Вариант № 1

начало
- **для каждого** $j \in \{1, 2\}$ **выполнять**
  - **если** $z_{1\,j} \neq$ `f`, **то** /* *канал I фазы не свободен* */
    - **если** $t_{1\,j} \leq t$, **то** /* *обработка в I фазе завершена* */
      - **если** $z_{2\,1} =$ `f`, **то** /* *канал II фазы свободен* */
        - `Моделировать_К` $(2, 1, t_{2\,1})$;
        - $z_{2\,1} \leftarrow$ `w` ; /* *канал II фазы теперь занят* */
        - $z_{1\,j} \leftarrow$ `f` ; /* *канал I фазы теперь свободен* */
      - **иначе**
        - $z_{1\,j} \leftarrow$ `b` ; /* *канал I фазы заблокирован* */

Вариант № 2

функция *Simple_ Reflex_ Vacuum_ Agent ([положение, состояние])*:
- **входные данные:** положение: $\in \{A, B\}$;
- состояние: $\in \{$Чисто, Грязно$\}$;
- **начало**
  - **если** *состояние* $=$ *Грязно* **то**
    - **вернуть** *Всасывать*;
  - **иначе если** *положение* $= A$ **то**
    - **вернуть** *Вправо*;
  - **иначе если** *положение* $= B$ **то**
    - **вернуть** *Влево*;

Вариант № 3

начало
- **если** $t_i < t$, **то** /* *поступила заявка* */
  - `Заявка_не_обслужена` $\leftarrow$ `Истина` ;
  - $j \leftarrow 1$;
  - **пока** $j \leq 2$ *и* `Заявка_не_обслужена`, **выполнять**
    - **если** $z_{1\,j} =$ `f`, **то** /* *канал свободен* */
      - `Моделировать_К` $\left(1, j, t_{1\,j}\right)$;
      - $z_{1\,j} \leftarrow$ `w` ; /* *теперь канал занят* */
      - `Заявка_не_обслужена` $\leftarrow$ `Ложь` ;
    - $j \leftarrow j + 1$;
  - Обработка необслуженной заявки ;
  - `Моделировать_И` (t_i) ;

4.1

Вариант № 4

начало
  $\vec{F}_i \leftarrow \vec{0} \quad \forall i \in [1,n]_{\mathbb{N}}$;
  для каждого $i \in [1,n-1]_{\mathbb{N}}$ **выполнять**
    для каждого $j \in [i+1,n]_{\mathbb{N}}$ **выполнять**
      $r_{ij}^2 \leftarrow \rho^2\left(\vec{p}_i, \vec{p}_j\right)$; // *квадрат расстояния*
      $F_{ij} \leftarrow \left(G\, m_i\, m_j\right) / r_{ij}^2$; // *величина силы*
      $\vec{f}_{ij} \leftarrow \vec{p}_j - \vec{p}_i$; // *направление силы*
      $\vec{F}_i \leftarrow \vec{F}_i + F_{ij}\, \vec{f}_{ij} / r_{ij}$; // *учёт силы на i-е тело*
      $\vec{F}_j \leftarrow \vec{F}_j - F_{ij}\, \vec{f}_{ij} / r_{ij}$; // *учёт силы на j-е тело*

Вариант № 5

входные данные: Граница сетки, $k_{\max}$ // *лимит итераций*
выходные данные: $G \in M_{n\times n}(\mathbb{R})$
начало
  обрабатывать $\leftarrow$ „истина“; $k \leftarrow 0$; Инициализировать G;
  пока обрабатывать, **выполнять**
1   **для каждого** $i,j \in [2,n-1]_{\mathbb{N}}$ **выполнять**
        $g'_{ij} \leftarrow 0{,}25\, \left(g_{i-1,j} + g_{i+1,j} + g_{i,j-1} + g_{i,j+1}\right)$;
    $k \leftarrow k+1$;
    если $k \geq k_{\max}$ **или** $\rho\left(G, G'\right) < \varepsilon$ **то**
      обрабатывать $\leftarrow$ „ложь“;
    иначе
      $G \leftarrow G'$;

Вариант № 6

входные данные: $A \in M_{n\times n}(\mathbb{R})$
выходные данные: $A \in M_{n\times n}(\mathbb{R})$ // *L U-разложение*
начало
  для $i = 1,2,\ldots,n$ **выполнять** // *прямой ход*
    $q \leftarrow$ № строки с $\max\left\{|a_{ki}| \quad \forall k \in [i,n]_{\mathbb{N}}\right\}$; // *ведущего*
    Обменять местами строки a_i и a_q;
    $r \leftarrow a_{ii}$; // *вед. эл-т теперь на главной диагонали*
    для $k = i+1, i+2, \ldots, n$ **выполнять**
      $s \leftarrow a_{ki}/r$; $\quad a_{ki} \leftarrow s$;
      для $j = i+1, i+2, \ldots, n$ **выполнять**
        $a_{kj} \leftarrow a_{kj} - s\, a_{ij}$;

Вариант № 7

вспомогательные данные: $w = m/p$; // *ширина полосы*
входные данные: $A \in M_{n\times w}(\mathbb{R})$; // *также выходные*
начало
  для $i = 1, 2, \ldots, n$ **выполнять** // *прямой ход*
    $q \leftarrow$ № строки с $\max\left\{|a_{ki}| \quad \forall k \in [i,n]_{\mathbb{N}}\right\}$;
    Обменяться q, a_{qi} с другими, найти глобальный max;
    Обменять местами строки a_i и a_q (в индексах);
    если *владелец* a_q, **то**
      Отправить a_q остальным;
    иначе
      `Принять_у_любого`(a_q);
    Выполнить элементарные преобразования строк;

