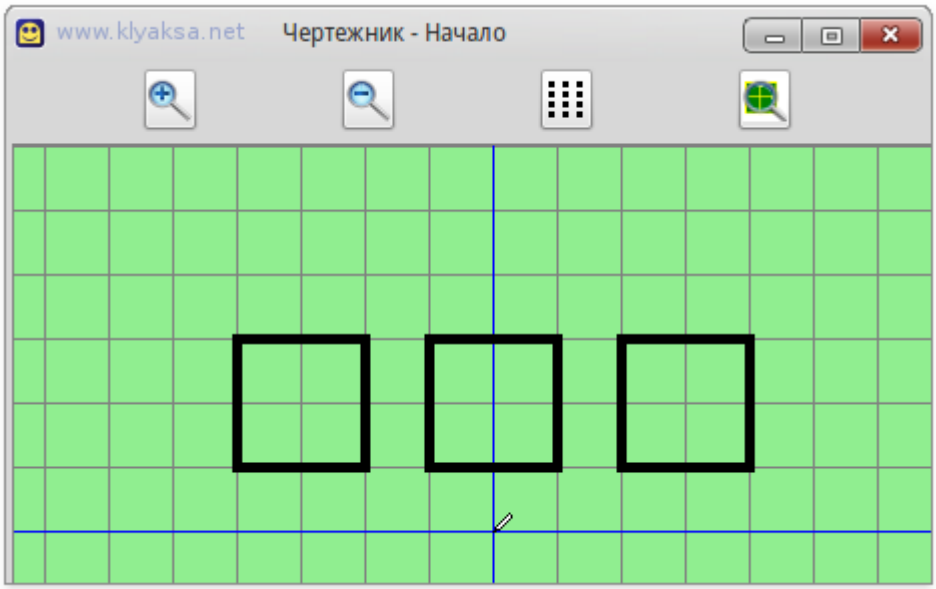
Чертежник учится, или Использование вспомогательных алгоритмов

Рассмотрим задание: необходимо нарисовать несколько (например, три) квадрата.



Алгоритм может выглядеть так (см. ниже). Но получается слишком много одних и тех же строк, выделим их желтым цветом. А если нам захочется нарисовать десять одинаковых квадратов? Сто?

```
использовать Чертежник
алг квадраты 1
нач
. сместиться в точку(-4,1)
. опустить перо
. сместиться на вектор(0,2)
. сместиться на вектор(2,0)
. сместиться на вектор(0,-2)
. сместиться на вектор(-2,0)
. поднять перо
. сместиться в точку(-1,1)
. опустить перо
. сместиться на вектор(0,2)
. сместиться на вектор(2,0)
. сместиться на вектор(0,-2)
. сместиться на вектор(-2,0)
. поднять перо
. сместиться в точку(2,1)
. опустить перо
. сместиться на вектор(0,2)
. сместиться на вектор(2,0)
. сместиться на вектор(0,-2)
. сместиться на вектор(-2,0)
. поднять перо
. сместиться в точку(0,0)
кон
```

При создании средних по размеру программ используется структурное программирование, идея которого заключается в том, что структура программы должна отражать структуру решаемой задачи, чтобы алгоритм решения был ясно виден из исходного текста. Программа разбивается на множество подпрограмм (вспомогательных алгоритмов), каждая из которых выполняет какое-то действие, предусмотренное исходным заданием.

Комбинируя вспомогательные алгоритмы, удается сформировать итоговый алгоритм используя блоки кода (подпрограммы), имеющих определенную смысловую нагрузку. Обращаться к ним, можно по их имени. Очень важная характеристика вспомогательных алгоритмов - это возможность их повторного использования.

Блоки кода, выделенные желтым цветом, запишем отдельным алгоритмом. Назовем его квадрат. Таким образом, мы как бы научим Чертежника новой команде квадрат.

```
использовать Чертежник
алг квадраты
нач
. сместиться в точку(-4,1)
. квадрат
. сместиться в точку(-1,1)
. квадрат
. сместиться в точку(2,1)
. квадрат
. сместиться в точку(0,0)
кон
алг квадрат
нач
. опустить перо
. сместиться на вектор(0,2)
. сместиться на вектор(2,0)
. сместиться на вектор(0,-2)
. сместиться на вектор(-2,0)
. поднять перо
кон
```

Получили более компактный и понятный алгоритм. Там, где какие-то части программы повторяются (используются) несколько раз, удобно использовать вспомогательные алгоритмы.

Обратите внимание — если нам потребуется изменить алгоритм так чтобы рисовались квадраты со стороной не два, а три, то при использовании вспомогательного алгоритма количество правок заметно уменьшается.

Примечание: все алгоритмы в программе на КуМире пишутся друг за другом, для удобства их можно отделять пустой строкой, главным считается алгоритм записанный первым, именно он начинает выполняться после запуска программы.

Метод последовательной детализации

Использованный нами подход облегчает программирование сложных задач. Задача разбивается на более простые подзадачи. Решение каждой оформляется в виде вспомогательного алгоритма, а основной алгоритм организует связку между ними.

Метод программирования, при котором сначала пишется основная программа, в ней записываются обращения к пока еще не составленным подпрограммам, а потом описываются эти подпрограммы, называется методом последовательной (пошаговой) детализации. Причем количество шагов детализации может быть гораздо большим, чем в нашем примере, поскольку сами подпрограммы могут содержать внутри себя обращения к другим подпрограммам.

Сборочный метод

Возможен и другой подход к построению сложных программ: первоначально составляется множество подпрограмм, которые могут понадобиться при решении задачи, а затем пишется основная программа, содержащая обращения к ним. Подпрограммы могут быть объединены в библиотеку подпрограмм и сохранены в долговременной памяти компьютера. Такую библиотеку можно постепенно пополнять новыми подпрограммами.

