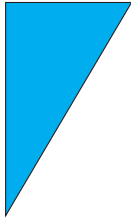


## § 18. Складанне алгарытмаў для работы з графікай

### 18.1. Разлікі ў графічных пабудовах

**Прыклад 18.1.** Нарысаваць прамавугольны трохвугольнік, які адпавядае рысунку (катэты трохвугольніка паралельныя восьям каардынат). Даўжыні катэтаў і каардынаты прамога вугла ўводзяцца.



Этапы выканання задання

I. Зыходныя даныя:  $a$  і  $b$  (даўжыні катэтаў),  $x$  і  $y$  (каардынаты вяршыні прамога вугла).

II. Вынік: адлюстраванне прамавугольнага трохвугольніка.

III. Алгарытм рашэння задачы.

1. Увод зыходных даных.

2. Каб адлюстраваць трохвугольнік, трэба выканаць наступныя дзеянні:

1) пабудаваць лініі з пункта з каардынатай  $(x; y)$  у пункты з каардынатамі  $(x + a; y)$  і  $(x; y + b)$ ;

2) злучыць лініяй пункты  $(x + a; y)$  і  $(x; y + b)$ ;

3) зафарбаваць трохвугольнік. Для зафарбоўвання трохвугольніка трэба ведаць каардынаты якога-небудзь пункта ўнутры трохвугольніка. Такім пунктам

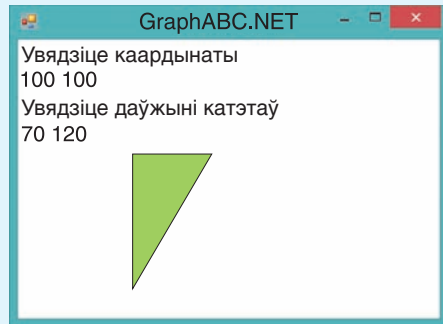
#### Прыклад 18.1.

V. Праграма:

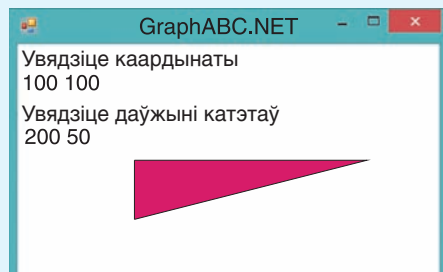
```
uses GraphABC;
var a,b,x,y,x_c, y_c:integer;
begin
writeln('Увядзіце каардынаты');
read(x,y); writeln(x,' ',y);
writeln('Увядзіце даўжыні
      катэтаў');
read(a,b); writeln(a,' ',b);
Line(x,y,x+a,y); Line(x,y,x,y+b);
Line(x+a,y,x,y+b);
//Каардынаты пункта
//Унутры трохвугольніка
x_c := x + 2; y_c := y + 2;
FloodFill(x_c,y_c,clRandom);
end.
```

VI. Тэсціраванне.

Запусціць праграму і ўвесці значэнні: каардынаты (100; 100), даўжыні катэтаў 70 і 120. Вынік:



Іншы варыянт:



**Прыклад 18.2\*.**

V. Праграма:

**uses** GraphABC;**var** x,y,d: integer;**begin**

writeln('Каардынаты');

read(x,y);

writeln(x, ' ',y);

writeln('Старана');

read(d);

writeln(d);

SetBrushColor(clYellow);

  Ellipse(x + d div 3,y,  
          x + 2 \* d div 3,y + d);

SetBrushColor(clBrown);

  Pie(x + d div 2,y + d div 3,  
      d div 3,0,180);**end.**

VI. Тэсціраванне.

Запусціць праграму і ўвесці значэнні: каардынаты (100; 100), старана 150. Вынік:



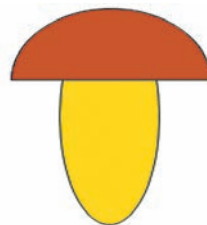
\* Грыб можна ўпісаць не ў квадрат, а ў прамавугольнік. У гэтым выпадку трэба задаваць дзве велічыні, якія вызначаюць памеры прамавугольніка: даўжыню ( $d_1$ ) і шырыню ( $d_2$ ).

