Пошаговое объяснение для определения понятия "целое\_число" в восьмеричном представлении в форме БНФ (форма Бэкуса-Наура):

1. <целое\_число> ::= <восьмеричный\_префикс> <восьмеричные\_цифры> - Это означает, что целое число в восьмеричном представлении состоит из восьмеричного префикса и последующих восьмеричных цифр.

2. <восьмеричный\_префикс> ::= "0o" - Здесь определяется, что восьмеричное число начинается с префикса "0o".

3. <восьмеричные\_цифры> ::= <восьмеричная\_цифра> | <восьмеричные\_цифры> <восьмеричная\_цифра> - Восьмеричные цифры могут состоять из одной восьмеричной цифры или последовательности восьмеричных цифр.

4. <восьмеричная\_цифра> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" - Здесь перечислены все возможные восьмеричные цифры: от 0 до 7.

Таким образом, данная грамматика описывает целое число в восьмеричном представлении, начинающееся с префикса "0o", за которым следует последовательность цифр от 0 до 7.

Пошаговое объяснение для определения понятия "выражение" в языке программирования в форме БНФ (форма Бэкуса-Наура):

1. <выражение> ::= <арифметическое\_выражение> | <логическое\_выражение> - Здесь определяется, что выражение может быть как арифметическим, так и логическим.

2. <арифметическое\_выражение> ::= <терм> {<оператор\_сложения> <терм>} - Арифметическое выражение состоит из терма, за которым может следовать один или несколько операторов сложения и последующих термов.

3. <терм> ::= <фактор> {<оператор\_умножения> <фактор>} - Терм состоит из фактора, за которым может следовать один или несколько операторов умножения и последующих факторов.

4. <фактор> ::= <число> | "(" <арифметическое\_выражение> ")" | <переменная> - Фактор может быть числом, арифметическим выражением в скобках или переменной.

5. <оператор\_сложения> ::= "+" | "-" - Оператор сложения может быть либо плюс, либо минус.

6. <оператор\_умножения> ::= "\*" | "/"- Оператор умножения может быть либо умножить, либо разделить.

7. <логическое\_выражение> ::= <логический\_терм> {<логический\_оператор> <логический\_терм>} - Логическое выражение состоит из логического терма, за которым может следовать один или несколько логических операторов и последующих логических термов.

8. <логический\_терм> ::= <логический\_фактор> | "!" <логический\_фактор> - Логический терм может быть либо логическим фактором, либо отрицанием логического фактора.

9. <логический\_фактор> ::= <арифметическое\_выражение> <сравнение> <арифметическое\_выражение> | "(" <логическое\_выражение> ")" - Логический фактор может быть сравнением двух арифметических выражений или логическим выражением в скобках.

10. <сравнение> ::= "==" | "!=" | "<" | ">" | "<=" | ">=" - Здесь перечислены возможные операторы сравнения: равно, не равно, меньше, больше, меньше или равно, больше или равно.

11. <логический\_оператор> ::= "&&" | "||"- Логический оператор может быть логическим ИЛИ или логическим И.

12. <число> ::= <цифра> {<цифра>} - Число состоит из одной цифры или последовательности цифр.

13. <цифра> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" - Здесь перечислены все возможные цифры: от 0 до 9.

14. <переменная> ::= <буква> {<буква> | <цифра>} - Переменная начинается с буквы и может содержать последовательность букв или цифр.

15. <буква> ::= "a" | "b" | "c" | ... | "z" | "A" | "B" | "C" | ... | "Z" - Здесь перечислены все буквы английского алфавита в верхнем и нижнем регистрах.