

Пояснительная записка

Фонд оценочных средств (далее – ФОС) для промежуточной аттестации по дисциплине «Программирование» разработан в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по направлению подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки), утвержденного приказом Министерства образования и науки Российской Федерации от 22 февраля 2018 г. № 125, основной профессиональной образовательной программой «Математика» и «Информатика» с учетом требований профессионального стандарта «Педагог (педагогическая деятельность в сфере дошкольного, начального общего, основного общего, среднего общего образования) (воспитатель, учитель)», утвержденного приказом Министерства труда и социальной защиты Российской Федерации от 18 октября 2013 г. № 544н. (зарегистрирован Министерством юстиции Российской Федерации 6 декабря 2013 г., регистрационный № 30550), с изменениями, внесенными приказами Министерства труда и социальной защиты Российской Федерации от 25 декабря 2014 г. № 1115н (зарегистрирован Министерством юстиции Российской Федерации 19 февраля 2015 г., регистрационный № 36091) и от 5 августа 2016 г. № 422н (зарегистрирован Министерством юстиции Российской Федерации 23 августа 2016 г., регистрационный № 43326).

Цель ФОС для промежуточной аттестации – установление уровня сформированности компетенции части компетенции УК-1.

Задачи ФОС для промежуточной аттестации - контроль качества и уровня достижения результатов обучения по формируемым в соответствии с учебным планом компетенциям:

Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач (УК-1)

Знает: этапы решения задачи на компьютере; способы формализации алгоритмов на языках программирования высокого уровня; методы процедурного, объектно-ориентированного, функционального и визуального программирования; основные конструкции языков программирования; основные типы данных и операторы; приемы оптимизации алгоритмов по памяти и времени; основные виды ошибок, возникающих при решении задачи.

Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы; выполнять оценку сложности алгоритмов, проводить анализ и оценивание полученных результатов.

Требование к процедуре оценки:

Помещение: нет особых требований

Оборудование: не предусмотрено

Инструменты: не предусмотрено

Расходные материалы: лист бумаги, ручка

Доступ к дополнительным справочным материалам: не предусмотрен.

Нормы времени: 45 мин.

Комплект оценочных средств для проведения промежуточной аттестации

Проверяемая компетенция:

Универсальная компетенция УК-1.

Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач.

Проверяемый индикатор достижения компетенции:

УК-1.1: анализирует задачу, выделяя этапы ее решения, действия по решению задачи.

Проверяемые результаты обучения:

Знает: этапы решения задачи на компьютере.

Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи.

Проверяемый индикатор достижения компетенции: УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения

Проверяемые результаты обучения:

Знает: способы формализации алгоритмов на языках программирования высокого уровня.

Умеет: осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи.

Проверяемый индикатор достижения компетенции:

УК-1.3. Рассматривает различные варианты решения задачи, оценивает их преимущества и риски

Проверяемые результаты обучения:

Знает: методы процедурного, объектно-ориентированного, функционального и визуального программирования; основные конструкции языков программирования; основные типы данных и операторы; приемы оптимизации алгоритмов по памяти и времени.

Умеет: строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня.

Проверяемый индикатор достижения компетенции:

УК-1.4. Грамотно, логично, аргументированно формирует собственные суждения и оценки; отличает факты от мнений, интерпретаций, оценок в рассуждениях других участников деятельности.

Проверяемые результаты обучения:

Знает: основные виды ошибок, возникающих при решении задачи.

Умеет: комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.

Проверяемый индикатор достижения компетенции:

УК-1.5. Определяет и оценивает практические последствия возможных вариантов решения задачи

Проверяемые результаты обучения:

Умеет: выполнять оценку сложности алгоритмов, проводить анализ и оценивание полученных результатов.

Тип (форма) задания: тестовые задания с выбором одного или нескольких вариантов ответа для выявления порогового уровня сформированности компетенций, практические задания – для выявления продвинутого и высокого уровней.

Пример типовых заданий (оценочные материалы): Тестовые задания по дисциплине «Программирование» для выявления порогового уровня сформированности компетенции УК - 1.

Семестр 1

A1. Какое высказывание наиболее точно определяет понятие «модель»

- a) точная копия оригинала;
- b) оригинал в миниатюре;
- c) образ оригинала с наиболее присущими свойствами;
- d) начальный замысел будущего объекта?

A2. Какой этап решения задач на ЭВМ непосредственно предшествует этапу программирования

- a) постановка задачи;
- b) анализ и исследование задачи, модели;
- c) разработка алгоритма;
- d) тестирование и отладка?

A3. Запись алгоритма на языке конкретного исполнителя – это...

- a) алгоритм;
- b) команда;
- c) программа;
- d) процедура.

A4. Фактические параметры процедуры:

- a) описываются в ее заголовке;
- b) указываются при описании данных в программе;
- c) перечисляются при ее вызове;
- d) нигде не указываются.