у дадзеным выпадку можа быць пункт з каардынатамі  $(x + 2; y + 2)$ <sup>1</sup>.

IV. Апісанне пераменных: усе пераменныя маюць тып integer.

Шмат якія графічныя пабудовы можна падагульніць, калі дапусціць, што фігура павінна быць упісана ў квадрат. У гэтым выпадку для пабудовы фігуры дастаткова задаць каардынаты  $(x; y)$  верхняга левага вугла квадрата і даўжыню яго стараны —  $d$ . Выкарыстоўваючы гэтыя велічыні, можна атрымаць каардынаты іншых вяршынь квадрата:  $(x + d; y)$ ,  $(x; y + d)$ ,  $(x + d; y + d)$ . Можна атрымаць каардынаты сярэдзіны стараны  $(x + d \text{ div } 2; y)$  або цэнтры квадрата  $(x + d \text{ div } 2; y + d \text{ div } 2)$ .

**Прыклад 18.2\*.** Нарысаваць у графічным акне грыб. Задаць каардынаты верхняга левага вугла квадрата і даўжыню яго стараны для вызначэння месцазнаходжання і памераў грыба.



## Этапы выканання задання

I. Зыходныя даныя:  $x$  і  $y$  — каардынаты верхняга левага вугла квадрата, у які ўпісаны грыб,  $d$  — даўжыня яго стараны.

II. Вынік: відарыс грыба.

<sup>1</sup> Для адвольнага трохвугольніка можна выкарыстаць формулы:

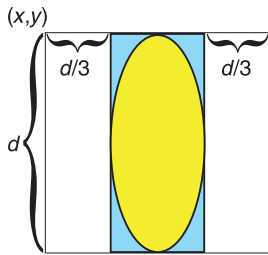
$$\left( \frac{x_1 + x_2 + x_3}{3}; \frac{y_1 + y_2 + y_3}{3} \right).$$

### III. Алгарытм рашэння задачы.

1. Увод зыходных даных.

2. Для таго каб пабудаваць грыб, трэба выканаць наступныя дзеянні:

1) пабудаваць авал для адлюстравання ножкі грыба. Параметры каманды для адлюстравання эліпса вызначым наступным чынам: `ellipse(x + d div 3, y, x + 2*d div 3, y + d)`;



2) нарысаваць шапачку грыба. Для гэтага можна выкарыстоўваць каманду `Pie` (пабудова сектара круга). Каардынаты цэнтра  $(x + \frac{d}{2}; y + \frac{d}{3})$ , радыус —  $\frac{d}{3}$ . Вуглы роўны  $0^\circ$  і  $180^\circ$  адпаведна.

IV. Апісанне пераменных: усе пераменныя маюць тып `integer`.

Выпадковыя лікі маюць шырокае выкарыстанне ў праграмаванні. Яны выкарыстоўваюцца ў шыфраванні і ў мадэляванні. Шмат якія камп'ютарныя гульні выкарыстоўваюць выпадковыя лікі. На аснове выпадковых лікаў генерыруюцца капчы і паролі, рэалізуюцца розныя латарэі.

