


МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Тульский государственный университет»

Институт прикладной математики и компьютерных наук
Кафедра «Прикладная математика и информатика»

Утверждено на заседании кафедры
«Прикладная математика и информатика»
« 21 » января 2021 г., протокол № 6

Заведующий кафедрой

 В.И. Иванов

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ (ОЦЕНОЧНЫЕ МАТЕРИАЛЫ) ДЛЯ
ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО
ДИСЦИПЛИНЕ (МОДУЛЮ)**

«Алгоритмы и алгоритмические языки»

**основной профессиональной образовательной программы
высшего образования – программы бакалавриата**

по направлению подготовки

01.03.02 Прикладная математика и информатика

с направленностью (профилем)

Прикладная математика и информатика

Форма обучения: очная

Идентификационный номер образовательной программы: 010302-01-21

Тула 2021 год

ЛИСТ СОГЛАСОВАНИЯ
фонда оценочных средств (оценочных материалов)

Разработчик:

Смирнов О.И., доцент каф. ПМиИ, к.ф.-м.н., доцент

(ФИО, должность, ученая степень, ученое звание)



(подпись)

1 Описание фонда оценочных средств (оценочных материалов)

Фонд оценочных средств (оценочные материалы) включает в себя контрольные задания и (или) вопросы, которые могут быть предложены обучающемуся в рамках текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю). Указанные контрольные задания и (или) вопросы позволяют оценить достижение обучающимся планируемых результатов обучения по дисциплине (модулю), установленных в соответствующей рабочей программе дисциплины (модуля), а также сформированность компетенций, установленных в соответствующей общей характеристике основной профессиональной образовательной программы.

Полные наименования компетенций и индикаторов их достижения представлены в общей характеристике основной профессиональной образовательной программы.

2 Оценочные средства (оценочные материалы) для проведения текущего контроля успеваемости обучающихся по дисциплине (модулю)

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.1)

1. В машине Тьюринга рабочий алфавит:
 - 1) { a0, a1, a2,...,at };
 - 2) {q1, q2, q3, ..., qn};
 - 3) { q1, q2, q3, ..., qn, a0, a1, a2,...,at};
2. В машине Тьюринга состояниями являются:
 - 1) { a0, a1, a2,...,at };
 - 2) {q1, q2, q3, ..., qn};
 - 3) { q1, q2, q3, ..., qn, a0, a1, a2,...,at};
3. В машине Тьюринга предписание L для лентопротяжного механизма означает:
 - 1) переместить ленту вправо;
 - 2) переместить ленту влево;
 - 3) остановить машину;
 - 4) занести в ячейку символ.
4. В машине Тьюринга предписание R для лентопротяжного механизма означает:
 - 1) переместить ленту вправо;
 - 2) переместить ленту влево;
 - 3) остановить машину;
 - 4) занести в ячейку символ.
5. В машине Тьюринга предписание S для лентопротяжного механизма означает:
 - 1) переместить ленту вправо;
 - 2) переместить ленту влево;
 - 3) остановить машину;
 - 4) занести в ячейку символ.

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.2)

1. Чтобы окно пользователя по завершении программы не исчезало с экрана, в конец программы нужно добавить инструкцию:
 - 1) writeln;
 - 2) readln;
 - 3) repeat;

- 4) read;
2. Переменные, используемые в программе, объявляются:
 - 1) в начале программы, после слова Program;
 - 2) в любом месте программы между begin и end;
 - 3) в начале программы, после слова var;
 - 4) в любом месте программы после оператора for.
3. Какое значение будет иметь переменная x после выполнения следующих строк программы: $x:=2$; $y:=x+1$; $x:=y*x$; $y:=y+x$; $x:=x - y$;
 - 1) -1;
 - 2) -4;
 - 3) -3;
 - 4) 5;
 - 5) 3.
4. Выберите правильный вариант вывода на экран значения переменной z:
 - 1) writeln z;
 - 2) writeln ['z'];
 - 3) writeln (z);
 - 4) writeln 'z';
 - 5) writeln ('z');
5. Оператор выбора начинается словом ...
 - 1) case;
 - 2) if;
 - 3) var;
 - 4) else;
 - 5) until.