A5. Сколько раз будут выполнены операторы тела цикла при выполнении следующего фрагмента программы:

A:=1; N:=0; S:=0;

While A>1/1050 Do

Begin A:=Exp(-N*Ln(2)); S:=S+A; N:=N+1 End;

- a) 11;
- b) 1050;
- c) 10;
- d) 100?

A6. При истинности какого условия последовательность переменных A, B, C не является упорядоченной по возрастанию:

- a) $(A < B) \text{ AND } (\text{NOT}(B \geq C))$;
- b) $\text{NOT} ((A > B) \text{ OR } (B > C))$;
- c) $(A \leq B) \text{ AND } (\text{NOT}(B \geq C))$;
- d) $\text{NOT} ((A \leq B) \text{ AND } (B \leq C))$?

A7. Какое значение будет иметь переменная d после выполнения операторов

d = 0;

If a>b then begin d:=b; d:=d+a end else; d:=d*10;

при a = 1; b = 3:

- a) 0;
- b) 4;
- c) 10;
- d) 30?

A8. Какое значение будет иметь переменная d после выполнения операторов при k=1: d:=1;

case k mod 10 of

2,3,5: d:=2;

1,4: d:=3;

7..9: d:=4

End;

- a) 1;
- b) 2;
- c) 3;
- d) 4?

A9. Какое значение будет иметь переменная y после выполнения следующих фрагментов программы:

var y: real; k: integer;

y:=1; for k:=1 to 3 do y:=y+k; y:=y*10;

- a) 70;
- b) 2230;
- c) 30;
- d) 300?

A10. Какое значение будет иметь переменная y после выполнения следующих фрагментов программы:

Var y: real; k: integer;

y:=0; k:=1; repeat y:=y+1/k; k:=k-1 until k<=1;

- a) 1;
- b) 0;
- c) -1;
- d) 0.5?

A11. Служебное слово VAR в программе на языке Паскаль фиксирует начало раздела программы, содержащего:

- a) операторы;
- b) список меток;
- c) перечень констант;
- d) описание переменных.

A12. В алфавит языка Паскаль не входит служебное слово:

- a) THEN;
- b) BEGIN;

- c) END;
- d) STEP.

A13. Комментарий к тексту программы на языке Паскаль заключается:

- a) в фигурные скобки;
- b) в круглые скобки;
- c) в квадратные скобки;
- d) в апострофы.

A14. В качестве имени в языке Паскаль можно использовать

- a) 1r;
- b) or;
- c) перем;
- d) sum.

A15. Операторы в программе на языке Паскаль отделяются друг от друга:

- a) двоеточием;
- b) пробелом;
- c) запятой;
- d) точкой с запятой.

A16. Чему равен результат выражения $3 - 8 + 21 \text{ div } 3$

- a) 2;
- b) 5;
- c) 12;
- d) 18 ?

A17. Какое из приведенных неравенств верно:

- a) $-4.9876543234 \text{ E}-02 < -0,03$;
- b) $-4.9876543234 \text{ E}-02 < -0,5$;
- c) $-4.9876543234 \text{ E}-02 > -0,03$;
- d) $-4.9876543234 \text{ E}-02 < -0,4$?

A18. Какая из перечисленных операций не относится к логическим:

- a) OR;
- b) MOD;
- c) NOT;
- d) AND?

A19. Определите, какой элемент пропущен в программе, предназначенной для подсчета числа делителей натурального числа N.

Var res, I, N: integer;

Begin Readln(N); res:=0; for I:=1 to N do if ... then res:=res+1; writeln(res); end.

- a) $N \text{ div } 2=0$;
- b) $I \text{ mod } 2 <> 0$;
- c) $N \text{ mod } I=0$;
- d) $N \text{ div } I=1$.

A20. Определите, какой элемент пропущен в программе, предназначенной для подсчета числа делителей натурального числа N.

Var res, I, N: integer;

Begin Readln(N); res:=0; for I:=1 to N do if ... then res:=res+1; writeln(res); end.

- a) $N \text{ div } 2=0$
- b) $I \text{ mod } 2 <> 0$;
- c) $N \text{ mod } I=0$;
- d) $N \text{ div } I=1$.

Оценочный лист к типовому заданию А

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
c	c	c	c	a	d	a	c	a	a
A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
d	d	a	d	d	a	a	b	c	c

Примеры заданий для выявления продвинутого уровня сформированности компетенции УК-1.