В `PascalABC` для атрымання выпадковага ліку выкарыстоўваюць функцыю `random`. Спосабы запісу функцыі:

`Random(a, b)`; — вяртае выпадковыя цэлыя ў дыяпазоне ад  $a$  да  $b$ ;

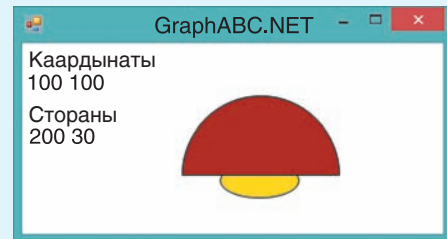
`Random(a)`; — вяртае выпадковыя цэлыя ў дыяпазоне ад  $0$  да  $a - 1$ ;

### Прыклад 18.2\*. Працяг.

У праграму рысавання грыба трэба ўнесці змяненні, якія дазваляюць разлічыць становішча ножкі і шапачкі грыба адносна каардынат  $(x; y)$  і велічынь  $d1$  і  $d2$ .

```
var x,y,d1,d2: integer;
...
writeln('Стораны');
read(d1, d2);
writeln(d1, ' ', d2);
SetBrushColor(clYellow);
Ellipse(x+d1 div 3, y,
        x + 2 * d1 div 3, y + d2);
SetBrushColor(clBrown);
Pie(x + d1 div 2, y + d2 div 3,
    max(d1,d2) div 3, 0, 180);
```

Вынік:



**Выпадковы лік** — лік, які прымае адно значэнне з мноства, прычым з'яўленне таго ці іншага значэння нельга дакладна прадказаць. Напрыклад, калі б лікі з'явіліся ў выніку выцягвання бочачак у лато, то такая паслядоўнасць лікаў была б выпадковай.

У мовах праграмавання выкарыстоўваюць **псеўдавыпадковыя лікі**, якія атрымліваюць з выкарыстаннем генератара выпадковых лікаў — алгарытму, што спараджае паслядоўнасць лікаў, элементы якой амаль незалежныя адзін ад аднаго і звычайна размеркаваны раўнамерна на зададзеным інтэрвале.

**Прыклад 18.3.**

V. Праграма:

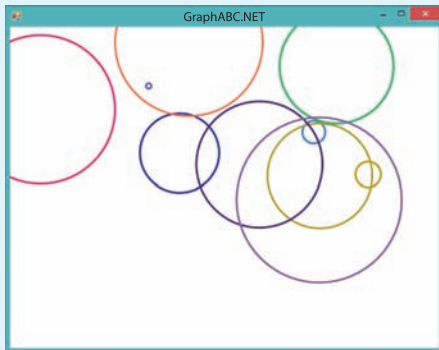
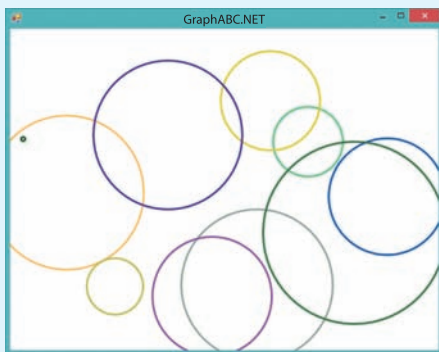
```

uses GraphABC;
var x,y,r: integer;
begin
  SetPenWidth(3);
  SetBrushStyle(bsClear);
  for var i:= 1 to 10 do
  begin
    x := random(600);
    y := random(400);
    R := random(150);
    SetPenColor(clRandom);
    circle(x,y,r);
  end;
end.

```

VI. Тэсціраванне.

Запусціць праграму. Павінна быць нарысавана 10 акружнасцей. Розныя варыянты работы праграмы:



Random; — вяртае выпадковы рэчыўны ў дыяпазоне [0..1).

Функцыя clRandom дазваляе задаць выпадковы колер.

**Прыклад 18.3.** Напісаць праграму для рысавання на экране 10 рознакаляровых акружнасцей. Размяшчэнне акружнасцей, іх радыусы і колер вызначаюцца выпадковым чынам.

Этапы выканання задання

I—II. Вынікам работы праграмы, якая не залежыць ад зыходных даных, будзе адлюстраванне 10 акружнасцей.

III. Алгарытм рашэння задачы.

1. Устаноўім таўшчыню ліній у 3 пікселі і празрыстую заліўку.