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.3)

1. Составить алгоритм поиска наибольшего и наименьшего элемента в трехмерном массиве.
2. Составить процедуру умножения матриц.
3. Составить функцию проверки ортогональности квадратной матрицы заданного порядка (скалярное произведение любой пары различных строк равно 0, а скалярное произведение каждой строки на себя равно одному и тому же числу не равному 0).
4. Составить процедуру сортировки элементов одномерного массива по возрастанию и убыванию.
5. Дан массив A размера n, отсортированный по возрастанию, кроме первого элемента. Сделать массив упорядоченным. Вспомогательный массив использовать нельзя.
6. Задан набор из n точек (x, y) в виде двух массивов X и Y размерности n (можно использовать один массив, элементы которого есть записи из двух полей x и y). Задано число R. Написать алгоритм, отбирающий только те точки, которые лежат внутри окружности с центром в начале координат и радиуса R.
7. Описать рекурсивную функцию Root3(x,n), возвращающую результат вещественного типа — приближенное значение корня 3-й степени из числа x по формуле: $y_0 = 1$, $y_{n+1} = y_n - (y_n^3 - x) / (3y_n^2)$, где y_n есть очередное приближение к корню. Параметры функции: $x > 0$ — вещественное, $n > 0$ — целое (количество итераций вычисления).
8. Дана матрица A размера n x n. Вывести на экран верхний треугольник матрицы особым образом: сначала главную диагональ, потом диагональ над ней и т.д.
9. Описать функцию MaxElement(a, n, m), возвращающую максимальный элемент матрицы a размера n x m.
10. Описать функцию MinElement(a, n, m), возвращающую минимальный элемент матрицы a размера m x n.

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-4 (контролируемый индикатор достижения компетенции ОПК-4.1)

1. В алгоритме Маркова ассоциативным исчислением называется:
 - 1) совокупность всех слов в данном алфавите;
 - 2) совокупность всех допустимых систем подстановок;
 - 3) совокупность всех слов в данном алфавите вместе с допустимой системой подстановок;
 - 4) когда все слова в алфавите являются смежными.
2. В алгоритме Маркова дана цепочка P_1, P_2, \dots, P_k . Если слова P_1, P_2, \dots, P_k — смежные, то цепочка называется:
 - 1) ассоциативной;
 - 2) эквивалентной;
 - 3) индуктивной;
 - 4) дедуктивной.
3. В алгоритме Маркова дана цепочка P_1, P_2, \dots, P_k . Если слова P_1, P_2, \dots, P_k — смежные и цепочка существует и в обратную сторону, то слова P_1, P_2, \dots, P_k называют:
 - 1) ассоциативными;
 - 2) эквивалентными;
 - 3) индуктивными;
 - 4) дедуктивными.
4. Инструкции в линейной программе ...
 - 1) выполняются в определенном порядке, в зависимости от некоторого условия;
 - 2) выполняются строго последовательно друг за другом;
 - 3) могут выполняться несколько раз.
5. Как называется графическое представление алгоритма:
 - 1) Псевдокод
 - 2) Блок-схема
 - 3) Таблица
 - 4) Словесное описание

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-4 (контролируемый индикатор достижения компетенции ОПК-4.2)

1. Из следующих утверждений выберите истинное:
 - 1) переменная, используемая в операторе For, может быть типа real;
 - 2) в операторе While проверка условия осуществляется после тела цикла;
 - 3) перед Else всегда ставится символ «;» ;
 - 4) оператором Write производится вывод данных на экран.
2. Что будет выведено на экран в результате выполнения следующих инструкций?
 for i:=1 to 5 do write('*');
 - 1) Будут выведены пять звездочек в одной строке.
 - 2) Будут выведены цифры от 1 до 5 в одной строке.
 - 3) Будут выведены пять звездочек столбиком.
 - 4) Будут выведены цифры от 1 до 5 столбиком.
3. Чему будет равно значение переменной n после выполнения следующих инструкций?
 n:=0; while n<=5 do n:=n+1;
 - 1) 1;
 - 2) 3;
 - 3) 0;
 - 4) 6;

- 5) 5.
4. Для обозначения числа 3.14 нужно использовать переменную типа ...
 - 1) real;
 - 2) integer;
 - 3) string;
 - 4) char.
5. Какой формат имеет цикл ПОКА?
 - 1) if – then – else;
 - 2) for – to – do;
 - 3) while – do;
 - 4) repeat – until.