B1. Определите, что будет напечатано в результате работы следующего фрагмента программы:

алг

нач

цел k, s

s := 5

k := 0

нц пока k < 15

k := k + 2

s := s + k

кц
 ВЫВОД s
 КОН

B2. Определите, какое число будет напечатано в результате выполнения следующего алгоритма.

```
var a,b,t,M,R: integer;
  Function F (x:integer): integer;
    begin
      F := x*x + 6*x + 10
    end;
begin
  a := -10; b := 10;
  M := a; R := F(a);
  for t := a to b do begin
    if (F(t) < R) then begin
      M := t;
      R := F(t)
    end
  end;
  write(M)
end.
```

B3. Ниже на языке Паскаль записаны две рекурсивные функции F и G. Чему будет равно значение, вычисленное при выполнении вызова F(6)?

```
function F(n: integer): integer;
begin
  if n > 2 then
    F := F(n - 1) + G(n - 2)
  else
    F := n;
end;
function G(n: integer): integer;
begin
  if n > 2 then
    G := G(n - 1) + F(n - 2)
  else
    G := n+1;
end;
```

B4. Ниже записан алгоритм на языке Паскаль. Получив на вход число x, этот алгоритм печатает два числа: L и M. Укажите наименьшее число x, при вводе которого алгоритм печатает сначала 5, а потом 7.

```
var x, L, M: integer;
begin
  readln(x);
  L := 0;
  M := 0;
  while x>0 do
  begin
    M := M + 1;    if x mod 2 <> 0 then
      L := L + 1;
    x := x div 2;
  end;
  writeln(L);
  writeln(M);
end.
```

Оценочный лист к типовому заданию B

B1	B2	B3	B4
77	-3	13	79

Примеры заданий для выявления высокого уровня сформированности компетенции УК-1

C1. Составьте оптимальную по времени и использованию памяти программу, проверяющую можно ли представить число n в виде суммы двух простых слагаемых. Если такое возможно, то выводятся эти числа на экран.

Оценочный лист к типовому заданию C (модельный ответ)

C1. Пример решения программы:

Program C1;

```
var i,n:integer; l:boolean;
function prost (a:integer):boolean;
var j:integer;b:boolean;
begin
b:=true;j:=2;
while (j<=round(sqrt(a))) and b do begin
if a mod j=0 then b:=false;
inc(j);
prost:=b;
end;
end;
begin
writeln ('введите N');
readln(n);
l:=true;
for i:=2 to n div 2 do
if prost(i) and prost(n-i) then begin writeln (i,' ',n-i); l:=false; end;
if l then writeln ('Нельзя представить')
end.
```

Критерии правильного ответа к заданию С1

1. Программа выдает верный ответ как в случае, если число можно представить в виде суммы двух простых слагаемых, так и в противном случае.
2. Ответ «нельзя представить в виде двух простых слагаемых» выводится за циклом (то есть один раз).
3. Проверка числа на «простоту» осуществляется в подпрограмме.
4. Проверка на «простоту» прекращается после нахождения первого делителя числа отличного от двойки.

Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Код контролируемой компетенции (индикаторы)	Наименование оценочного средства	Максимальное количество баллов	Всего баллов	Уровень освоения компетенции (в баллах)		
				Пороговый (56-70%)	Продвинутый (71-85%)	Высокий (86-100%)
УК-1.1	Задание А 1-10	10	10	5-6	7-8	9-10
УК-1.2	Задание А 11-20	10	10	5-6	7-8	9-10
УК-1.3	Задание В 1-2	10	10	5-6	7-8	9-10
УК-1.4	Задание В 3-4	10	10	5-6	7-8	9-10
УК-1.5	Задание С	16	12	9-11	12-13	14-16

Полученное число баллов (29-56) выставляется в графу «Промежуточная аттестация» балльно-рейтинговой карты дисциплины.

Тестовые задания по дисциплине «Программирование»
для выявления порогового уровня сформированности компетенции УК-1
Семестр 2

A1. Структура данных – это

- список данных, в котором порядок элементов списка задан посредством указателей, включенных в их запись;
- множество элементов данных, объединенных и упорядоченных определенным образом;
- представление логической организации данных в виде множества типов записей данных и связей между ними
- список данных, в котором порядок элементов никак не обозначен.

A2. Укажите наиболее полный перечень способов записи алгоритмов

- словесный;
- словесный, графический, псевдокод, программный;
- графический, программный;
- словесный, программный.

A3. Суть такого свойства алгоритма, как дискретность заключается в том, что

- алгоритм всегда состоит из последовательности дискретных шагов;
- для записи алгоритма используются команды, которые входят в систему команд исполнителя;
- алгоритм обеспечивает решение не одной конкретной задачи, а некоторого класса задач;
- при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов и привести к определенному результату.

A4. Алгоритм называется линейным, если

- он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
- последовательность выполнения его команд зависит от истинности тех или иных условий;
- его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;

d) он представим в табличной форме.

A5. Системы программирования

- обеспечивают непосредственное решение пользовательских задач;
- позволяют создавать новые программы на языках программирования;

- c) обеспечивают работу всех аппаратных устройств компьютера и доступ пользователя к ним;
- d) представляют собой совокупность программ, используемых для различных операций с документами.

A6. Переменная в программировании наиболее полно характеризуется

- a) именем;
- b) именем, значением и типом;
- c) именем и типом;
- d) значением.