2. Значэнні каардынат цэнтра акружнасці і яе радыуса вызначаюцца функцыяй random. Значэнне колеру для мяжы круга — clRandom.

3. Паколькі колькасць паўтарэнняў вядомая, будзем выкарыстоўваць цыкл for.

IV. Апісанне пераменных:  $x$ ,  $y$  (каардынаты цэнтра),  $r$  (радыус) маюць тып integer.

## 18.2. Выкарыстанне дапаможных алгарытмаў

Пабудову фігур можна афармляць у выглядзе дапаможных алгарытмаў. Гэта дазволіць выкарыстоўваць такія алгарытмы для рашэння іншых задач.

Усе графічныя працэдуры, якія выкарыстоўваліся раней, мелі параметры. Яны дазвалялі вызначаць месцазнаходжанне і памер фігур. Карыстальнік таксама можа скласці свой дапаможны алгарытм з параметрамі.

Агульны выгляд працэдуры з параметрамі:

```
procedure <імя> (<спіс  
параметраў>:тып);  
  
var ...  
begin  
  <каманды>  
end;
```

Пры выкліку працэдуры важна памятаць, што колькасць параметраў і іх парадак павінны адпавядаць таму, як працэдура апісана.

**Прыклад 18.4.** Напісаць праграму для пабудовы  $n$  раўнабедраных прамавугольных трохвугольнікаў з даўжынёй катэта  $a$ . Размяшчэнне трохвугольнікаў вызначаецца выпадковым чынам.

Этапы выканання задання

I. Зыходныя даныя:  $n$  (колькасць трохвугольнікаў),  $a$  (даўжыня катэта).

II. Вынік: адлюстраванне  $n$  трохвугольнікаў.

III. Алгарытм рашэння задачы.

1. У прыкладзе 18.3 адлюстроўвалі акружнасці, размешчаныя выпадковым чынам, а ў прыкладзе 18.1 — прамавугольныя трохвугольнікі. Выкарыстаем праграмы гэтых прыкладаў.

2. Увод значэнняў пераменных  $n$  і  $a$ .

3. Паколькі колькасць паўтарэнняў вядомая, будзем выкарыстоўваць цыкл **for**.

4. Зменім праграму з прыкладу 18.3. Для гэтага каманду **circle** (пабудова акружнасці) заменім на каманду пабудовы трохвугольніка:

1) месцазнаходжанне трохвугольніка задаецца каардынатамі прамога вугла, якія вызначым выпадковым чынам;

#### Прыклад 18.4.

V. Праграма:

```
uses GraphABC;  
var n, x, y, a : integer;  
procedure pr_treug (x, y, a,  
  b : integer);  
var x_c, y_c:integer;  
begin  
  line(x, y, x + a,y);  
  line(x, y, x, y + b);  
  line(x + a, y, x, y + b);  
  x_c := x + 2; y_c := y + 2;  
  FloodFill(x_c,y_c,clRandom);  
end;  
begin  
  writeln('Увядзіце колькасць');  
  read(n); writeln (n);  
  writeln('Увядзіце даўжыню  
  катэта');  
  read(a); writeln (a);  
  for var i:= 1 to n do  
    begin  
      x:= random(500);  
      y:= random(400);  
      pr_treug(x, y, a, a);  
    end;  
end.  
VI. Тэсціраванне.  
Запусціць праграму. Вынік:
```



Пры ўводзе іншых значэнняў вынікі будуць іншымі.