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-4 (контролируемый индикатор достижения компетенции ОПК-4.3)

1. Описать рекурсивную функцию DigitSum(k), возвращающую результат целого типа — сумму цифр целого числа k.
2. Дан массив A размера n. После каждого отрицательного элемента массива вставить элемент с нулевым значением (не затирая существующие).
3. Дан массив A размера n. Сформировать массив той же размерности, i-й элемент которого содержит количество повторений i-го элемента массива A.
4. Задана строка из n символов. Вывести строку посимвольно в заданном порядке: сначала выводятся элементы 1, 4, 7 и т.д., затем 2, 5, 8 и т.д., а затем 3, 6, 9 и т.д.
5. Описать функцию NegCount(a, n, m), возвращающую целое число — количество отрицательных чисел в массиве a размера n х m и индексы этих чисел.
6. Дан массив A размера n х m. Вывести его содержимое на экран построчно, причем каждая нечетная строка выводится слева направо, а каждая четная — справа налево.
7. Дан массив A размера n и целое k ($k < n$). Разработать алгоритм, осуществляющий циклический сдвиг элементов массива вправо на k позиций. Элемент, сдвигающийся за пределы массива, размещается с первой позиции. Вспомогательный массив использовать нельзя.
8. Дано 200 вещественных чисел. Определить, сколько из них больше своих «соседей», т.е. предыдущего и последующего чисел. Вывести номера таких чисел в последовательности.
9. Составить алгоритм процедуры вычисления суммы и числа положительных элементов каждого столбца прямоугольной матрицы. Определить столбец с максимальной такой суммой.
10. Составить функцию проверки ортогональности квадратной матрицы заданного порядка (скалярное произведение любой пары различных строк равно 0, а скалярное произведение каждой строки на себя равно одному и тому же числу не равному 0).

3 Оценочные средства (оценочные материалы) для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.1)

1. Какой оператор определяет равенство двух значений?
 - 1) =
 - 2) ==
 - 3) =:
 - 4) :=
2. В каком из условных операторов допущена синтаксическая ошибка?
 - 1) if B = 0 then Writeln('Деление на нуль невозможно.');

- 2) if $a > b$ then $\max := a$ else $\max := b$;
- 3) if $(a > b)$ and $(b > 0)$ then $c := a + b$;
- 4) if $a < b$ then $\min := a$; else $\min := b$;
- 3. Какие типы циклов существуют в языке Паскаль?
 - 1) for
 - 2) do...while
 - 3) while
 - 4) repeat...until
 - 5) loop
- 4. Какого раздела не существует в программе, написанной на языке Паскаль?
 - 1) Заголовка
 - 2) Примечания
 - 3) Описаний
 - 4) Операторов
- 5. Структурированная программа — это программа, составленная из фиксированного множества базовых конструкций. Сколько таких конструкций в языке Паскаль?
 - 1) 3
 - 2) 4
 - 3) 5
 - 4) 6
 - 5) 7

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.2)

- 1. В чём особенности стека?
 - 1) он открыт с обеих сторон на вставку и удаление;
 - 2) доступен любой элемент;
 - 3) он открыт с одной стороны на вставку и удаление.
- 2. Какую дисциплину обслуживания принято называть FIFO?
 - 1) стек;
 - 2) очередь;
 - 3) дек.
- 3. Каково правило выборки элемента из стека?
 - 1) первый элемент;
 - 2) последний элемент;
 - 3) любой элемент.
- 4. Сколько указателей используется в односвязных списках?
 - 1) 1
 - 2) 2
 - 3) сколько угодно.
- 5. В чём отличительная особенность динамических объектов?
 - 1) порождаются непосредственно перед выполнением программы;
 - 2) возникают уже в процессе выполнения программы;
 - 3) задаются в процессе выполнения программы.

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.3)

- 1. На вход алгоритма подается массив сортированных целых чисел. Проверить работу алгоритма для разных массивов данных, например, $\{1, 3, 3, 5, 10, 12\}$, $\{2, 2, 4, 4, 4, 4\}$. Указать, что будет возвращено в результате работы алгоритма. Что делает данный алгоритм?

АЛГОРИТМ SortM(A[0..n-1]: массив целых чисел) : целое

i:=0; r:=0;

while i<=n-1 **do**

| k:=1; v:=A[i];

| **while** i+k<=n-1 **and** A[i+k]=v **do**

| | k:=k+1;

| **if** k>r **then**

| | r:=k; Value:=v;

| i:=i+k;

return Value;

2. Проверить работу алгоритма (привести трассировочную таблицу). Указать, что будет напечатано в результате работы алгоритма. Что делает данный алгоритм?

