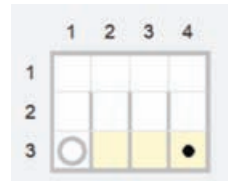


Продолжение таблицы

Программа, записанная первым учащимся	Программа, записанная вторым учащимся
<pre> fill_cell(), move_down() fill_cell(), move_down() fill_cell(), move_right() fill_cell(), move_right() fill_cell(), move_up() fill_cell(), move_up() fill_cell(), move_up() move_left() fill_cell(), move_up() move_left() fill_cell(), move_up() fill_cell(), move_up() run_tasks() </pre>	<pre> fill_cell(), move_down() fill_cell(), move_down() fill_cell(), move_right() fill_cell(), move_right() fill_cell(), move_up() fill_cell(), move_up() fill_cell(), move_up() fill_cell(), move_up() move_left() fill_cell(), move_up() move_left() fill_cell(), move_up() fill_cell(), move_up() run_tasks() </pre>

8 Составьте программу для решения задачи `upr_12_8`. Реализуйте два алгоритма: один с использованием алгоритмической конструкции *следование*, другой — с использованием вспомогательного алгоритма. Сравните решения.



§ 13. Алгоритмическая конструкция повторение

13.1. Алгоритмы с циклами

В окружающем мире можно наблюдать ситуации, при которых различные действия, процессы и события повторяются. Одни повторяются несколько раз и завершаются. Другие могут повторяться очень долго (например, круговорот воды в природе, бесконечное движение планет в космическом пространстве и т. д.).

Понятие цикла используется в различных сферах человеческой деятельности. Под циклом понимают совокупность явлений, процессов, составляющих кругооборот в течение определённого промежутка времени. С этой точки зрения можно говорить о годовом цикле вращения Земли вокруг Солнца или о производственном цикле.

Пример 13.1. Приготовление пельменей.

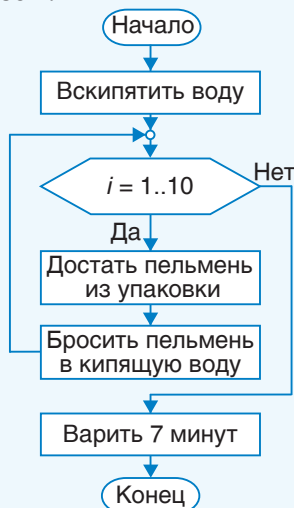


Алгоритм:

1. Вскипятить воду.
2. Для i от 1 до 10 повторять:
 - а) достать пельмень из упаковки;
 - б) бросить пельмень в кипящую воду.
3. Варить 7 минут.

В данном примере параметр цикла i изменяется от 1 до 10. Действия «Достать пельмень из упаковки» и «Бросить пельмень в кипящую воду» составляют тело цикла. Тело цикла выполнится 10 раз (для значений $i = 1, 2, \dots, 10$).

Блок-схема данного алгоритма будет выглядеть следующим образом:



Человеку также регулярно приходится выполнять повторяющиеся действия: умываться, готовить еду, посещать парикмахерскую, ходить на работу и др.

Как правило, человек составляет программы, в которых каждая команда в отдельности и весь алгоритм в целом выполняются за конечное число повторений.

Алгоритмическая конструкция повторение (цикл) определяет последовательность действий, выполняемых многократно. Эту последовательность действий называют **телом цикла**.

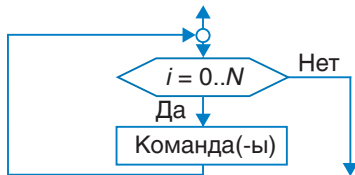
Существует несколько способов управления тем, сколько раз будет повторяться тело цикла.

Команда **цикл с параметром (цикл со счётчиком, цикл «для»)** — способ организации алгоритмической конструкции **повторение (цикл)**, при котором количество повторов зависит от начального и конечного значений параметра цикла.

Цикл с параметром организует выполнение команд тела цикла заранее известное число раз (пример 13.1).

Параметр цикла играет роль счётчика и определяет нумерацию повторов тела цикла. Часто нумерацию начинают с 0 и ограничивают числом N . Цикл выполнится N раз для значений $0, 1, \dots, N - 1$. Если нумерация установлена двумя произвольными числами $N1$ (начальное значение) и $N2$ (конечное значение), то цикл выполнится $(N2 - N1)$ раз (примеры 13.2, 13.3).

