|  |  |
| --- | --- |
| **Метрика** | **Краткое описание** |
| *n1* | число уникальных операторов |
| *N1* | общее число всех операторов |
| *n2* | число уникальных операндов |
| *N2* | общее число всех операндов |
| *n* | *n = n1 + n2* – словарь программы |
| *N* | *N = N1 + N2* – длина программы |
| *N'* | – теоретическая длина программы |
| *V* | *V = N \* log2(n)* – объем программы |
| *V'* | *V' = N' \* log2(n')* – теоретический объем программы |
| *n’* | количество переменных, которые вводятся в программу + количество переменных, которые программа выводит |
| *L* | *L = V' / V* – уровень качества программы |
| *λ* | *λ = L / V'* – уровень языка программирования |
| *D* | *D =* ***1*** */ L'* – трудоемкость кодирования программы |
| *I* | *I = V / D* – информационное содержание программы, данная характеристика позволяет определить умственные затраты на создание программы |
| *E* | *E = V / L* – необходимые интеллектуальные усилия |
| *T* | *T = E / S* – время составления программы, (*S = 5..20*) |
| *T'* | оценка времени составления программы |
| *B* | *B = E2/3 / 3000* – число дефектов в программе |
| *B'* | *B' = V / 3000* – оценка числа дефектов в программе |
| *Mх* | *Mх = n’ / 6* – М Холстеда (количество модулей) |
| *Мм* | Цикломатическое число Маккейба. Способы нахождения:   1. На диаграмме деятельности количество ребер – количество узлов (блоков) + 2; 2. Количество условных операторов (if, else, else if, for, switch, while, do..while) + 1. Вложенные условные конструкции нужно также считать (например, **for** внутри **for**). Конструкция **if..else** считается как 1. |