АЛГОРИТМ2

n:=10; a:=1; b:=1;

for j=1 **to** n **do**

| c:=b; b:=a+b; a:=c;

print(b)

3. Проверить работу алгоритма (привести трассировочную таблицу). Указать, что будет напечатано в результате работы алгоритма.

АЛГОРИТМ

for k=1 **to** 10 **do**

| j:=2+k*k; m:=1; n:=1;

| **while** n<=j **do**

| | **if** n=j **then print**(k,m);

| | m:=m+1; n:=m*m*m;

4. Проверить работу алгоритма для произвольного массива целых чисел (например, {4, 3, 4, 1, 5, 2}). Указать, что будет возвращено в результате работы алгоритма. Что делает данный алгоритм?

АЛГОРИТМ SBub(A[1..n] : массив целых чисел) : массив целых чисел

for i=1 **to** n-1 **do**

| **for** j=n **to** i+1 **step** -1 **do**

| | **if** A[j]<A[j-1] **then**

| | | c:=A[j];

| | | A[j]:=A[j-1];

| | | A[j-1]:=c;

return (A);

5. Проверить работу алгоритма для произвольного массива целых чисел (например, {60, 35, 81, 98, 14, 47}). Указать, что будет возвращено в результате работы алгоритма. Что делает данный алгоритм?

АЛГОРИТМ CCS(A[0..n-1] : массив целых чисел) : массив целых чисел

for i=0 **to** n-1 **do**

| C[i]:=0;

for i=0 **to** n-2 **do**

| **for** j=i+1 **to** n-1 **do**

| | **if** A[i]<A[j] **then**

| | | C[j]:=C[j]+1;

| | **else**

| | | C[i]:=C[i]+1;

for i=0 **to** n-1 **do**

| S[C[i]]:=A[i]

return S;

6. Проверить работу алгоритма для произвольного массива целых чисел (например, {4, 3, 4, 1, 5, 2}). Указать, что будет возвращено в результате работы алгоритма. Что делает данный алгоритм?

АЛГОРИТМ SBub(A[1..n] : массив целых чисел) : массив целых чисел

```

for i=1 to n-1 do
    | for j=n to i+1 step -1 do
    |     | if A[j]<A[j-1] then
    |     |     | c:=A[j];
    |     |     | A[j]:=A[j-1];
    |     |     | A[j-1]:=c;
return (A);

```

7. Проверить работу алгоритма для произвольного массива целых чисел. Указать, что будет возвращено в результате работы алгоритма. Что делает данный алгоритм?

АЛГОРИТМ SB(A[1..n] : массив целых чисел) : целое число

```

mm:=0
k:=0
for i=1 to n-1 do
    | if A[i]=0 then
    |     | k:=k+1;
    | else
    |     | if k>mm then
    |     |     | mm:=k
    |     | k:=0
if k>mm then
    | mm:=k
return (mm);

```

8. Проверить работу алгоритма для разных значений m и n ($m \geq 0$, $n \geq 0$). Указать, что будет возвращено в результате работы алгоритма. Что делает данный алгоритм?

АЛГОРИТМ gcc(m,n: целые) : целое

```

if n=0 then return m;
else
    | if m<n then r=gcc(n,m);
    | else       r=gcc(m-n,n);
    | return r;

```

9. Проверить работу алгоритма для разных значений n ($n \geq 2$). Указать, что будет возвращено в результате работы алгоритма. Что делает данный алгоритм?

АЛГОРИТМ Sieve(n: целое) : массив целых чисел

```

for p=2 to n do
    | A[p]:=p;
for p=2 to  $\lfloor \sqrt{n} \rfloor$  do
    | if A[p]≠0 then
    |     | j:=p*p;
    |     | while j<=n do
    |     |     | A[j]:=0; j:=j+p;
i:=0;
for p=2 to n do
    | if A[p] ≠0 then
    |     | L[i]:=A[p]; i:=i+1;
return L;

```

10. Проверить работу алгоритма для разных значений n и k ($n \geq 0$, $k \geq 0$). Указать, что будет возвращено в результате работы алгоритма. Что делает данный алгоритм?

```

АЛГОРИТМ P(n,k: целые) : целое
if n<k then c:=0;
else
    | if k=0 then c:=1;
    | else
    | | if k=n then c:=1;
    | | else      c:=P(n-1,k-1)+P(n-1,k);
return c;

