

```

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

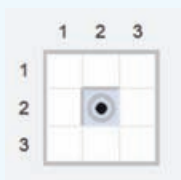
Робот

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.

Пример 14.1. Начальная обстановка *Робота*.



В данном случае для *Робота* будут истинны следующие условия:

```
wall_up()
wall_left()
cell_is_filled()
```

Ложными будут условия:

```
wall_down()
wall_right()
```

Пример 14.2. Сбор грибов.

Использование цикла с параметром при составлении алгоритма решения этой задачи может привести к разным результатам. Или корзина может быть полупустой, или не все найденные грибы в неё поместятся.



понятное ему высказывание, которое может быть истинным (соблюдаться) либо ложным (не соблюдаться). Рассмотрим систему условий для компьютерного исполнителя *Робот*.

Условие	Результат
wall_up()	Истинно, если стена <i>сверху</i>
wall_down()	Истинно, если стена <i>снизу</i>
wall_left()	Истинно, если стена <i>слева</i>
wall_right()	Истинно, если стена <i>справа</i>
cell_is_filled()	Истинно, если текущая клетка закрашена

Истинность условий для исполнителя *Робот* проверяется для конкретной обстановки (пример 14.1).

14.2. Команда *цикл с предусловием*

Цикл с параметром используется при составлении алгоритма в том случае, когда заранее известно количество повторений. Однако часто до выполнения цикла количество повторений неизвестно.

Пример 14.2. Вы с родителями пошли в лес собирать грибы (предполагается, что они там есть). Ваши действия можно описать командами: найти гриб, срезать гриб, положить гриб в корзину. Эти действия будут выполняться в цикле, но вы заранее не знаете, сколько грибов поместится в корзину. Поэтому можно говорить не о количестве повторений (количество грибов), а об условии, при котором вы будете продолжать сбор грибов: пока корзина не заполнена.

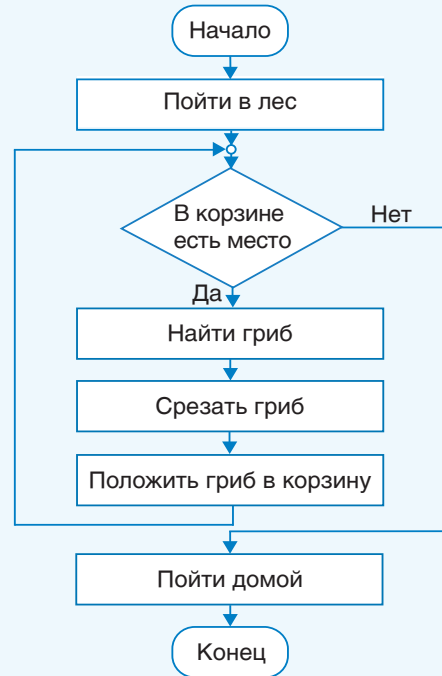
Команда цикл с условием (цикл «пока») — это такой способ организации алгоритмической конструкции *повторение* (цикл), при котором количество выполнений команд тела цикла зависит от истинности или ложности условия цикла.

Цикл с условием используется в том случае, когда количество повторений тела цикла заранее неизвестно, но известно условие продолжения работы.

Условие цикла определяет, как долго будет выполняться цикл.

Пример 14.2. Продолжение.

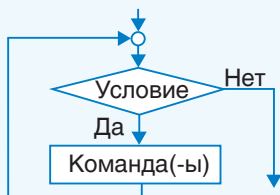
При использовании цикла с условием домой унесём полную корзину грибов.



Циклы выполняются до тех пор, пока условие цикла истинно. Иногда нужно прервать выполнение цикла без проверки условия. Для этого используют команду `break`, которая прекращает выполнение цикла и передаёт управление команде, следующей за циклом.

Иногда возникает необходимость принудительно начать следующую итерацию цикла, пропустив часть команд в его теле. Для таких случаев существует ключевое слово `continue`.

Пример 14.3. Блок-схема цикла с предусловием.



Пример 14.4. Принцип работы команды *цикл с предусловием*.

В качестве условия записывается логическое выражение, которое определяет, как долго будет выполняться цикл. В теле цикла записываются повторяющиеся команды алгоритма, которые выполняются пока верно условие цикла (да).

На блок-схеме команды тела цикла заключены в прямоугольники, а условие — в ромб.

Сначала проверяется условие, и если оно истинно, то выполняются команды тела цикла. После каждого выполнения команд тела цикла автоматически происходит возврат к проверке истинности условия.

Как только условие станет ложным (нет), цикл завершится.

Количество итераций цикла будет на 1 меньше, чем количество проверок истинности условия цикла. Если условие сразу ложно, то цикл не выполнится ни разу, т. е. количество итераций цикла будет равно нулю.

Пока условие истинно, выполняются команды, составляющие тело цикла. Цикл прекращает выполняться тогда, когда условие становится ложным. *Цикл с предусловием* имеет такое название, поскольку проверка условия предваряет выполнение команд тела цикла.

В примере 14.3 показана блок-схема *цикла с предусловием*. Принцип работы *цикла с предусловием* описан в примере 14.4.

Если условие в цикле будет всегда истинно (всегда да), то такой цикл не сможет завершиться. Возникшую ситуацию называют **зацикливанием**.

Для записи цикла с предусловием в Python используется команда `while`. Формат записи команды:

