1 Пакеты MATLAB

При разработке сложных проектов в MATLAB часто возникает необходимость объединять связанные функции и классы в логические группы, а также избегать конфликтов имён. Пакеты (папки с префиксом +) решают эти задачи, предоставляя изолированное пространство имён.

2 Создание пакета

Пакет представляет собой папку, имя которой начинается с символа +. Например, для создания пакета `MyMath` необходимо создать папку `+MyMath`. Внутри неё могут находиться:

- `.m`-файлы с функциями,
- определения классов (`classdef`),
- другие вложенные пакеты (папки с +).

Листинг 1. Структура пакета `MyMath`

```
+MyMath/
    add.m
    subtract.m
    multiply.m
    +Private/
        helper.m
```

Файл `+MyMath/add.m`:

```
function result = add(a, b)
    result = a + b;
end
```

Файл `+MyMath/subtract.m`:

```
function result = subtract(a, b)
    result = a - b;
end
```

Вложенный пакет `+Private` обычно используется для хранения вспомогательных (приватных) функций, которые не должны вызываться извне. Например, `+MyMath/+Private/helper.m`:

```
function h = helper(x)
    % Приватная вспомогательная функция
    h = x * 2;
end
```

3 Использование пакета

3.1 Квалифицированные имена

Доступ к элементам пакета осуществляется через точечную нотацию:

Листинг 2. Вызов функций пакета

```
>> MyMath.add(5, 3)
ans =
     8

>> MyMath.subtract(10, 4)
ans =
     6
```

3.2 Импорт пакета

Команда `import` позволяет использовать имена функций без указания пакета. Импорт действует в пределах текущей рабочей области (функции, скрипта, командного окна).

Листинг 3. Импорт всего пакета

```
>> import MyMath.*
>> add(5, 3)
ans = 8
```

Можно импортировать только отдельные функции:

```
>> import MyMath.add
>> add(2, 2)
ans =
     4
>> subtract(5, 1)  % Ошибка, subtract не импортирована
```

3.3 Приватные функции

Функции из вложенного пакета `+Private` недоступны напрямую извне, но могут вызываться внутри других функций того же пакета.
Например, `+MyMath/multiply.m`:

```
function result = multiply(a, b)
    result = a * b + Private.helper(a);
end
```

Вызов `MyMath.multiply(3,4)` будет использовать приватную функцию `helper`.

4 Классы внутри пакета

Пакеты могут содержать определения классов. Рассмотрим класс `Calculator` в пакете `MyMath`:

Листинг 4. +MyMath/Calculator.m

```
classdef Calculator
    methods
        function result = add(obj, a, b)
            result = a + b;
        end
        function result = multiply(obj, a, b)
            result = a * b;
        end
    end
end
```

Использование:

```
>> calc = MyMath.Calculator;
>> calc.add(10, 20)
ans =
    30
```

5 Вложенные пакеты

Пакеты могут содержать другие пакеты, что позволяет строить иерархические пространства имён. Например:

```
+MyMath/
    +Geometry/
        area.m
        +Private/
            const.m
```

Тогда для вызова функции `area` из подпакета `Geometry` необходимо использовать `MyMath.Geometry.area()`.