A7. Формальные параметры процедуры:

- a) описываются в ее заголовке;
- b) перечисляются при вызове процедуры;
- c) указываются при описании данных в программе;
- d) указываются при описании внутренних переменных процедуры.

A8. В какую из перечисленных ниже структур можно объединять данные различного типа

- a) запись;
- b) файл;
- c) массив;
- d) множество?

A9. Запись – это

- a) именованный набор с фиксированным количеством однотипных данных;
- b) совокупность разнородных данных, описываемых и обрабатываемых как единое целое;
- c) ограниченная апострофами последовательность любых символов;
- d) множество элементов данных, объединенных и упорядоченных определенным образом.

A10. Какой язык относится к процедурным языкам программирования

- a) Pascal;
- b) Visual Basic;
- c) Лого;
- d) Java?

A11. Результатом вычисления функции $\text{Cory}(\text{'информатика'}, 3, 5)$ будет слово

- a) фор;
- b) рма;
- c) инфор;
- d) форма.

A12. Определите, какому условию задачи соответствует данная программа:

```

Var x:string; i,j,s,t:integer;
Begin readln(x); s:=0; for i:=1 to length(x) do
begin
t:=0;
for j:=1 to length(x) do
If x[i]=x[j] then t:=t+1;
if t>1 then s:=s+1/t; end; writeln(t); End.
    
```

- a) составьте программу, подсчитывающую количество различных символов в строке X;
- b) составьте программу, подсчитывающую количество неповторяющихся символов строки X;
- c) составьте программу, подсчитывающую количество повторяющихся символов в строке X;
- d) составьте программу, подсчитывающую количество символов в строке X.

A13. При наборе программы вычисления суммы отрицательных элементов массива вместо оператора $s:=s+a[k]$ ошибочно был записан оператор $s:=s+1$. Каким оказался ответ после исполнения неверной программы, если в качестве элементов массива были введены числа: -1, 3, -2, 4, -5, 6, -7, 8?

```

Var a: array[1..8] of Integer; s, k: Integer;
Begin
For k:=1 to 8 Do Readln(a[k]); s:=0;
For k:=1 to 8 Do If a[k]<0 Then s:=s+a[k]; Writeln(s) End
    
```

- a) -15;
- b) -3;
- c) 4;
- d) 10.

A14. Чему будет равно K после исполнения фрагмента программы:

```
K:=1;
```

While (A[K]<>X) AND (K<=10) Do K:=K+1; если в качестве элементов массива будут введены числа 2, 3, 5, 7, 9, 12, 0, 7, 6, 7, а X=7

- a) 14;
- b) 1;
- c) 10;
- d) 8?

A15. Определите, какая ошибка была допущена в программе, предназначенной для подсчета неотрицательных элементов двумерного массива.

```
Var A: array [1..10, 1..10] of real; N,I,J,t:integer; begin readln(N);
  for I:=1 to N do for J:=1 to N do readln(A[I,J]); t:=0;
  for i:=1 to N do for j:=1 to N do if a[I,J]>0 then t:=t+1; writeln(t); end.
```

- программа подсчитывает количество отрицательных элементов;
- программа вычисляет сумму элементов;
- не учитываются элементы со значением «0»;
- программа не выводит результат.

A16. Физическим именем файла в процедурном языке программирования называют:

- имя переменной, использующейся в программе при осуществлении операций над файлом;
- имя программы обработки файла;
- имя, под которым программа обработки файла хранится на диске;
- имя файла, под которым он записан на внешнем устройстве.

A17. Открывает существующий файл процедура

- reset;
- rewrite;
- close;
- rename.

A18. Определите, какому условию задачи соответствует представленная программа.

```
var w:text; s:integer; a:char;
begin
  s:=0; assign(w,'a.txt'); reset(w); while not eof(w) do
  begin if eoln(w) then s:=s+1; read(w,a); write(a)
  end; close(w); write(s) end.
```

- вывести текстовый файл на экран и подсчитать количество строк в данном текстовом файле;
- вывести текстовый файл на экран и подсчитать количество символов в данном текстовом файле;
- создать текстовый файл и подсчитать количество символов в нем;
- подсчитать количество символов в файле.

A19. Каким оказался ответ после исполнения программы:

```
const N=10;
type T1=1..N;
var A,B,C: set of T1; j: integer;
begin
  A:=[1,2,3,5,6]; B:=[1,4,6]; C:=A*B;
  for j:=1 to N do
    if j in C then write(j, ' '); end.
```

- 6;
- 1 2 3 4 5 6;
- 2 3 5;
- 4?

A20. В процедурном языке программирования основное различие между процедурами и функциями заключается в том, что:

- в процедуре допускается описание локальных переменных, а в функции – нет;
- в программе обращение к процедуре может осуществляться многократно, тогда как к функции только один раз;
- в процедуре допускается использование глобальных переменных, а в функции – нет;
- в результате работы процедуры можно получить любое количество данных, а функции – только одно.

