

2 Система команд Редактора Сред исполнителя Путник

2.1 Команды конструктора

1. **public** редакторПоля()

Конструктор безаргументный, иницирующий поле формата 10×10 и устанавливающий Путника в СЗ угол с ориентацией на Север

2. **public** редакторПоля(**int** колРядов, **int** колКолонок)

Конструктор с параметрами поля, иницирующий поле формата *колРядов* \times *колКолонок* и устанавливающий Путника в СЗ угол с ориентацией на Север,

где *колРядов* – устанавливаемое число рядов поля

колКолонок – устанавливаемое число колонок поля

3. **public** редакторПоля(**int** колРядов, **int** колКолонок, **положениеПутника** образПутника)

Конструктор с параметрами поля и начальным положением Путника, иницирующий поле формата *колРядов* \times *колКолонок* устанавливающий Путника в клетку с координатами *образПутника.координатаX* \times *образПутника.координатаY* ориентацией на *образПутника.ориентация*,

где *колРядов* – устанавливаемое число рядов поля

колКолонок – устанавливаемое число колонок поля

образПутника – местоположение и ориентация начального положения Путника

2.2 Команды внесения изменений в среду

1. **public void** поставитьПрепятствие(**int** номерРяда, **int** номерКолонки)

Общедоступный метод установки препятствия в клетку поля с заданными координатами

где *номерРяда* – номер ряда поля с препятствием

номерКолонки – номер колонки поля с препятствием

2. **public void** установитьПутника(**int** номерРяда, **int** номерКолонки, **ориентацияПутника** ориентация)

Общедоступный метод установки исполнителя Путник (начальное положение) в клетку поля с заданными координатами

где *номерРяда* – номер ряда поля с препятствием

номерКолонки – номер колонки поля с препятствием

ориентация – значение перечисляемого типа *ориентацияПутника* со множеством значений : **наСевер**, **наВосток**, **наЮг**, **наЗапад**

3. **public void** пометитьКлетку(**int** номерРяда, **int** номерКолонки, **char**[] символ)

Общедоступный метод установки меток в клетке поля с заданными координатами

где *номерРяда* – номер ряда поля

номерКолонки – номер колонки поля

символ – символ-массив пометки клетки (допускается одновременная разметка клетки символами в нескольких местах)¹.

4. **public void** закраситьКлетку(**int** номерРяда, **int** номерКолонки, **гаммаЦветов** цвет)

Общедоступный метод закраски клетки поля с заданными координатами

где **номерРяда** – номер ряда поля закраски

номерКолонки – номер колонки поля закраски

цвет – значение перечисляемого типа **гаммаЦветов** со множеством значений : белый, красный, оранжевый, желтый, зеленый, голубой, синий, фиолетовый

2.3 Команды обратной связи конструктора и свойства

5. **private bool** координатыКорректны(**int** ряд, **int** колонка)

Внутренний метод анализа корректности координат, переданных в методы с адресуемыми параметрами

где **ряд** – анализируемый номер ряда

колонка – анализируемый номер колонки

true – координаты корректны и **false** – координаты некорректны

6. **public bool** естьПрепятствие(**int** номерРяда, **int** номерКолонки)

Общедоступный метод выдающий информацию о наличии/отсутствии препятствия в клетке поля с указанными координатами

где **номерРяда** – номер ряда

номерКолонки – номер колонки

true – препятствие есть и **false** – препятствия нет

7. **public Поле** обстановка(**out положениеПутника** начПоложение)

Общедоступный метод с дополнительным выходным параметром, выдающий информацию о ситуации на поле и о начальном местоположении исполнителя Путник

где **начПоложение** – выходной параметр структурного типа **положениеПутника**, позволяющий получить информацию о координатах начального положение Путника и о ориентации Путника

Поле – информация о множестве клеток поля – исходной среды обитания исполнителя Путник

8. **public int** количествоКолонок

Общедоступное свойство, выдающего информацию о ширине поля²

¹ - 0-коорд. - верхняя метка клетки, доступная для обработки Путником
- 1-коорд. - нижняя метка клетки, исключительно для поясняющей разметки
- 2-коорд. - символ-точка для пометки клетки, как конечного положения Путника
² Заметим, что нумерация колонок ведется от 0.