**Прыклад 18.5\*.**

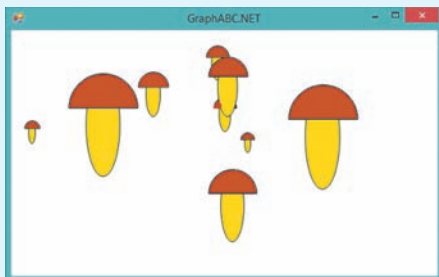
```

V. Праграма:
uses GraphABC;
var x,y,d: integer;

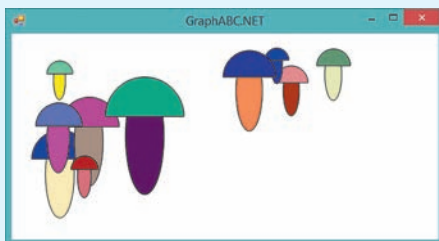
procedure grib(x,y,d:integer);
begin
  SetBrushColor(clYellow);
  Ellipse(x + d div 3, y,
    x + 2*d div 3, y + d);
  SetBrushColor(clBrown);
  Pie(x + d div 2, y + d div 3,
    d div 3, 0, 180);
end;

Begin
  for var i:= 1 to 10 do
  begin
    x:= random(400);
    y:= random(200);
    d:= random(150);
    grib(x,y,d);
  end;
end.
VI. Тэспіраванне.
Вынік можа быць наступным:

```



Можна дабавіць размалёўванне выпадковым колерам:



2) катэты прамавугольнага трохвугольніка маюць аднолькавую даўжыню — значэнне  $a$ .

5. Пабудову аднаго прамавугольнага трохвугольніка апішам у дапаможным алгарытме `pr_treug`. Параметры працэдуры пабудовы трохвугольніка — каардынаты вяршыні прамога вугла і даўжыні катэтаў. Алгарытм апісаны ў прыкладзе 18.2.

IV. Апісанне пераменных: усе пераменныя маюць тып `integer`.

**Прыклад 18.5\*.** Нарысаваць 10 грыбоў. Размяшчэнне і іх памеры вызначаюцца выпадковым чынам.

Этапы выканання задання

I—II. Вынікам работы праграмы, які не залежыць ад зыходных даных, будзе адлюстраванне 10 грыбоў.

III. Алгарытм рашэння задачы.

1. Пабудову аднаго грыба апішам у дапаможным алгарытме. Алгарытм апісаны ў прыкладзе 18.2.

2. Значэнні каардынат верхняга левага вугла і памер грыба вызначаюцца функцыяй `random`.

3. Паколькі колькасць паўтарэнняў вядомая, будзем выкарыстоўваць цыкл `for`.

IV. Апісанне пераменных:  $x$ ,  $y$  (каардынаты верхняга левага вугла),  $d$  (памер) — `integer`.

**Прыклад 18.6.** Запоўніць графічнае акно акружнасцямі з радыусам 10.

Этапы выканання задання

I—II. Зыходныя даныя адсутнічаюць. Акружнасці павінны запоўніць усё графічнае акно.

III. Алгарытм рашэння задачы.

1. Задача з'яўляецца абагульненнем задачы з прыкладу 17.1. Каманды праграмы неабходна паўтарыць для нека-


лькіх радоў акружнасцей. Колькасць радоў вызначаецца вышыняй акна. Рысаванне аднаго рада аформім як дапаможны алгарытм `row`.

2. Становішча любога рада акружнасцей вызначаецца каардынатай  $y$ . Для кожнага значэння  $y$ , пакуль ён не стане большым, чым вертыкальны памер экрана, выконваем у цыкле наступнае:

- 1) рысуем рад акружнасцей;
- 2) змяняем  $y$ .

IV. Апісанне пераменных:  $x, y, r$  — `integer`.

У прыкладзе 18.6 паказана, як запоўніць графічнае акно акружнасцямі. Унёсшы невялікія змяненні ў гэту праграму, можна запаўняць графічнае акно любымі іншымі фігуркамі. Для гэтага дастаткова замяніць каманду `Circle(x,y,r)` у працэдуры `row` на іншую каманду. Можна выбраць графічны прымітыў з бібліятэкі `GraphABC` або самастойна напісаць працэдуру рысавання фігуркі (напрыклад, выкарыстаць працэдуру рысавання грыба з прыкладу 18.5).