Оценочный лист к типовому заданию А

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
a	b	a	c	b	b	a	a	b	a
A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
d	c	c	d	c	d	a	a	a	d

Примеры заданий для выявления продвинутого уровня сформированности компетенции УК-1

B1. В программе используется одномерный целочисленный массив А с индексами от 0 до 9. Значения элементов равны 3, 0, 4, 6, 5, 1, 8, 2, 9, 7 соответственно, т.е. $A[0] = 3$, $A[1] = 0$ и т.д. Определите значение переменной с после выполнения следующего фрагмента этой программы

```
s := 0
нц для i от 1 до 9
если A[i-1] > A[i] то
  c := c + 1
t := A[i]
```

```

A[i] := A[i-1]
A[i-1] := t
все
кц

```

В2. Запишите подряд все числа, которые будут напечатаны на экране при выполнении вызова F(9). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

```

procedure F(n: integer);
begin
  if n > 0 then
  begin
    write(n);
    F(n - 3);
    F(n div 3)
  end
end;

```

В3 Какие ошибки допущены в следующем описании массива:
 Var a: array[1...5.5] of char;

Оценочный лист к типовому заданию В

В1	В2	В3
5	9 6 3 1 2 3 1	Диапазон задается двумя точками, а не тремя, индексом не может быть дробное число

Примеры заданий для выявления высокого уровня сформированности компетенции УК-1

С1. Опишите любой из известных вам алгоритмов сортировки массива целых чисел по убыванию.

С2. Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Напишите программу на процедурном языке программирования, которая находит количество элементов массива, больших 100 и при этом кратных 5, а затем заменяет каждый такой элемент на число, равное найденному количеству. Гарантируется, что хотя бы один такой элемент в массиве есть. В качестве результата необходимо вывести измененный массив, каждый элемент массива выводится с новой строки. Например, для массива из шести элементов: 4 115 7 195 25 106 программа должна вывести числа 4 2 7 2 25 106

С3. Расскажите основные идеи метода быстрой сортировки (изучается в углубленном курсе информатики), опишите алгоритм.

Оценочный лист к типовому заданию С

С1. Допускается следующий вариант ответа:

- Для исходного массива выбрать минимальный элемент.
- Поменять его местами с последним элементом (после этого самый большой элемент будет стоять на своем месте).
- Повторить п.п. 1-2 с оставшимися $n-1$ элементами, то есть рассмотреть часть массива, начиная с первого элемента до предпоследнего, найти в нем минимальный элемент и поменять его местами с предпоследним ($n-1$)-м элементом массива, затем с оставшиеся ($n-2$)-мя элементами и так далее, пока не останется один элемент, уже стоящий на своем месте.

Представлен алгоритм сортировки методом простого выбора. Допускается также алгоритм сортировки методом пузырька и др.

С2. Допускается следующий вариант ответа:

```

Const N = 30;
var
  a: array [1..N] of longint;
  i, j, k: longint;
begin
  for i := 1 to N do
    readln(a[i]);
  k := 0;
  for i := 1 to N do
    if (a[i] > 100) and (a[i] mod 5 = 0) then
      k:=k+1;

```

```
for i := 1 to N do begin
  if (a[i] > 100) and (a[i] mod 5 = 0) then
    a[i] := k;
  writeln(a[i])
end.
```

С3. Быстрая сортировка является существенно улучшенным вариантом алгоритма сортировки с помощью прямого обмена. Принципиальное отличие состоит в том, что в первую очередь производятся перестановки на наибольшем возможном расстоянии и после каждого прохода элементы делятся на две независимые группы.

Алгоритм состоит из трёх шагов:

1. Выбрать опорный элемент из массива.
2. Разбиение: перераспределение элементов в массиве таким образом, что элементы меньше опорного помещаются перед ним, а больше или равные после.
3. Рекурсивно применить первые два шага к двум подмассивам слева и справа от опорного элемента. Рекурсия не применяется к массиву, в котором только один элемент или отсутствуют элементы.

Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Код контролируемой компетенции (индикаторы)	Наименование оценочного средства	Максимальное количество баллов	Всего баллов	Уровень освоения компетенции (в баллах)		
				Пороговый (56-70%)	Продвинутый (71-85%)	Высокий (86-100%)
УК-1.1	Задание А 1-10	10	10	5-6	7-8	9-10
УК-1.2	Задание А 11-20	10	10	5-6	7-8	9-10
УК-1.3	Задание В 1	5	5	2-3	4	5
УК-1.4	Задание В 2-3	20	20	10-14	15-16	17-20
УК-1.5	Задание С	15	15	8-10	11-13	14-15

Полученное число баллов (30-60) выставляется в графу «Промежуточная аттестация» балльно-рейтинговой карты дисциплины.

