все
    все
  кц
```

После построения каждой строки символа требуется переход на новую строку:

```
    Рез := Рез + нс;
  кц
```

Построенное ASCII-изображение возвращается:

```
  знач := Рез;
кон
```

Служебный алгоритм ПринадлежитКругу (цел x, y, x0, y0, Радиус) возвращает «да», если точка с координатами (x, y) принадлежит окружности с центром в точке (x0, y0) и радиусом Радиус, и «нет», в противоположном случае:

```
алг лог ПринадлежитКругу(цел x, y, x0, y0, Радиус)
нач
  знач := (x - x0) ** 2 + (y - y0) ** 2 <= Радиус ** 2;
кон
```

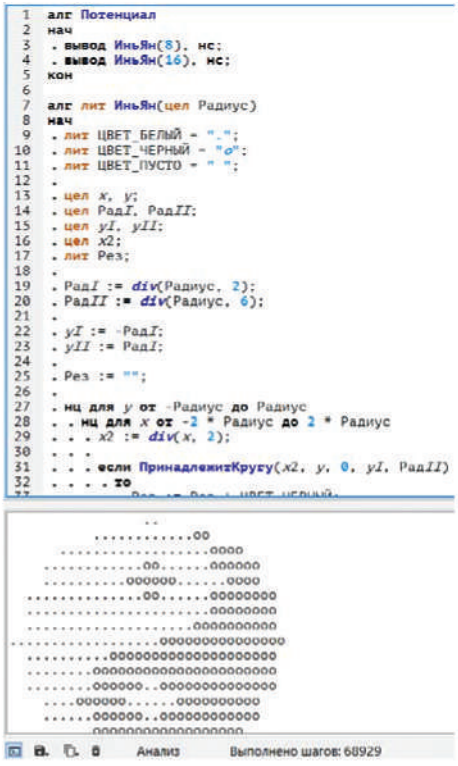

Рис. 3. Снимок экрана

Полный исходный текст программы находится в приложении А, результаты работы программы – приложении Б, запущенная в среде разработки программа – на снимке экрана (рис. 3).

Полученная программа позволит рассмотреть и изучить работу с алгоритмами, циклами, ветвлениями на школьном алгоритмическом языке на достаточно интересном и непростом примере.

Библиографический список использованной литературы

1. ASCII-графика // Википедия. URL: https://ru.wikipedia.org/wiki/ASCII-графика (дата обращения: 01.06.2018).
2. Yin and yang // Rosetta Code. URL: http://rosettacode.org/wiki/Yin_and_yang (дата обращения: 26.05.2018).
3. Окружность // Википедия. URL: https://ru.wikipedia.org/wiki/Окружность (дата обращения: 07.06.2018).
4. *К.С. Ткаченко.* Генерируемая ASCII-графика в 1С / К.С. Ткаченко // Системный администратор. М.: Издательский дом «Положевец и партнеры».ISSN: 1813-5579. № 9 (190), 2018. С.44–47.URL: http://samag.ru/archive/ /article /3716.

ПРИЛОЖЕНИЕ А
Полный исходный текст программы

```
алг Потенциал
нач
  вывод ИньЯн(8), нс;
  вывод ИньЯн(16), нс;
кон

алг лит ИньЯн(цел Радиус)
нач
  лит ЦВЕТ_БЕЛЫЙ = ".";
  лит ЦВЕТ_ЧЕРНЫЙ = "o";
  лит ЦВЕТ_ПУСТО = " ";

  цел x, y;
  цел РадI, РадII;
  цел yI, yII;
  цел x2;
  лит Рез;

  РадI := div(Радиус, 2);
  РадII := div(Радиус, 6);

  yI := -РадI;
  yII := РадI;

  Рез := "";

  нц для y от -Радиус до Радиус
    нц для x от -2 * Радиус до 2 * Радиус
      x2 := div(x, 2);

      если ПринадлежитКругу(x2, y, 0, yI, РадII)
        то
          Рез := Рез + ЦВЕТ_ЧЕРНЫЙ;

        иначе
          если ПринадлежитКругу(x2, y, 0, yII, РадII)
            то
              Рез := Рез + ЦВЕТ_БЕЛЫЙ;
```

```
            иначе
              если ПринадлежитКругу(x2, y, 0, 0, Радиус)
                то

                  если ПринадлежитКругу(x2, y, 0, yI, РадI)
                    то
                      Рез := Рез + ЦВЕТ_БЕЛЫЙ;

                    иначе
                      если ПринадлежитКругу(x2, y, 0, yII, РадI)
                        то
                          Рез := Рез + ЦВЕТ_ЧЕРНЫЙ;

                        иначе
                          если x < 0
                            то
                              Рез := Рез + ЦВЕТ_БЕЛЫЙ;

                            иначе
                              Рез := Рез + ЦВЕТ_ЧЕРНЫЙ;
                          все
                      все
                  все

                иначе
                  Рез := Рез + ЦВЕТ_ПУСТО;
              все
          все
      все
    кц

    Рез := Рез + нс;
  кц

  знач := Рез;
кон

алг лог ПринадлежитКругу(цел x, y, x0, y0, Радиус)
нач
  знач := (x - x0) ** 2 + (y - y0) ** 2 <= Радиус ** 2;
кон
```

ПРИЛОЖЕНИЕ Б
Результаты работы программы