**РЕЦЕНЗИРОВАНИЕ РАВНЫМИ ПО РАНГУ**

ИНСТРУКЦИЯ: Укажите Вашу фамилию или идентификационный номер в поле "Рецензирующий", затем укажите номера рецензируемых программ, после этого ответьте, пожалуйста, на вопросы, указывая оценку по шкале от 1 до 7:

НЕТ: 1, НЕ ЗНАЮ или БЕЗРАЗЛИЧНО: 2...6, ДА: 7

Перед постановкой оценок следует ознакомиться с текстом оцениваемых программ.

**Рецензирующий:** \_\_\_Чепоков\_Елизар\_Сергеевич\_(ПИ-18-2)\_\_\_\_\_\_

Замечание: Приведенные примеры написаны на языке Паскаль. При оценке Вам надо переложить их на язык программирования оцениваемой программы.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| СУБЪЕКТИВНЫЕ ВОПРОСЫ | № программы | | | |
| P20 | P35 | P52 | P58 |
| 1. Программа оформлена в одном стиле. Если в каких-то правилах допускается вариативность, то всегда применяется какой-то один вариант. | 7 | 7 | 7 | 7 |
|  |  |  |  |  |
| **Переменные. Имена (идентификаторы)** |  |  |  |  |
| 1. Любая переменная используется в одном единственном смысле. | 5 | 7 | 7 | 5 |
| 1. Смысл переменных, подпрограмм, типов и пр. отражен в их имени (идентификаторе). | 6 | 7 | 7 | 1 |
| 1. Имя содержит не менее 3 символов. | 3 | 7 | 7 | 3 |
| 1. Имя не совпадает по написанию или звучанию с другими именами (Ol и OI, max и maks, fon и phone) и служебными словами (than, programs). | 7 | 7 | 7 | 7 |
| 1. Не используются комбинации трудноразличимых символов o/0 (буква «о» и нуль), 1/I/l (единица, латинские буквы «и» и «л»): a01 – ao1, I1 – l1. | 7 | 7 | 7 | 7 |
| 1. Для обозначения групп взаимосвязанных переменных используются префиксы и/или постфиксы: TekStud, TekGroup, MaxStud, MaxGroup. | 5 | 7 | 7 | 1 |
| 1. Внутри «сложного» имени, состоящего из нескольких частей, используются большие буквы: TekStud, TekGroup. | 7 | 7 | 7 | 1 |
| 1. Используются стандартные обозначения, которые одинаковы во всех программах. | 1 | 7 | 7 | 7 |
| 1. Имена типов начинаются с буквы «Т большое»: Tstud, TGroup. | 1 | 1 | 1 | 1 |
|  |  |  |  |  |
| **Комментарии** |  |  |  |  |
| 1. Строки с комментариями составляют не менее трети строк программы. | 1 | 3 | 1 | 1 |
| 1. Есть вступительные (начальные) комментарии к программе. | 5 | 1 | 1 | 1 |
| 1. Есть начальные комментарии к подпрограммам (процедурам, функциям, методам). | 7 | 7 | 1 | 1 |
| 1. В начальных комментариях указаны решаемая задача, реализуемый алгоритм, требования к ресурсам, автор, время написания и пр. | 1 | 1 | 1 | 1 |
| 1. Есть комментарии к данным. | 7 | 7 | 1 | 1 |
| 1. Есть комментарии и формальные параметрам. | 6 | 1 | 1 | 1 |
| 1. Комментарии к данным поясняют смысл каждой описанной в программе переменной (независимо от мнемоничности ее идентификатора). | 7 | 7 | 1 | 1 |
| 1. Комментарии к данным выравнены по вертикали, чтобы они начинались с одной позиции: | 2 | 1 | 1 | 1 |
| var Tek: real; {текущий элемент последовательности }  LenLine: integer; {длина строки} | | | | |
| 1. Есть текущие комментарии к алгоритму. | 1 | 7 | 1 | 1 |
| 1. В качестве комментариев к алгоритму записан план программы. составленный при нисходящем программировании. Детализация плана записывается с отступом. | 1 | 1 | 1 | 1 |
| {начать}  {загрузить БД}  Assign(f,’f.dat’);  Reset(f);  {установить параметры отображения БД}  LenLine:= 60;  Razdelit:= ‘\*’;  {инициализировать переменные}  sum:= 0;  proizv:= 1;  {продолжить}  {провести поиск информации}  {провести расчеты}  {закончить}  {распечатать результаты}  {сохранить БД} |  |  |  |  |
|  |  |  |  |  |
| 1. В виде комментариев записаны смысл (цель) конструкция (сформулированный при нисходящем программировании). Например: | 1 | 1 | 1 | 1 |
| while {не конец набора данных}  k <= N do |  |  |  |  |
| 1. Прокомментированы принимаемые в программе решения, неочевидные моменты и пр. | 7 | 7 | 1 | 1 |