Вариант № 8

начало
  для каждого $j \in \{1, 2\}$ **выполнять**
    если $z_1 > 0$, **то** /* *накопитель не пуст* */
      если $z_{1\,j} =$ `f`, **то** /* *канал свободен* */
        `Моделировать_K` $\left(1, j, t_{1\,j}\right)$;
        $z_{1\,j} \leftarrow$ `w` ; /* *теперь канал занят* */
        $z_1 \leftarrow z_1 - 1$; /* *освобождение накопителя* */

Вариант № 9

функция *Tree_ Search (задача, стратегия)*:
- **входные данные:** стратегия: стратегия поиска;
- **вспомогательные данные:** дерево_поиска;
- узел: ссылка на узел из «дерево_поиска»;
- **результат:** решение или неудача;
- **начало**
  - Добавить в «дерево_поиска» начальное состояние из «задача»;
  - **пока** *Истина,* **выполнять**
    - **если** *дерево_ поиска* $\not\ni$ *кандидатов на разёртывание* **то**
      - **вернуть** *неудача*;
    - узел $\leftarrow$ узел из «дерево_поиска» $\sim$ «стратегия»;
    - **если** *узел* $\ni$ *целевое состояние,* **то**
      - решение $\leftarrow \sim$ целевому состоянию;
      - **вернуть** *решение*;
    - **иначе**
      - развернуть узел, добавить полученные узлы в «дерево_поиска»;

Вариант № 10

входные данные: $A, B \in M_{n\times n}(\mathbb{R})$
выходные данные: $C \in M_{n\times n}(\mathbb{R})$
начало
- **для каждого** $i \in [1,n]_{\mathbb{N}}$ **выполнять**
  - **для каждого** $j \in [1,n]_{\mathbb{N}}$ **выполнять**
    - 1 $c_{ij} \leftarrow 0$;
    - **для каждого** $k \in [1,n]_{\mathbb{N}}$ **выполнять**
      - 2 $c_{ij} \leftarrow c_{ij} + a_{ik}\, b_{kj}$;

Вариант № 11

начало
- `Принять_у_процесса`$_0(a_{i*})$
- `Принять_у_процесса`$_0(B)$
- **для каждого** $j \in [1,n]_{\mathbb{N}}$ **выполнять**
  - $c_{ij} \leftarrow 0$;
  - **для каждого** $k \in [1,n]_{\mathbb{N}}$ **выполнять**
    - $c_{ij} \leftarrow c_{ij} + a_{ik}\, b_{kj}$
- `Отправить_процессу`$_0(c_{i*})$;

Вариант № 12

входные данные: $i \in \mathbb{Z}_+, a_{i*}, b_{*i} \in \mathbb{R}^n$
выходные данные: $c_{i*} \in \mathbb{R}^n$
начало
  $c_{ii} \leftarrow \langle a_{i*}, b_{*i} \rangle$
  $i_{\text{next}} \leftarrow 1 + (i+1) \bmod n$; // *№ следующего по кругу*
  $i_{\text{prev}} \leftarrow 1 + (i-1) \bmod n$; // *№ предыдущего по кругу*
  для каждого $j \in [1, i-1]_{\mathbb{N}} \cup [i+1, n]_{\mathbb{N}}$ **выполнять**
    `Отправить_процессу`$_{i_{\text{next}}}(b_{*i})$
    `Принять_у_процесса`$_{i_{\text{prev}}}(b_{*j})$; // *в ту же перем.* b_{*i}
    $c_{ij} \leftarrow \langle a_{i*}, b_{*j} \rangle$;

Вариант № 13

начало
  для каждого $i \in [1, n]_{\mathbb{N}}$ **выполнять**
    `Отправить_процессу`$_1(a_{i*})$;
  для каждого $j \in [1, n]_{\mathbb{N}}$ **выполнять**
    `Отправить_процессу`$_1(b_{*j})$;
  для $i = n, n-1, \ldots, 1$ **выполнять** // *в обратном порядке*
    `Принять_у_процесса`$_n(c_{i*})$;

Вариант № 14

вспомогательные данные: *// глобальные*
$n_r = 0$; *// количество читающих*
$s_{nr} = 1$; *// для защиты n_r*
$s_w = 1$; *// для записи*
$s_r = 1$; *// блокировка одного читателя при запросе на запись*
$n_w = 0$; *// количество пишущих*
$s_{nw} = 1$; *// для защиты n_w*
$s_{rr} = 1$; *// блокировка остальных читателей при запросе на запись*
начало
- **ждать** s_{rr}
  - **ждать** s_r
    - **ждать** s_{nr}
      - $n_r \leftarrow n_r + 1$
      - **если** $n_r = 1$, **то**
        - `Ждать`(s_w)
    - **отпустить**
  - **отпустить**
- **отпустить**

Вариант № 15

входные данные: s ; *// переменная-семафор*
начало
- **пока** `ПУ`$(s.flag)$, **выполнять**
  - Пережидание;
- $s.count \leftarrow s.count - 1$
- **если** $s.count < 0$, **то**
  - Поместить текущий процесс в $s.queue$
  - Заблокировать текущий процесс
- $s.flag \leftarrow 0$;