Тестовые задания по дисциплине «Программирование»
для выявления порогового уровня сформированности компетенции УК-1
Семестр 3

Примеры тестовых заданий

A1. Суть такого свойства алгоритма, как результативность, заключается в том, что:

- алгоритм всегда состоит из последовательности дискретных шагов;
- для записи алгоритма используются команды, которые входят в систему команд исполнителя;
- алгоритм обеспечивает решение не одной конкретной задачи, а некоторого класса задач;
- при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов и привести к определенному результату;

A2. Суть такого свойства алгоритма, как понятность, заключается в том, что:

- алгоритм всегда состоит из последовательности дискретных шагов;
- для записи алгоритма используются команды, которые входят в систему команд исполнителя;
- алгоритм обеспечивает решение не одной конкретной задачи, а некоторого класса задач;
- при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов и привести к определенному результату;
- алгоритм должен состоять из команд, однозначно понимаемых исполнителем.

A3. Алгоритм называется циклическим, если:

- он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
- последовательность выполнения его команд зависит от истинности тех или иных условий;
- его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;

d) он представим в табличной форме;

e) он включает в себя вспомогательный алгоритм.

A4. Характерным признаком линейной программы является:

- выполнение операторов в порядке их записи;
- наличие в каждой программной строке только одного оператора;

- с) использование в ней исключительно операторов присваивания;
- д) присутствие в ней операторов условного и безусловного перехода;
- е) присутствие в ней операторов цикла.

A5. Массив — это:

- а) поименованный набор фиксированного числа однотипных данных;
- б) ограниченная апострофами последовательность любых символов;
- с) совокупность разнородных данных, описываемых и обрабатываемых как единое целое;
- д) именованный набор однотипных данных на диске; 5) набор переменных, начинающихся с одной и той же

буквы.

A6. Фактические параметры процедуры:

- а) описываются в ее заголовке;
- б) перечисляются при вызове процедуры;
- с) указываются при описании данных в программе;
- д) указываются при описании внутренних переменных процедуры.

A7. Структура данных определяет:

- а) диапазон допустимых значений;
- б) внутреннее представление каждого элемента структуры;
- с) взаимное расположение, способ связи и выборки элементов структуры в памяти ПК;
- д) способ упорядочения переменных.

A8. При каком исходном значении переменной X результатом выполнения команды $X = X \bmod 3$ будет 0:

- а) При любом, кратном 3;
- б) 5;
- с) 4;
- д) 2?

A9. Операторы в языке Object Pascal отделяются друг от друга:

- а) двоеточием;
- б) точкой с запятой;
- с) запятой;
- д) пробелом.

A10. Служебное слово const в программе на языке Object Pascal фиксирует начало раздела программы, содержащего:

- а) описание переменных;
- б) описание меток;
- с) описание констант;
- д) описание сложных типов данных.

A11. Объектно-ориентированные технологии имеют следующие преимущества:

- а) уменьшенную связность между модулями;
- б) повышенное качество кода;
- с) высокий уровень абстракции;
- д) все из перечисленного.

A12. Какое из следующих утверждений верно:

- а) объект – это экземпляр класса;
- б) класс – это абстрактное определение для множества объектов;
- с) объект может быть более чем в одном классе;
- д) объект имеет линию жизни;
- е) все из перечисленного?

A13. Какое из следующих утверждений правильное:

а) классы на диаграммах классов могут быть сгруппированы в пакеты, чтобы показать общую организацию модели (архитектуру системы);

- б) на диаграмме объектов наименования экземпляров должны быть выделены наклонным шрифтом;
- с) если пакет В зависит от пакета А, то любые изменения в А вызывают изменения в пакете В;
- д) диаграммы объектов и диаграммы классов взаимозаменяемы.

A14. Какое утверждение описывает отношение наследования наилучшим образом:

- а) отношение «имеет»;
- б) отношение «является»;
- с) отношение «выполняет»;
- д) отношение «использует»;
- е) отношение «является элементом»?

A15. Машина имеет 4 колеса. Какой тип отношений имеет класс «Машина» и класс «Колесо»:

- а) ассоциация;
- б) агрегация;
- с) наследование;
- д) реализация?

A16. Какое из следующих утверждений является неверным:

- a) класс может иметь отношение к самому себе;
- b) объект может иметь отношение к другим объектам такого же
- c) класса;
- d) класс может иметь только одно отношение к другому классу;
- e) отношение может существовать с помощью «0»;
- f) класс может существовать без каких-либо отношений к другим объектам?

A17. Диаграмма Последовательности содержит:

- a) объекты;
- b) сообщения;
- c) видимость объекта;
- d) временные ограничения;
- e) все из перечисленного;
- f) все из перечисленного, кроме видимости объекта;
- g) все из перечисленного, кроме объектов.

A18. Какое из следующих утверждений является неверным:

- a) в какой-либо момент времени объект может быть более чем в одном состоянии;
- b) автомат может иметь множество конечных состояний;
- c) автомат может запоминать, какое подсостояние было последним состоянием;
- d) переход может срабатывать условно?

A19. Для чего нужен словарь модели:

- a) чтобы ограничивать количество используемых терминов;
- b) чтобы формировать терминологию предметной области;
- c) чтобы добавлять новые термины?