-  1. Як задаць выпадковы лік?  
2. Як задаць выпадковы колер?  
3. Як апісаць працэдуру з параметрамі?



### Практыкаванні

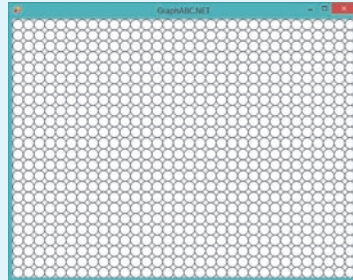
1. Выканайце заданні для прыкладу 18.1.
  1. Паэксперыментуйце з праграмай, уводзячы розныя значэнні зыходных даных.
  2. Раствлумачце, што адбываецца пры ўводзе адмоўных значэнняў даўжынь катэтаў.
  3. Што адбудзецца, калі ўвесці адмоўныя значэнні каардынат? Раствлумачце вынік.
2. Выканайце заданні для прыкладу 18.3.
  1. Выканайце праграму некалькі разоў. Зніміце празрыстую заліўку. Раствлумачце, чаму некаторыя акружнасці нябачныя.

#### Прыклад 18.6.

```
V. Праграма:
uses GraphABC;
var x, y, r : integer;
procedure row(y : integer);
begin
  x := 10; R := 10;
  while x < WindowWidth do
    begin
      Circle(x,y,r); x := x+20;
    end;
end;
begin
  y := 10;
  while y <= WindowHeight do
    begin
      Row(y); y := y + 20;
    end;
end.
```

VI. Тэсціраванне.

Вынік:



2. Унясіце ў праграму такія змяненні, каб можна было адлюстравачь 20 кругоў; 100 кругоў.

3. Які максімальны памер можа мець радыус круга ў праграме? Унясіце ў праграму змяненні так, каб рысаваліся кругі з радыусам, не большым за 20. Колькасць кругоў устанавіце роўнай 10 000.

4. Унясіце змяненні ў праграму так, каб карыстальнік мог уводзіць колькасць кругоў, якія адлюстроўваюцца на экране.

3 Выканайце заданні для прыкладу 18.4.

1. Запусціце праграму некалькі разоў. Растлумачце, чаму пры некаторых запусках трохвугольнікі рысуюцца зверху тэкста ў верхнім левым вугле экрана. Змяніце праграму так, каб трохвугольнікі рысаваліся ніжэй за тэкст (правей за тэкст).

2. Дабаўце ў праграму магчымасць уводу даўжыні другога катэта.

3. Змяніце праграму так, каб даўжыні катэтаў задаваліся выпадковым чынам.

4 Напішыце праграму, якая будзе выпадковым чынам відарысы 20 гарызантальных адрэзкаў даўжынёй 30 пікселяў. Распрацуйце два варыянты рашэння задачы. Адзін з выкарыстаннем цыкла `while`, а іншы — цыкла `for`.

1. Параўнайце дзве праграмы рашэння задачы. Які варыянт рашэння дадзенай задачы ўяўляецца вам лепшым? Чаму?

2. Задайте ў праграме таўшчыню адрэзка ў 3 пікселі.

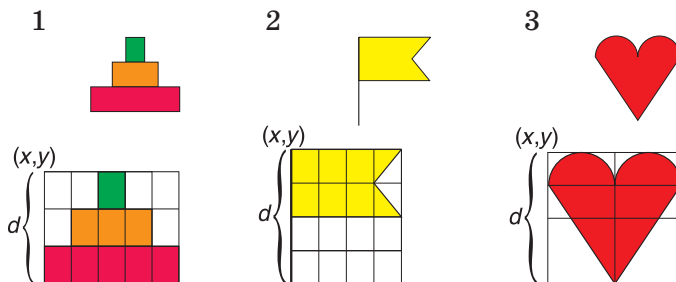
3. Якія змяненні патрэбны ў праграме, каб таўшчыня адрэзка была выпадковым лікам з прамежка [2; 8]?