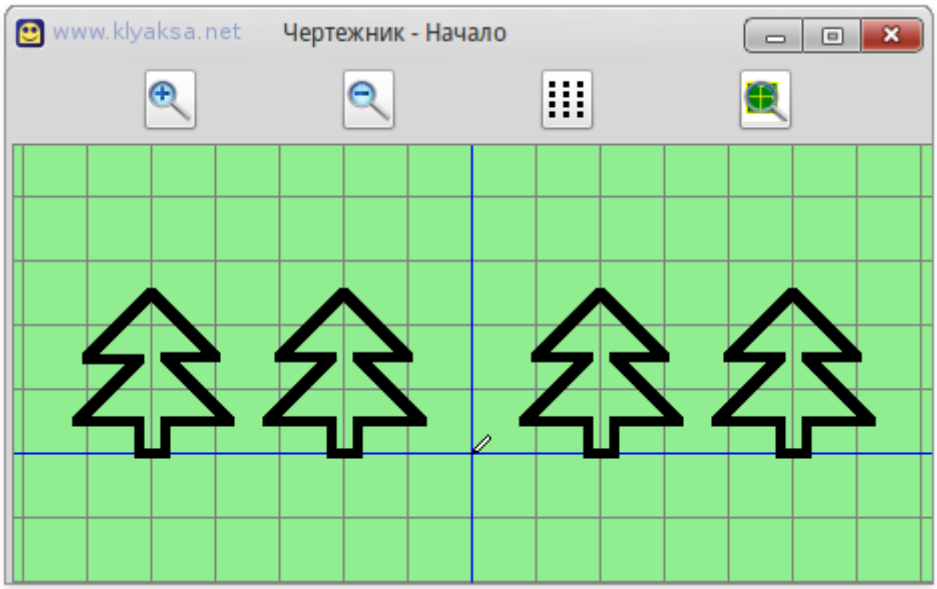
Например, если для управления графическим исполнителем создать библиотеку процедур рисования всех букв и цифр, то программа получения любого текста будет состоять из команд обращения к библиотечным процедурам.

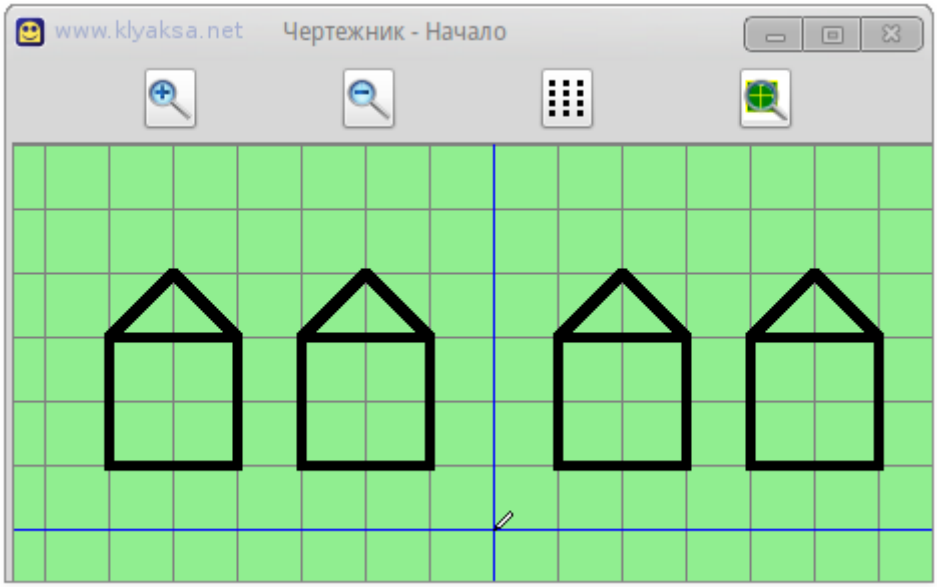
Описанный метод называется сборочным программированием.

Часто в литературе по программированию используется такая терминология: метод последовательной детализации называют программированием сверху вниз, а сборочный метод — программированием снизу вверх.

Задание 1. Измените полученный алгоритм, так чтобы рисовалось 5, 7 или 10 квадратов.

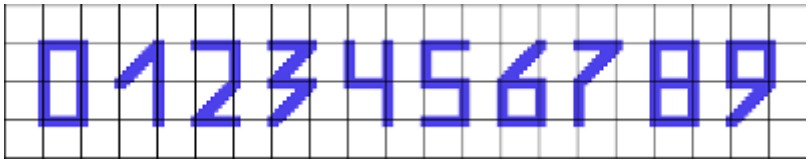
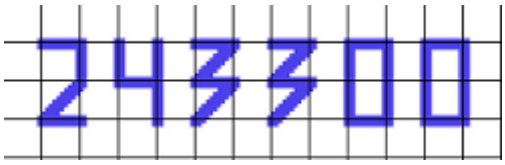
Задание 2. Составьте программы лес, используя вспомогательный алгоритм елка, и улица, используя вспомогательный алгоритм домик.





Домашнее задание

Задание 3. Цифры в почтовом индексе записываются следующим образом:



Составьте вспомогательные алгоритмы рисования этих цифр. Можно сделать это с одноклассниками — каждому по цифре, а потом объедините их вместе. Составьте программу рисования индексов городов Унеча (243300), Елец (399770), Таганрог (347900), Торжок (172011), Плёс (155555). В программе используйте вспомогательные алгоритмы.