```

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-4 (контролируемый индикатор достижения компетенции ОПК-4.1)

1. Структура данных представляет собой
 - 1) набор правил и ограничений, определяющих связи между отдельными элементами и группами данных;
 - 2) набор правил и ограничений, определяющих связи между отдельными элементами данных;
 - 3) набор правил и ограничений, определяющих связи между отдельными группами данных;
 - 4) некоторую иерархию данных.
2. Линейный список, в котором доступен только последний элемент, называется
 - 1) стеком;
 - 2) очередью;
 - 3) деком;
 - 4) массивом;
 - 5) кольцом.
3. Структура данных, работа, с элементами которой организована по принципу FIFO (первый пришел — первый ушел) это —
 - 1) Стек;
 - 2) Дек;
 - 3) Очередь;
 - 4) Список.
4. Линейный последовательный список, в котором включение исключение элементов возможно с обоих концов, называется
 - 1) стеком;
 - 2) очередью;
 - 3) деком;
 - 4) кольцевой очередью.
5. В чём особенности очереди?
 - 1) она открыта с обеих сторон;
 - 2) она открыта с одной стороны на вставку и удаление;
 - 3) доступен любой элемент.

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-4 (контролируемый индикатор достижения компетенции ОПК-4.2)

1. В памяти ЭВМ бинарное дерево удобно представлять в виде:
 - 1) связанных линейных списков;
 - 2) массивов;
 - 3) связанных нелинейных списков.
2. Элемент дерева, на который нет ссылок, называется
 - 1) корнем;

- 2) промежуточным;
 - 3) терминальным (листом).
3. Дерево называется полным бинарным, если степень исходов вершин равна:
- 1) 2 или 0;
 - 2) 2;
 - 3) M или 0;
 - 4) M.
4. Даны три условия окончания просеивания при сортировке прямым включением. Найдите среди них лишнее.
- 1) найден элемент $a(i)$ с ключом, меньшим чем ключ u ;
 - 2) найден элемент $a(i)$ с ключом, большим чем ключ u ;
 - 3) достигнут левый конец готовой последовательности.
5. Что из перечисленных ниже понятий является одним из типов сортировки?
- 1) внутренняя сортировка;
 - 2) сортировка по убыванию;
 - 3) сортировка данных;
 - 4) сортировка по возрастанию.

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-4 (контролируемый индикатор достижения компетенции ОПК-4.3)

1. Проверить работу алгоритма для разных значений n ($n \geq 0$, $k \geq 0$). Указать, что будет возвращено в результате работы алгоритма. Что делает данный алгоритм?

```

АЛГОРИТМ F(n,k: целые) : целое
if n<k then c:=0;
else
    | if k=0 then c:=1;
    | else
    |     | if k=n then
    |         | c:=1;
    |         | else c:=F(n-1,k-1)+F(n-1,k);
return c;

```

2. Проверить работу алгоритма для разных значений n ($n \geq 0$, $m \geq 0$). Указать, что будет возвращено в результате работы алгоритма. Что делает данный алгоритм?

```

АЛГОРИТМ gcd(m,n: целые) : целое
if n=0 then return m;
else
    | if m<n then r:=gcd(n,m);
    | else r:=gcd(m-n,n);
    | return r;

```

3. Проверить работу алгоритма для разных массивов данных. Указать, что будет возвращено в результате работы алгоритма. Что делает данный алгоритм?

```

АЛГОРИТМ MDst(n: целое, A[0..n-1]: массив целых чисел) : целое
d:=∞;
for i=0 to n-1 do
    | for j=0 to n-1 do
    |     | if i≠j and |A[i]-A[j]|<d then
    |         | d:=|A[i]-A[j]|;
return d;

```

4. Проверить работу алгоритма. Указать, что будет напечатано в результате работы алгоритма.

```

АЛГОРИТМ
for k=1 to 10 do
    | j:=2+k*k; m:=1; n:=1;
    | while n<=j do
    |     | if n=j then print(k,m);
    |     | m:=m+1; n:=m*m*m;

```

5. Проверить работу алгоритма, задавая массивы целых чисел, с увеличивающейся размерностью. Рекомендуется исходный массив брать сортированным, например, для $n = 3$, $A = \{1, 2, 3\}$ и т.д. Что делает данный алгоритм?

```

АЛГОРИТМ HP(n: целое)
if n=1 then print(A);
else
    | for i=1 to n do
    |     | HP(n-1)
    |     | if n - нечетное then обмениваем A[1] и A[n];
    |     | else обмениваем A[i] и A[n];

```

Примечания. 1) Массив $A[0..n - 1]$ считать заданным глобально. 2) Функция $\text{print}(A)$ выводит на печать весь массив A .