```
while <условие>:
    тело цикла
```

Строка `while <условие>`: является **заголовком цикла**. Эту строку можно прочитать следующим образом: «Пока верно условие, делай...». Команды тела цикла записываются со сдвигом относительно заголовка цикла.

14.3. Использование команды *цикл с предусловием* для исполнителя *Робот*

Пример 14.5. Написать программу для решения следующей задачи. *Робот* находится в верхнем левом углу поля. Снизу от *Робота* вдоль всего поля расположена горизонтальная линия с проходом в одну нижнюю клетку. Составить алгоритм, выполнив который, *Робот* сможет пройти через проход и закрасить клетку. Расположение прохода заранее неизвестно.

Нам известно, что проход не ограничен стеной снизу. *Робот* может двигаться вправо до тех пор, пока внизу есть стена.

Пока снизу стена **повторять**

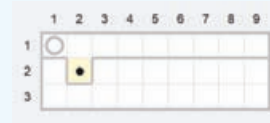
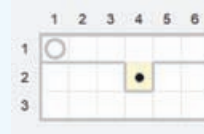
Вправо

Робот остановится в той клетке, у которой снизу нет стены. После этого *Робот* должен переместиться вниз и закрасить клетку.

Пример 14.6. Написать программу для закраски клеток коридора переменной длины.

В данной задаче нам не известна длина коридора. *Робот* может двигаться, пока справа пусто, и закрашивать клетки:

Пример 14.5. Возможные начальные обстановки.

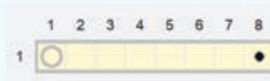


Задачи, в которых могут быть различные обстановки, проверяются на 4–6 различных тестовых примерах. Задача считается решённой, если программа правильно работает для каждого из них.

Программа
для исполнителя *Робот*:

```
from pyrob.api import *
@task
def prim_14_5():
    while wall_down():
        move_right()
        move_down()
        fill_cell()
run_tasks()
```

Пример 14.6. Одна из возможных начальных обстановок.



Пример 14.6. Продолжение.

Программа
для исполнителя *Робот*:

```
from pyrob.api import *
@task
def prim_14_4():
    while not wall_right():
        fill_cell()
        move_right()
        fill_cell()
run_tasks()
```

Пока нет стены справа **повторять**
Закрасить
Вправо

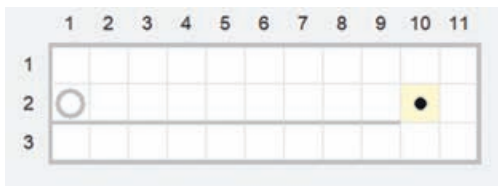
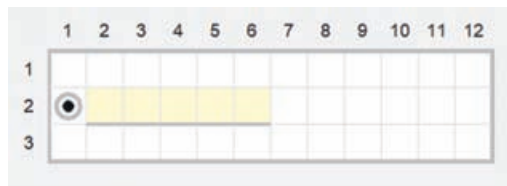
После прохождения всего коридора *Робот* должен закрасить последнюю клетку. Это действие происходит после выполнения цикла (команда `fill_cell()`), так как для последней клетки условие «справа пусто» уже не выполняется — там расположена стена.



1. Что понимают под условием для исполнителя?
2. В каких случаях в алгоритмах используется команда *цикл с предусловием*?
3. Как работает *цикл «пока»*?
4. Как записывается команда *цикл с предусловием* на языке Python?
5. Когда возникает ситуация зацикливания?

**Упражнения**

1. Напишите программы для решения задач `upr_14_1_a` и `upr_14_1_b` из встроенного задачника. Обращайте внимание на начальное и конечное положение *Робота*.

`upr_14_1_a``upr_14_1_b`

2 Для исполнителя *Робот* был написан следующий алгоритм:

```
from pyrob.api import *

@task
def upr_14_2():
    while not wall_right():
        fill_cell()
        move_down()
        move_right()
        fill_cell()
        move_up()
        move_right()

run_tasks()
```

Нарисуйте в тетради результат работы алгоритма. Какими должны быть размеры поля, чтобы *Робот* не врезался в стену? Определите начальное положение *Робота*.

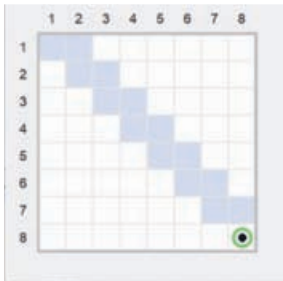
3 Составьте алгоритм, выполнив который, *Робот* нарисует «узор» из упражнения 2 вдоль левого края поля исполнителя (задача `upr_14_3`). Каким должен быть вертикальный размер поля исполнителя?

4 *Робот* находится на квадратном поле неизвестного размера. Начальное положение *Робота* — верхний левый угол. Составьте и выполните алгоритм, по которому *Робот* переместится из начального положения в нижний правый угол и закрасит все клетки своего пути. На каком (каких) из рисунков изображено решение этой задачи? Поясните.

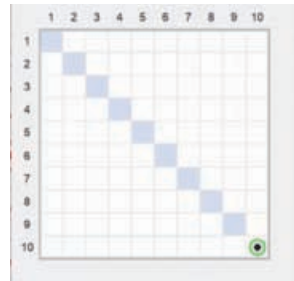
а)



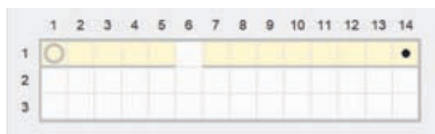
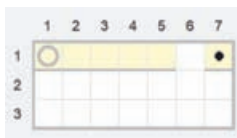
б)



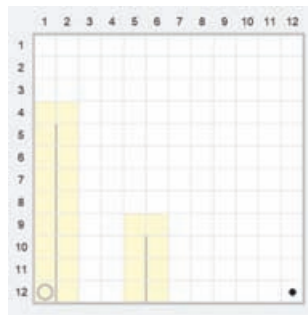
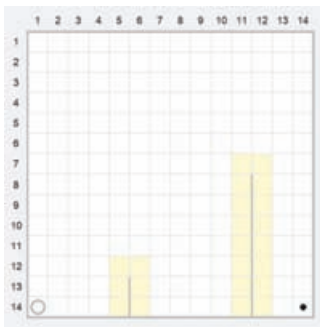
в)



5 На поле *Робота* размещён «забор» — горизонтальная стена. Забор нужно «покрасить» — написать программу для закрашивания всех клеток сверху стены (задача `upr_14_5`). В «заборе» могут быть одни «ворота» — клетка без линий. Длина «забора» и расположение «ворот» неизвестны.



6* Решите задачу `upr_14_6`. Используйте вспомогательный алгоритм для обхода каждой из двух стен, расположенных на поле *Робота*.



§ 15. Алгоритмическая конструкция ветвление

Пример 15.1. Выбор обуви весной в зависимости от погоды.

Если на улице дождь, **то**
надеть резиновые сапоги
Иначе
надеть туфли

В данном примере в текущий момент времени может быть выполнена только одна команда из двух: или надеть сапоги, или надеть туфли.

15.1. Команда ветвление

Довольно часто на поставленный вопрос человек получает ответ «да» или «нет». В зависимости от ответа человек определяет свои действия и выполняет одну или другую «команду» или группу «команд» (пример 15.1).

Роботы и другие технические устройства также могут выпол-