A20. Система обеспечивает принцип наследования в том случае, если:

- a) в системе имеется хотя бы один «наследник»;

b) в любом месте, где допустимо использование объекта, принадлежащего классу-предку, точно также допустимо использование класса-наследника:

c) в любом месте, где допустимо использование объекта, принадлежащего классу-наследнику, точно также допустимо использование класса-предка.

A21. Что из перечисленного является преимуществом объектноориентированного подхода к программированию перед структурным программированием:

- a) В ООП не используются функции и процедуры.
- b) ООП позволяет объединять состояние объектов и их поведение;
- c) ООП не поддерживает повторное использование компонентов; 4) ООП поддерживает разработку программ «сверху-вниз»; 5) ООП не поддерживает понятие абстракции?

A22. Какое наиболее важное преимущество предоставляет персистентность объектов в приложении:

- a) Персистентность делает систему более защищенной от внешних атак;
- b) атак;
- c) Персистентность делает исходный код системы простым в понимании и обеспечивает повторное использование кода;
- d) Персистентность позволяет системе быстрее стартовать;
- e) Персистентность позволяет хранить и восстанавливать состояние объектов между выключениями системы;
- f) Персистентность увеличивает скорость поиска объектов по атрибутам?

A23. Какому принципу ООП удовлетворяет обоняние собаки если: «Собака чует кошку – она лает, собака чует пищу – она бежит к миске»:

- a) Инкапсуляция;
- b) Полиморфизм;
- c) Наследование?

A24. Какая разница между объектом и классом:

- a) Класс – это исходный код, а объект – скомпилированный и выполняемый код;
- b) Класс описывает категорию, к которой могут либо принадлежать, либо не принадлежать объекты данного класса;

c) Класс может иметь много экземпляров, а объект – один или ни одного;

d) Класс может инстанцировать объекты, а сам объект – нет;

e) Объект – это экземпляр класса?

A25. Тип TSomeClass объявлен следующим образом TSomeClass = class private

A: array[1..10] of Byte; end;

Какое значение будет иметь выражение SizeOf(TSomeClass):

- a) 1;
- b) 4;
- c) 10;
- d) 10 + SizeOf(TObject)?

Оценочный лист к типовому заданию А

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
d	b	a	a	a	b	c	a	a	c
A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
c	e	a	a	b	b	f	f	a	b
A21	A22	A23	A24						
b	e	b	e						

Примеры заданий для выявления продвинутого уровня сформированности компетенции УК-1

B1. На некоторой форме вручную создаются компоненты TLabel и TStaticText на одном и том же месте
 procedure TForm1.Button1Click(Sender: TComponent); var L: TLabel; S: TStaticText;
 begin

```
S := TStaticText.Create(Self);
S.Caption := 'Это TStaticText';
S.Top := 10;
S.Left := 10;
S.Parent := Self;
L := TLabel.Create(Self);
L.Caption := 'Это TLabel';
L.Top := 10;
L.Left := 10;
L.Parent := Self; L.BringToFront; end;
```

Какую надпись и почему увидит пользователь?

B2. Требуется удалить компоненты, принадлежащие форме и удовлетворяющие некоторому условию. Среди компонентов могут встречаться как созданные при проектировании, так и созданные программно. Для этого в одном из методов формы используется такой код:

```
var
  I: Integer;
for I := 0 to ComponentCount - 1 do if {Проверка условия} then Components[I].Free;
Что неправильно в этом коде? Поясните свой ответ.
```

B3. На некоторой неглавной форме Form2 встречается такой код:

```
procedure TForm2.Button1Click(Sender: TObject); begin
  Form2.Label1.Caption := 'Кнопка нажата'; end;
Всё ли правильно в этом коде (при условии что компонент Label1:
TLabel действительно существует на форме)? Если нет, то что именно?
```

Оценочный лист к типовому заданию В (модельный ответ):

- B1 «Это TStaticText», потому что TStaticText оконный компонент, а TLabel неоконный.
 B2 Возникнет исключение, потому что индекс выйдет за границу списка.
 B3. В данном случае нельзя обращаться к компоненту Label1 через переменную Form2.