6. Проверить работу алгоритма. Указать, что будет возвращено в результате работы алгоритма. Что делает данный алгоритм?

```

АЛГОРИТМ Grn(x: вещ) : вещ
eps:=1.0E-2;
y:=1.0; z:=0;
while |z-y|>=eps do
    | z:=y;
    | y:=(y+x/y)/2;
return y;

```

7. Проверить работу алгоритма. Что делает данный алгоритм?

```

АЛГОРИТМ FS (n,s)
//A[0..n-1] - массив целых чисел, заданный глобальным
//n - первый раз указывается количество элементов
//s - произвольное целое число
1. j:=-1;
2. if n<0 or s<0 return(-1)
3. if s=0 return(1)
4. for i=n-1 to 0 step -1
    | j:= FS (i , s - a[i])
    | if j > 0
    |     | print (a[i])
    |     | goto 5
5. return (j)

```

8. Проверить работу алгоритма. Указать, что будет возвращено в результате работы алгоритма. Что делает данный алгоритм?

```

АЛГОРИТМ Zr(f: функция, a,b: вещ) : вещ
eps:=1.0E-1;
fa:=f(a); fb:=f(b);
if fa*fb>0 then
    | print("Ошибка");
    | stop;
while |a-b|>=eps do
    | mid:=(a+b)/2; fm:=f(mid);
    | if fa*fm<=0 then
    |     | b:=mid; fb:=fm;
    | else
    |     | a:=mid; fa:=fm;
if |fa|<|fb| then return a;
else return b;
ФУНКЦИЯ f(x: вещ) : вещ
return 2*x-1;

```

9. Проверить работу алгоритма. Указать, что будет возвращено в результате работы алгоритма. Что делает данный алгоритм?

```

АЛГОРИТМ FFum(n,x)
//A[0..n-1] - массив целых чисел, заданный глобальным
//n - первый раз указывается количество элементов
//x - произвольное целое число
1. j:=-1;
2. if n<0 or x<0 return(-1)
3. if x=0 return(1)
4. for i=n-1 to 0 step -1
    | j:=FFum(i, x - a[i])
    | if j > 0
    |     | print (a[i])
    |     | goto 5
4. return (j)

```

10. Проверить работу алгоритма. Что делает данный алгоритм?

```

АЛГОРИТМ Ggn(x: вещ) : вещ
eps:=1.0E-2;
y:=1.0; z:=0;
while |z-y|>=eps do
    | z:=y;
    | y:=(y+x/y)/2;
return y;

```

4 Оценочные средства (оценочные материалы) для проведения промежуточной аттестации обучающихся (защиты курсовой работы (проекта)) по дисциплине (модулю)

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.1)

1. Сколько сравнений и перестановок элементов требуется в пузырьковой сортировке?
 - 1) $n \cdot \log(n)$;
 - 2) $(n \cdot n)/4$;
 - 3) $(n \cdot n - n)/2$.

2. Сколько дополнительных переменных нужно в пузырьковой сортировке помимо массива, содержащего элементы?
 - 1) 0 (не нужно);
 - 2) всего 1 элемент;
 - 3) n переменных (ровно столько, сколько элементов в массиве).
3. Как рассортировать массив быстрее, пользуясь пузырьковым методом?
 - 1) одинаково;
 - 2) по возрастанию элементов;
 - 3) по убыванию элементов.
4. Массив сортируется “пузырьковым” методом. За сколько проходов по массиву самый “лёгкий” элемент в массиве окажется вверху?
 - 1) за 1 проход;
 - 2) за $n-1$ проходов;
 - 3) за n проходов, где n — число элементов массива.
5. Где эффективен линейный поиск?
 - 1) в списке;
 - 2) в массиве;
 - 3) в массиве и в списке.