| 1. Логические части программы отделяются друг от друга пустыми строками и/или строками с разделителями. Например, перед каждой процедурой оставлена пустая строка, группы взаимосвязанных процедур отчеркнуты строкой-комментарием, состоящей из минусов. | 4 | 7 | 4 | 4 |
| 1. Имена подпрограмм подчеркнуты. Для этого используется строка-комментарий. | 1 | 1 | 1 | 1 |
| 1. После завершающего end’а процедуры в комментарии повторяется ее имя. | 1 | 1 | 1 | 1 |
| procedure SredBall(…);  { -------- Подсчет среднего балла за сессию} …  end; {SredBall} | | | | |
|  |  |  |  |  |
| **Операторы** |  |  |  |  |
| 1. Операторная часть подпрограммы целиком размещается на экране. (На одном экране размещаются begin и end подпрограммы, а, следовательно, и все операторы подпрограммы.) | 7 | 7 | 7 | 7 |
| 1. Если операторная часть подпрограммы не помещается на экране, то за экран выходят только независимые части, исполняемые однократно. Например, инициализация данных и/или распечатка результатов. Недопустимо, когда листание требуется для многократно просматриваемой части программы, типа тела цикла. (Если операторная часть всегда помещается на экране, за этот пункт ставится максимальный балл.) | 7 | 7 | 7 | 7 |
| 1. Тело цикла обязательно целиком помещается на экране. | 7 | 7 | 7 | 7 |
| 1. Каждый оператор начинается в своей строке. | 7 | 7 | 7 | 1 |
| 1. В случае если несколько операторов размещены в одной строке, то это могут быть только простые операторы, и один оператор достаточно хорошо отделен от другого. Например, с помощью нескольких пробелов. (Если каждый оператор начинается на отдельной строке, за этот пункт ставится максимальный балл.) | 7 | 7 | 7 | 7 |
| tek:= tek + step; sum:= sum + tek; |  |  |  |  |
| 1. Для выделения вложенности конструкций (как операторов, так и данных) используются абзацные отступы. | 4 | 7 | 4 | 4 |
| If tek > max then  max:= tek  else if tek < min then  min:= tek;  record  name: string;  age: 0..100;  end; |  |  |  |  |
|  |  |  |  |  |
| 1. Размер отступа – 3 позиции или более. | 1 | 1 | 1 | 5 |
| 1. Все абзацные отступы в программе одинаковы. | 7 | 7 | 7 | 1 |
| 1. В условном операторе всегда записана часть else, даже если на этой ветке ничего не делается: | 1 | 1 | 1 | 1 |
| If C1 then S1  else {ничего} |  |  |  |  |
| 1. В операторе ветвления слово end выравнено по слову case. Варианты записаны с отступом с одного и того же уровня: | 7 | 7 | 7 | 7 |
| case color of  red: S1;  green, blue: S2;  else S3  end |  |  |  |  |
| 1. После закрывающей операторной скобки записан комментарий, указывающий, какую именно конструкцию закрывает данная скобка. | 1 | 1 | 1 | 1 |
|  |  |  |  |  |
| **Вычисления** |  |  |  |  |
| 1. В операторной части не используются литеральные константы (т.е. значения, которые сами себя обозначают: 25, 3.14, ‘\*’, ‘Файла с таким именем не существует’). Всегда используются именованные константы. | 4 | 4 | 4 | 4 |
| const KolStud= 25;  pi= 3.14;  FileError= ‘Файла с таким именем не существует’; |  |  |  |  |
| 1. Отсутствует изменение счетчика цикла с помощью присваивания. | 7 | 7 | 7 | 7 |
| 1. Не используется инициализацию переменных по умолчанию. | 7 | 7 | 7 | 7 |
| 1. В выражении пробелы используются для выделения старшинства операций: | 7 | 7 | 7 | 7 |
| a:= b + c\*d |  |  |  |  |
| 1. Если выражение не помещается на одной строке, то оно прерывается на знаке операции таким образом, что эта строка без продолжения будет содержать синтаксическую ошибку: | 1 | 1 | 1 | 1 |
| if (a<b) and (c<d) and  (x=y) and (z>f) |  |  |  |  |
| 1. Если длинное выражение стоит в операторе присваивания, то после переноса оно продолжается правее знака присваивания: | 1 | 1 | 1 | 1 |
| a:= b + c + d + e + f + g +  h + i + j + k |  |  |  |  |
| 1. В операторе присваивания после знака присваивания перед выражением стоит пробел: | 7 | 7 | 7 | 7 |
| a:= b+c |  |  |  |  |

