Комментарии в коде

Привет!

В процессе разработки программного обеспечения часто бывает необходимо временно убрать фрагмент кода или написать к коду пояснение, дополнительную информацию. Это можно сделать с помощью комментариев. Рассмотрим, как они работают.

Одиночные комментарии

00:19-01:40

Вы написали код и поняли, что хотите переписать его часть. Чтобы код был перед глазами, но не выполнялся, его можно закомментировать. Закомментировать одну строку кода можно двойным слешем:

```
//Some comment
System.out.println("Message");
```

Если вы хотите закомментировать сразу несколько строк кода, можно их выделить и воспользоваться сочетанием клавиш Ctrl + / или \mathbb{H} + /.

Раскомментировать код можно так же: выделить его, затем нажать на Ctrl + / или \mathbb{H} + /, и все двойные слеши в начале каждой строки кода будут убраны.

С помощью двойного слеша можно писать текстовые комментарии к коду. Принято комментировать код сверху или справа:

```
int cappucinoCoffeeRequired = 60; //In grams
```

Комментарии TODO

01:40-02:35

Вторая разновидность комментариев — это комментарии TODO. Например, вы писали код и подумали, что хорошо бы его потом доработать или что-то в нём проверить. В этом случае вы можете написать комментарий с двумя слешами и словом TODO:

//TODO: check another types of coffee!!!

Такой комментарий будет подсвечен средой разработки, а все комментарии TODO в проекте собираются в среде разработки на специальной вкладке.

Многострочные комментарии

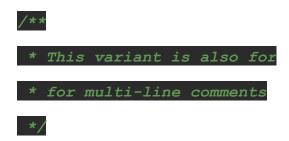
02:35-03:30

Теперь разберёмся, как писать многострочные комментарии. Например, вы хотите написать большой текст с описанием алгоритма работы кода над фрагментом. Это можно сделать с помощью многострочных комментариев. Пишутся они так:

```
/*
Многострочный комментарий с какой-то информацией */
```

Начинаются они со слеша и звёздочки, заканчиваются звёздочкой и слешем. Но их удобнее писать так: сначала слеш, потом две звёздочки, а потом Enter. В этом случае комментарий будет подсвечен в среде разработки, и из таких комментариев можно автоматически сформировать документацию к программному коду.

Есть специальный инструмент для генерации такой документации, он называется Javadoc.



Комментарии в коде бывают:

• Однострочными (состоящими из одной строки). Их принято писать над кодом, который вы комментируете, либо справа от него — это тоже допустимо, если комментарий короткий. Однострочный комментарий может быть помечен словом ТОДО и стать напоминанием что-либо сделать по этому коду. Также с помощью однострочных комментариев можно временно убирать ненужные фрагменты кода. Это можно делать как с одной строкой кода, так и с несколькими, используя сочетания

клавиш Ctrl / или \mathbb{H} I.

Правила написания комментариев

04:52-06:24

Эти правила использования комментариев делают код более читаемым и поддерживаемым.

1. Комментариев в коде должно быть минимум. В идеале не должно быть совсем. Код должен быть написан так, чтобы он был понятен без чтения комментариев.

Часто при изменении кода комментарии поменять забывают, в итоге может стать непонятно, где правильно, в комментарии или в коде. На этапе обучения вы можете комментировать непонятные фрагменты кода, но потом от этой практики необходимо будет отказаться в пользу написания такого кода, который будет достаточно понятен.

```
//Устанавливаем значение id = 546 int id = 546;
```

Такого рода комментарии писать не следует даже во время обучения. Комментарии, дублирующие то, что написано в коде, говорят о том, что автор совсем не понимает код.

2. Допускается временное комментирование кода в процессе работы. Такой код не стоит оставлять после завершения. Его нужно либо раскомментировать, либо удалить совсем.

Со временем в проекте может накопиться очень большое количество ненужного закомментированного кода, в нём легко запутаться и некогда убирать, особенно при работе в команде.

- 3. Комментарии рекомендуется писать на английском языке.
- 4. Комментарии должны быть максимально краткими и чёткими, только по

делу, без рассуждений и лишних слов.

Итоги

с 06:30 до конца

Вы узнали, какими бывают комментарии и как ими правильно пользоваться при разработке программного обеспечения на Java.