Алгоритмическая конструкция цикла с параметром может изображаться на блок-схеме следующим образом:



В этой конструкции в прямоугольнике(-ах) записываются повторяющиеся команды алгоритма (тело цикла), которые выполняются N раз (да). При этом после каждого выполнения команд тела цикла происходит проверка, сколько раз уже выполнен цикл. На блок-схеме переход на проверку условия изображается в виде стрелки, выходящей из тела цикла и возвращающейся

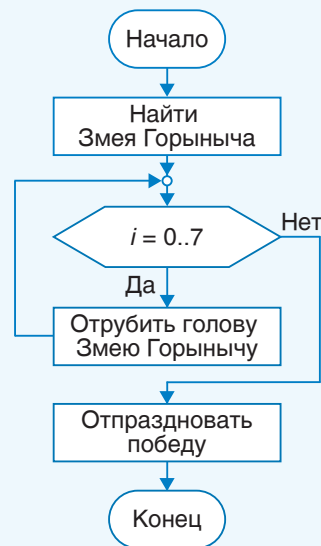
Пример 13.2. Вычислить a^n (например, $3^5 = 243$). Алгоритм возведения числа в степень может быть описан следующим образом:

1. Ввести значения a и n .
2. Определить начальное значение результата $r = 1$.
3. Для i от 0 до n повторять: умножить результат на a .
4. Записать результат.

Пример 13.3. В фольклорных произведениях часто встречается многоголовый Змей Горыныч (количество голов может быть равно, например, 7). Алгоритм победы над Змеем Горынычем может быть описан следующим образом:

1. Найти Змея Горыныча.
2. Для i до 7 повторять: отрубить голову Змею Горынычу.
3. Отпраздновать победу.

Блок-схема данного алгоритма:



Пример 13.4. Работа цикла `for i in range (N1, N2)`.

Вначале цикл выполняется для значения $i = N1$. Затем значение i увеличивается на 1 и цикл выполняется для значения $i = N1 + 1$. Так продолжается до тех пор, пока i не достигнет значения $N2$. При значении $i = N2$ цикл завершает работу и команды в теле цикла больше не выполняются. Последний раз команды в теле цикла выполняются для значения $i = N2 - 1$.

Многие роботы, которые используются в быту и на производстве, могут выполнять циклические алгоритмы. Примером такого робота является суши-робот, который может производить от 450 до 4000 заготовок для суши в час.



к проверке. Как только команды тела цикла выполнятся N раз (нет), цикл завершится. Если $N \leq 0$, то команды из тела цикла не выполнятся ни разу.

Для записи цикла с параметром в языке Python используется команда `for`. Формат записи команды:

```
for i in range (N1, N2):
    тело цикла
```

Строка `for i in range (N1, N2)` является заголовком цикла. Эту строку можно прочитать следующим образом: «Для переменной i в диапазоне от $N1$ до $N2$ делай...». Переменную i называют **переменной (параметром) цикла**. Одно выполнение тела цикла называют **итерацией цикла**. Процесс работы цикла `for` описан в примере 13.4.

Команды тела цикла записываются со сдвигом относительно заголовка цикла.

В языке Python существуют другие варианты записи цикла `for`. Если нужно указать только количество выполнений цикла, то его можно записать так:

```
for i in range (N):
    тело цикла
```

Такая запись соответствует записи:

```
for i in range (0, N):
    тело цикла
```

Если переменная цикла должна изменяться не на 1, то можно задать шаг её изменения:

```
for i in range (N1, N2, h):
    тело цикла
```

В этом случае при каждой итерации цикла значение переменной i будет изменяться на величину h . Значение h может быть положительным или отрицательным целым числом, но не может равняться нулю (пример 13.5).

13.2. Использование команды *цикл с параметром* для исполнителя *Робот*

Пример 13.6. Написать программу для закрашивания 10 клеток поля *Робота* по образцу.

Для закрашивания всех 10 клеток поля нужно в цикле выполнить 10 раз команды:

```
Закрасить
Вправо
```

Данные команды образуют тело цикла.

Командами, образующими тело цикла, могут быть любые команды из системы команд исполнителя. Кроме того, в теле цикла

Пример 13.5. Варианты цикла `for`.

Уменьшение переменной цикла на 1:

```
for i in range (5, 0, -1):
```

Переменная цикла будет принимать значения: 5, 4, 3, 2, 1.