9. `public int` количествоРядов

Общедоступное свойство, выдающего информацию о высоте поля³

2.4 Команда завершения работы конструктора

1. `public void` работаЗавершена()

Метод-регистратор сформированной среды Путника

2.5 Пример оформления решения строителя среды

1. Во-первых, заметим, метод построения среды оформляется в рамках управляющего класса *построительСред.cs* в одном из его вспомогательных классов типа *построительСредыПоТемеОператорWhile.cs*.

2. Во-вторых, во вспомогательном классе организуется внутренний метод⁴

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using библиотекаИсполнителяПутник;

namespace библиотекаСтроителяСред
{
    /// <summary>
    /// Класс методов построения сред исполнителя Путник для группы задач 1.
    /// Группа 1 : Задачи на использование оператора while и методов Путник
    /// </summary>
    public class построительСредыПоТемеОператорWhile
    {
        Random случайноеЧисло = new Random();
        private редакторПоля редактор;
        положениеПутника начальноеПоложениеПутника;
        Поле обстановка;
        . . . . .
        public Поле построительСредыWhile(int номерЗадачи,
            out положениеПутника начальноеПоложениеПутника)
        {
            switch (номерЗадачи)
            {
                case 1:
                    тестЗадачи_01();
                    return редактор.обстановка(out начальноеПоложениеПутника);
                case 2:
```

³ Заметим, что нумерация рядов ведется от 0.

⁴ Рекомендуется давать методу имя, включающее номер задачи, а перед методом вставлять комментарии с полным текстом задачи.

```

        //тестЗадачи_02();
        return редактор.обстановка(out начальноеПоложениеПутника);
        . . . . .
    }
}
/*
 *   Задача 01
 *   Где-то перед Путником одиночное препятствие, а за ним - край поля.
 *   Закрашен ряд клеток между начальным положением Путника и стеной,
 *   к которой он обращен в синий цвет.
 *   Путник - в исходном положении, с ориентацией противоположной
 *   начальной ориентации.
 */
private void тестЗадачи_01()
{
    редактор = new редакторПоля((int)случайноеЧисло.Next(10, 16),
        (int)случайноеЧисло.Next(10, 34));

    int ширина = редактор.количествоКолонок;
    int высота = редактор.количествоРядов;
    int кодОриентации = (int)случайноеЧисло.Next(0, 5);
    switch (кодОриентации)
    {
        case 0: // Ориентация на Север - препятствие сверху
            дляОриентацииНаСевер(высота, ширина);
            break;
        case 1: // Ориентация на Восток - препятствие справа
            дляОриентацииНаВосток(высота, ширина);
            break;
        case 2: // Ориентация на Юг - препятствие снизу
            дляОриентацииНаЮг(высота, ширина);
            break;
        default: // Ориентация на Запад - препятствие слева
            дляОриентацииНаЗапад(высота, ширина);
            break;
    }

    редактор.работаЗавершена();
    обстановка = редактор.обстановка(out начальноеПоложениеПутника);
    /*
        for (int j = колНачПоложенияРобота; j < ширина; j++)
            редактор.закрасить(j, рядСоСтеной);
        редактор.позначить(колНачПоложенияРобота, рядСоСтеной, 2, '1');
        редактор.кончитьРаботу("Pattern.out");
    */
}
private void дляОриентацииНаСевер(int колРядов, int колКолонок)
{

```