4. Унясіце змяненні ў праграму так, каб карыстальнік мог уводзіць колькасць адрэзкаў, якія адлюстроўваюцца на экране.

5. Якія змяненні трэба ўнесці ў праграму, каб замест гарызантальных адрэзкаў адлюстроўваліся вертыкальныя? Дыяганальныя?

5 Выкарыстоўваючы працэдуру рысавання трохвугольніка з прыкладу 18.4, нарысуйце рад трохвугольнікаў уздоўж верхняга (левага) краю графічнага акна.

6\* Напішыце праграму для рысавання адной з фігурак. Задаюцца каардынаты верхняга левага вугла і даўжыня стараны квадрата (даўжыні старон прамавугольніка):



- 7\* Напішыце праграму, якая нарысуе рад фігурак уздоўж краю графічнага акна. Выкарыстоўвайце фігуркі, якія рысавалі ў заданні 6.
- 8 Выканайце заданні для прыкладу 18.6.
1. Змяніце ў праграме значэнне  $r = 10$  на  $r = 20$ . Чаму атрымаўся такі рысунак? Паэксперыментуйце са значэннямі радыуса, устанавіўшы празрыстую заліўку.
  2. Якія змяненні трэба ўнесці ў праграму, каб экран запаўняўся кругамі з радыусам 20 без перасячэнняў?
  3. Унясіце змяненні ў праграму так, каб усе кругі былі чырвонымі або рознакаляровымі.
  4. Унясіце ў праграму змяненні так, каб графічнае акно можна было запаўняць кругамі ўведзенага радыуса.
- 9\* Напішыце праграму, якая запоўніць усё графічнае акно:
1. Грыбамі (прыклад 18.2).
  2. Фігуркамі з задання 6.

## § 19. Выкарыстанне асноўных алгарытмічных канструкцый для рашэння практычных задач

### 19.1. Выкарыстанне лікавых паслядоўнасцей

Лікавыя паслядоўнасці дазваляюць апісваць шмат якія працэсы, што адбываюцца ў прыродзе і грамадстве.

Напрыклад, паслядоўнасць лікаў 2, -1, 0, 2, 0, 1, -2 можа задаваць значэнні тэмпературы па днях тыдня; паслядоўнасць 746, 751, 758 — значэнні сярэдняй заробатнай платы супрацоўнікаў прадпрыемства за квартал і г. д.

Паслядоўнасці могуць задавацца формулай, у якой значэнне элемента залежыць ад таго, які ў яго нумар у паслядоўнасці (прыклад 19.1).

Іншым спосабам задання элементаў паслядоўнасці з'яўляецца вызначэнне

Пацвярджаннем важнасці лікавых паслядоўнасцей з'яўляецца той факт, што створана цэлая энцыклапедыя лікавых паслядоўнасцей<sup>1</sup>.

**Прыклад 19.1.** Элементы паслядоўнасці няцотных дадатных лікаў можна апісаць з дапамогай формулы  $a_n = 2n - 1$ . У гэтай формуле  $n$  — нумар элемента ў паслядоўнасці. Мінімальнае значэнне ліку  $n = 1$ . Выкарыстоўваючы формулу, атрымаем паслядоўнасць: 1, 3, 5, 7... .

Элементы паслядоўнасці могуць быць рэчаіснымі лікамі. Напрыклад, формула  $a_n = \frac{n}{n^2 + 1}$  задае наступную паслядоўнасць: 0.5, 0.4, 0.3, 0.235, 0.192, ....

<sup>1</sup> Анлайн-энцыклапедыя цэлалікавых паслядоўнасцей. Рэжым доступу: <http://oeis.org/?language=russian> (дата доступу: 17.01.2018).