# Итоговая оценка (запишите номера программ)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Место** | **1 место** | **2 место** | **3 место** | **4 место** |
| Программа | P35 | P20 | P52 | P58 |
| Всего баллов | 210 | 192 | 172 | 140 |
| Средний балл | 4,9 | 4,4 | 4 | 3,3 |
| Округленный балл | 5 | 4 | 4 | 3 |

1. Которая программа является, по Вашему мнению, самой лучшей?

Программа P52

1. Которая программа, по Вашему мнению, хуже всех?

Программа P58

1. Которую программа Вы бы поставили на второе место?

Программа P35

# Вопросы о процессе рецензирования равными по рангу (ответ по шкале от 1 до 7)

1. Узнали ли Вы в ходе рецензирования что-либо полезное о стиле программирования?

Оценка: 7

1. Изменили бы Вы стиль написания Ваших программ, если бы Вам сказали, что рецензирование равными по рангу будет проводиться каждые 6 месяцев?

Оценка: 7

1. Считаете ли Вы, что процесс рецензирования равными по рангу может оказаться эффективным средством улучшения программирования в Вашей организации?

Оценка: 7

1. Считаете ли Вы, что организаторы рецензирования сделали все возможное для сохранения анонимности?

Оценка: 7

1. Ваше мнение по поводу процедуры рецензирования равными по рангу и предложения по ее усовершенствованию?

Оценка: 7

# Предложения по изменению набора вопросов

1. Укажите, пожалуйста, два вопроса, ответ на которые вызвал наибольшие затруднения.

|  |
| --- |
| 37. В операторной части не используются литеральные константы (т.е. значения, которые сами себя обозначают: 25, 3.14, ‘\*’, ‘Файла с таким именем не существует’). Всегда используются именованные константы. |
| 34. В условном операторе всегда записана часть else, даже если на этой ветке ничего не делается: |

1. Если затруднения вызваны нечеткой или некорректной формулировкой этих вопросов, не могли бы Вы предложить для них свою формулировку, более четкую и корректную:

|  |  |
| --- | --- |
| 37. В операторной части не используются литеральные константы (т.е. значения, которые сами себя обозначают: 25, 3.14, ‘\*’, ‘Файла с таким именем не существует’). Всегда используются именованные константы. | 37. Используются ли константы в программе? (Если констант нет, то ставиться минимальный балл.) Если константы есть, то используются ли в операторной части литеральные константы? |
| 34. В условном операторе всегда записана часть else, даже если на этой ветке ничего не делается: | Есть ли в условном операторе часть else (даже если она ничего не делает) |

1. Укажите, пожалуйста, два вопроса, наименее полезные с Вашей точки зрения для оценивания качества программы.

|  |
| --- |
| 41. Если выражение не помещается на одной строке, то оно прерывается на знаке операции таким образом, что эта строка без продолжения будет содержать синтаксическую ошибку: |
| 42. Если длинное выражение стоит в операторе присваивания, то после переноса оно продолжается правее знака присваивания: |

1. Назовите, пожалуйста, темы или вопросы, важные, по Вашему мнению, но недостаточно отраженные в заполненном Вами бланке:

|  |
| --- |
| Доступность в чтении программы для других программистов |
| Разделены ли отдельные части на подпрограммы |
| Есть ли в программе отдельные классы (например: только для текста) |