Увеличение переменной цикла на 10:

```
for i in range (-20, 30, 10):
```

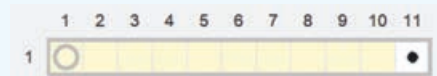
Переменная цикла будет принимать значения: -20, -10, 0, 10, 20.

Использование только нечётных чисел от 1 до 100:

```
for i in range (1, 100, 2):
```

Переменная цикла будет принимать значения: 1, 3, 5, ..., 97, 99.

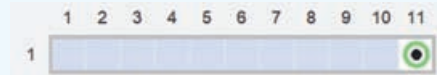
Пример 13.6. Начальная обстановка.



Программа
для исполнителя *Робот*:

```
from pyrob.api import *
@task
def prim_13_5():
    for i in range (10):
        fill_cell()
        move_right()
run_tasks()
```

Результат работы программы:



Пример 13.7. Начальная обстановка.



Программа
для исполнителя *Робот*:

```
from pyrob.api import *
@task
def prim_13_7():
    def kvadrat():
        fill_cell()
        move_down()
        fill_cell()
        move_left()
        fill_cell()
        move_up()
        fill_cell()
        move_left()

    for i in range (5):
        kvadrat()

run_tasks()
```

может вызываться вспомогательный алгоритм. Использование вспомогательного алгоритма позволяет сократить запись тела цикла и делает программу более понятной.

Пример 13.7. Написать программу для закрашивания клеток поля *Робота* по образцу.

На поле исполнителя *Робот* присутствуют стены. При обходе стен *Робот* выполняет следующие команды:

Закрасить, вниз
Закрасить, влево
Закрасить, вверх
Закрасить, влево

Для решения задачи данную группу команд *Робот* должен повторить 5 раз. Оформим эти команды как вспомогательный алгоритм `kvadrat` и вызовем в цикле этот алгоритм. В данном примере тело цикла состоит из одной команды `kvadrat`.

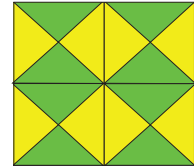


1. Какие циклические процессы, происходящие в окружающем мире, вы можете назвать?
2. Что понимают под алгоритмической конструкцией *повторение*?
3. Что такое тело цикла?
4. Что такое команда *цикл с параметром*?
5. Какие ключевые слова входят в описание заголовка цикла с параметром в Python?
6. Что такое итерация цикла?
7. Какие есть способы записи команды цикла в языке Python?

   **Упражнения**

1 Опишите словесно или изобразите с помощью блок-схемы следующие алгоритмы.

1. Рисование в графическом редакторе следующего изображения (повторить 4 раза рисование квадрата с диагоналями и закрасненными областями).



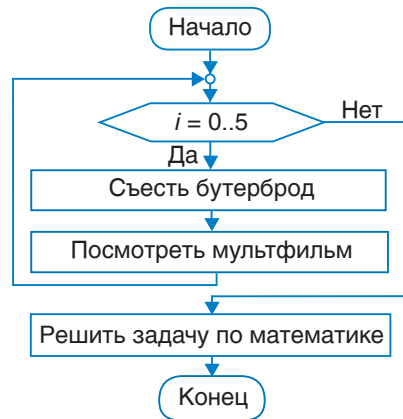
2. Сверление 10 отверстий.

3. Сервировка стола к обеду на 6 персон.

4. Каждую минуту бактерия делится надвое. Изначально есть одна бактерия. За бактериями наблюдали 10 минут. Определите количество бактерий в конце наблюдения. Заполните таблицу согласно алгоритму.

Время, мин	0	1	2	3	4	5	6	7	8	9	10
Количество бактерий	1	2									

2 Семиклассник Вася после школы пригласил своего друга Колю помочь ему в решении 5 задач по математике. В гостях Коля посоветовал Васе провести остаток дня, воспользовавшись следующим алгоритмом, записанным в виде блок-схемы. Почему Вася получил двойку по математике?



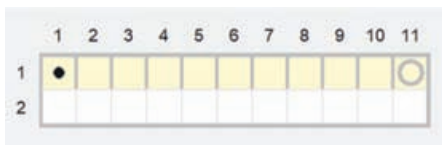
3 Составьте программу для решения задачи `upr_13_3`. Сравните алгоритм решения этой задачи с примером 13.5. Что у них общего? Чем они отличаются?