```

int рядПрепятствия = случайноеЧисло.Next(колРядов / 2, колРядов - 3);
int колонка_с_Препятствием = случайноеЧисло.Next(2, колКолонок - 2);
int рядНачПоложенияПутника = случайноеЧисло.Next(рядПрепятствия + 1,
    колРядов - 1);
редактор.установитьПутника(рядНачПоложенияПутника,
    колонка_с_Препятствием, ориентацияПутника.наСевер);
редактор.поставитьПрепятствие(рядПрепятствия, колонка_с_Препятствием);
char[] разметка = { ' ', 'Z', ' ' };
for (int j = 0; j < рядНачПоложенияПутника; j++)
    if (j != рядПрепятствия)
    {
        редактор.пометитьКлетку(j, колонка_с_Препятствием, разметка);
    }
char[] разметкаНачала = { ' ', 'H', ' ' };
редактор.пометитьКлетку(рядНачПоложенияПутника, колонка_с_Препятствием,
    разметкаНачала);
}
private void дляОриентацииНаВосток(int колРядов, int колКолонок)
{
    int ряд_с_Препятствием = случайноеЧисло.Next(2, колРядов - 2);
    int колонкаПрепятствия = случайноеЧисло.Next(колКолонок / 2, колКолонок - 3);
    int колНачПоложенияПутника = случайноеЧисло.Next(0, колонкаПрепятствия - 1);
    редактор.установитьПутника(ряд_с_Препятствием, колНачПоложенияПутника,
        ориентацияПутника.наВосток);
    редактор.поставитьПрепятствие(ряд_с_Препятствием, колонкаПрепятствия);

    char[] разметка = { ' ', 'Z', ' ' };
    for (int j = колНачПоложенияПутника; j < колКолонок; j++)
        if (j != колонкаПрепятствия)
        {
            редактор.пометитьКлетку(ряд_с_Препятствием, j, разметка);
        }
    char[] разметкаНачала = { ' ', 'H', ' ' };
    редактор.пометитьКлетку(ряд_с_Препятствием, колНачПоложенияПутника,
        разметкаНачала);
}
private void дляОриентацииНаЮг(int колРядов, int колКолонок)
{
    int рядПрепятствия = случайноеЧисло.Next(2, колРядов / 2);
    int колонка_с_Препятствием = случайноеЧисло.Next(2, колКолонок - 2);
    int рядНачПоложенияПутника = случайноеЧисло.Next(0, рядПрепятствия - 2);
    редактор.установитьПутника(рядНачПоложенияПутника, колонка_с_Препятствием,
        ориентацияПутника.наЮг);
    редактор.поставитьПрепятствие(рядПрепятствия, колонка_с_Препятствием);
    char[] разметка = { ' ', 'Z', ' ' };
    for (int j = рядНачПоложенияПутника; j < колРядов; j++)
        if (j != рядПрепятствия)

```

```

        {
            редактор.пoмeтитьКлетку(j, колонка_с_Препятствием, разметка);
        }
        char[] разметкаНачала = { ' ', 'H', ' ', ' ' };
        редактор.пoмeтитьКлетку(рядНачПоложенияПутника, колонка_с_Препятствием,
            разметкаНачала);
    }
    private void дляОриентацииНаЗапад(int колРядов, int колКолонок)
    {
        int ряд_с_Препятствием = случайноеЧисло.Next(2, колРядов - 2);
        int колонкаПрепятствия = случайноеЧисло.Next(2, колКолонок / 2);
        int колНачПоложенияПутника = случайноеЧисло.Next(колонкаПрепятствия + 2,
            колКолонок - 1);
        редактор.уcтанoвитьПутника(ряд_с_Препятствием, колНачПоложенияПутника,
            ориентацияПутника.наЗапад);
        редактор.пocтaвитьПрепятствие(ряд_с_Препятствием, колонкаПрепятствия);

        char[] разметка = { ' ', 'Z', ' ', ' ' };
        for (int j = 0; j < колНачПоложенияПутника; j++)
            if (j != колонкаПрепятствия)
            {
                редактор.пoмeтитьКлетку(ряд_с_Препятствием, j, разметка);
            }
        char[] разметкаНачала = { ' ', 'H', ' ', ' ' };
        редактор.пoмeтитьКлетку(ряд_с_Препятствием, колНачПоложенияПутника,
            разметкаНачала);
    }
}
}
}

```