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.2)

1. Какой поиск эффективнее?
 - 1) линейный;
 - 2) бинарный;
 - 3) без разницы.
2. В чём суть бинарного поиска?
 - 1) нахождение элемента массива x путём деления массива пополам каждый раз, пока элемент не найден;
 - 2) нахождение элемента x путём обхода массива;
 - 3) нахождение элемента массива x путём деления массива.
3. Как расположены элементы в массиве бинарного поиска?
 - 1) по возрастанию;
 - 2) хаотично;
 - 3) по убыванию.
4. Элемент дерева, который не ссылается на другие, называется
 - 1) корнем;
 - 2) листом;
 - 3) узлом;
 - 4) промежуточным.
5. Элемент дерева, на который не ссылаются другие, называется
 - 1) корнем;
 - 2) листом;
 - 3) узлом;
 - 4) промежуточным.

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.3)

1. Алгоритмы. Основные понятия и свойства. Примеры классических алгоритмов.
2. Основные конструкции алгоритмов. Блок-схемы.
3. Способы описания алгоритмов. Примеры.

4. История развития теории алгоритмов.
5. Основные принципы анализа трудоемкости алгоритмов.
6. Принципы анализа временной эффективности алгоритма. Пример.
7. Трудоемкость конструкций «следование», «ветвление», «цикл». Примеры.
8. Определение трудоемкости задачи поиска максимума в массиве.
9. Переход от функции трудоемкости к оценкам времени работы алгоритма.
10. Принципы пооперационного временного анализа.

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-4 (контролируемый индикатор достижения компетенции ОПК-4.1)

1. Элемент дерева, который имеет предка и потомков, называется
 - 1) корнем;
 - 2) листом;
 - 3) узлом;
 - 4) промежуточным.
2. Высотой дерева называется
 - 1) максимальное количество узлов;
 - 2) максимальное количество связей;
 - 3) максимальное количество листьев;
 - 4) максимальная длина пути от корня до листа.
3. Дерево называется бинарным, если
 - 1) количество узлов может быть либо пустым, либо состоять из корня с двумя другими бинарными поддеревьями;
 - 2) каждый узел имеет не менее двух предков;
 - 3) от корня до листа не более двух уровней;
 - 4) от корня до листа не менее двух уровней.
4. Бинарное дерево можно представить
 - 1) с помощью указателей;
 - 2) с помощью массивов;
 - 3) с помощью индексов;
 - 4) правильного ответа нет.
5. Как называются предки узла, имеющие уровень на единицу меньше уровня самого узла
 - 1) детьми;
 - 2) родителями;
 - 3) братьями.

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-4 (контролируемый индикатор достижения компетенции ОПК-4.2)

1. Стандартным способом устранения рекурсии при поиске в глубину является использование
 - 1) массива;
 - 2) очереди;
 - 3) стека;
 - 4) циклического списка.
2. Граф — это
 - 1) нелинейная структура данных, реализующая отношение «многие ко многим»;
 - 2) линейная структура данных, реализующая отношение «многие ко многим»;
 - 3) нелинейная структура данных, реализующая отношение «многие к одному»;
 - 4) нелинейная структура данных, реализующая отношение «один ко многим»;
 - 5) линейная структура данных, реализующая отношение «один ко многим».

3. Узлам (или вершинам) графа можно сопоставить:

- 1) отношения между объектами;
- 2) объекты;
- 3) связи;
- 4) типы отношений;
- 5) множества.

4. Рёбрам графа можно сопоставить:

- 1) связи;
- 2) типы отношений;
- 3) множества;
- 4) объекты;
- 5) отношения между объектами.

5. Граф, содержащий только ребра, называется.

- 1) ориентированным;
- 2) неориентированным;
- 3) простым;
- 4) смешанным.

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-4 (контролируемый индикатор достижения компетенции ОПК-4.3)

1. Сущность рекурсии. Блок-схема работы простой рекурсивной процедуры. Пример.
2. Сложная рекурсия. Пример.
3. Оценка трудоемкости рекурсивного алгоритма. Пример.
4. Основные классы эффективности алгоритмов.
5. Линейные структуры данных.
6. Стек, очередь: определение, представления, реализация операций.
7. Хеширование. Методы разрешения коллизий.
8. Машина Тьюринга. Схема построения.
9. Односвязные и двусвязные списки.
10. Интуитивное определение алгоритмов, свойства алгоритмов. Алгоритмические языки.