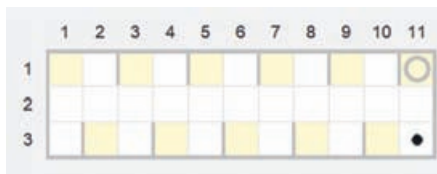
4 Составьте программу для решения задачи `upr_13_4`. Сравните её решение с программой из упражнения 3 и примера 13.5.



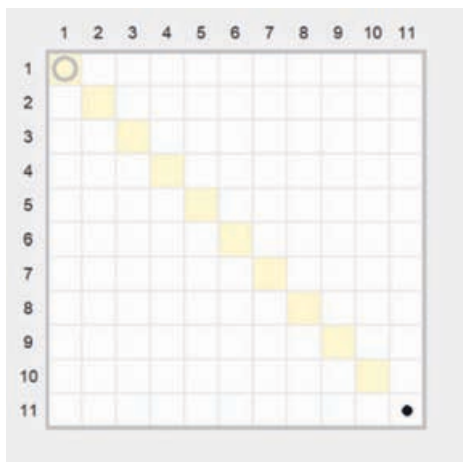
5 Составьте программу для решения задачи `upr_13_5`. Используйте вспомогательный алгоритм для решения задачи.



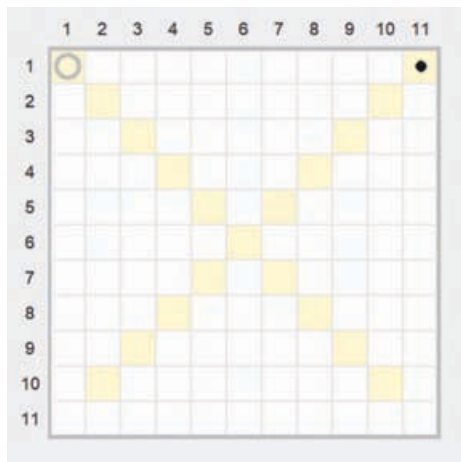
6* Составьте программу для решения задачи `upr_13_6`. Используйте вспомогательный алгоритм для решения задачи.



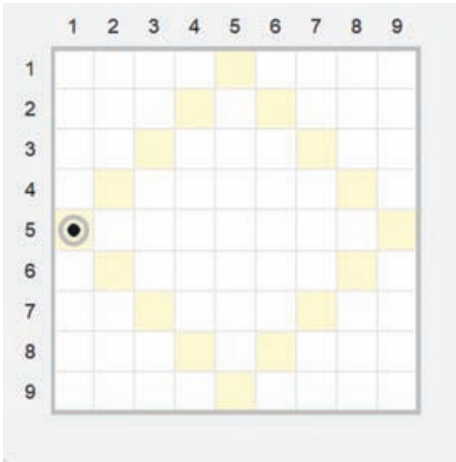
7 Составьте программу для решения задачи `upr_13_7`.



8* Составьте программу для решения задачи `upr_13_8`.



9 Для решения задачи `upr_13_9` Петя составил алгоритм и записал программу. Петин младший брат Олег удалил несколько команд. Сколько команд удалил Олег? Восстановите программу, которую написал Петя.



```

from pyrob. api import *
@task
def upr_13_9():
    for i in range (4):
        fill_cell()
        move_right()
        move_down()
    for i in range (4):
        move_right()
        move_up()
    for i in range (4):
        fill_cell()
        move_down()

run_tasks()

```

10 Вася пытается представить, как можно было бы использовать роботов в различных ситуациях, описанных в литературных произведениях. Например, для Тома Сойера, которого тётушка Поли отправила красить забор, Вася придумал робота-маляра. Вася решил, что такому роботу в системе команд достаточно одной команды: покрась доску. Алгоритм покраски забора из 20 досок Вася записал так:

1. Установить робота у левого края забора
2. Для $i = 0..20$ повторять
3. Покрась доску

Сможет ли робот-маляр покрасить забор? В чём ошибка Васи? Исправьте алгоритм, добавив необходимую(-ые) команду(-ы).

§ 14. Использование условий

14.1. Условия для исполнителя

Робот

Как и многие другие исполнители, исполнитель *Робот* может не только выполнять действия, но и проверять условия.

Условием для *Робота*, как и для любого исполнителя, является

Робот-пылесос проверяет истинность условий с помощью датчиков. Датчики расстояния позволяют ему «видеть» препятствия и не касаться их. Если он наезжает на препятствие под углом, то срабатывают датчики столкновения. В этом случае пылесос меняет своё направление по заданному алгоритму.