Примеры заданий для выявления высокого уровня сформированности компетенции УК-1

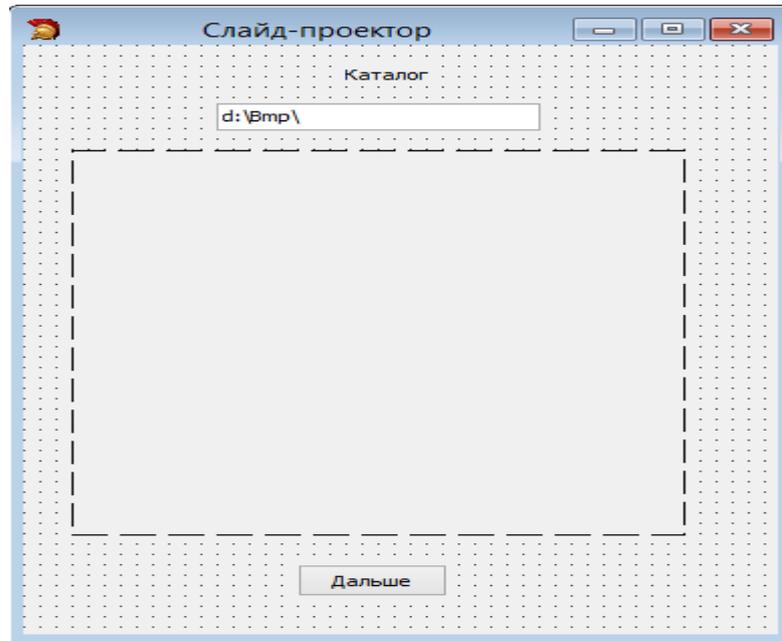
C1. Создайте и протестируйте новый класс, позволяющий хранить и осуществлять ввод/вывод информации об автомобиле. Данные должны быть представлены в виде полей: марка, цвет, категория транспортного средства (легковой или грузовой), год выпуска.

C2. Разработайте приложение, позволяющее просматривать графические изображения определенного формата (согласно Вашему варианту):

№ варианта	Графический формат
1, 4, 5, 9, 13, 15	JPG
2, 6, 8, 10, 13	JPEG
3, 7, 11, 12, 14	BMP

При этом к программе предъявляются следующие критерии:

1. Путь к каталогу с графическими файлами должен задаваться пользователем.
2. Путь и имя открытого графического файла должно отображаться в заголовке формы.



Оценочный лист (модельный ответ)

C1. Пример верно решенной задачи:

```
unit ClassAvto; interface
uses SysUtils, Dialogs, Classes; type
TAvto = class private
// Марка автомобиля
FCar: string;
// Цвет автомобиля
FColor: string;
// Тип транспортного средства: "Л" или "Г" FType: char;
// Год выпуска
FYear: word;
// Дополнительная информация
FMore: string;
```

```
protected
```

```
// Установка типа автомобиля procedure SetType(Value: char); public
```

```
property PCar: string read FCar write FCar; property PColor: string read FColor write FColor; property PYear: word
read FYear write FYear;
property PType: char read FType write SetType default 'Л'; property PMore: string read FMore write FMore;
end;
```

```
implementation
```

```
procedure TAvto.SetType(Value: char);
```

```
// Запись значения типа транспортного средства
```

```
begin
```

```
if Value in ['Л', 'Г'] then FType := Value
```

```
else ShowMessage('Введите правильное значение типа транспортного средства. Тип "' + Value +
"' является недопустимым'); end;
```

```
end.
```

Допускаются вариации, не влияющие на конечный результат вычисления, и не увеличивающие код существенно.

C2. Критерии оценивания:

1. Приложение позволяет просматривать изображения в формате, соответствующем варианту.
2. Путь к каталогу с графическими файлами задается пользователем.
3. Путь и имя открытого графического файла отображаются в заголовке формы.

Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Код контролируемой компетенции (индикаторы)	Наименование оценочного средства	Максимальное количество баллов	Всего баллов	Уровень освоения компетенции (в баллах)		
				Пороговый (56-70%)	Продвинутый (71-85%)	Высокий (86-100%)
УК-1.1	Задание А 1-10	10	10	5-6	7-8	9-10
УК-1.2	Задание А 11-20	10	10	5-6	7-8	9-10
УК-1.3	Задание А 21-25	10	10	5-6	7-8	9-10
УК-1.4	Задание В	20	20	10-14	15-16	17-20
УК-1.5	Задание С	10	10	5-6	7-8	9-10

Полученное число баллов (30-60) выставляется в графу «Промежуточная аттестация» балльно-рейтинговой карты дисциплины.

Тестовые задания по дисциплине «Программирование»
для выявления порогового уровня сформированности компетенции УК-1
Семестр 4

Примеры тестовых заданий

A1. Какие из нижеперечисленных парадигм программирования поддерживает язык Python

- a) объектно-ориентированное;
- b) структурное;
- c) функциональное;
- d) все вышеперечисленные?

A2. Язык Python является:

- a) регистрозависимым;
- b) компилируемым;
- c) высокоуровневым;
- d) всё вышеперечисленное.

A3. Инструкция list задает

- a) кортеж;
- b) список;
- c) множество;
- d) строку.

A4. Инструкция tuple задает

- a) кортеж;
- b) список;
- c) множество;
- d) строку.

A5. Равенство в Python обозначается конструкцией

- a) <>
- b) ==
- c) !=
- d) =!

A6. Что будет являться результатом выполнения следующих строк кода

```
x = input('введите число')  
print(type(x))
```

- a) <class 'int'>;
- b) <class 'float'>;
- c) <class 'str'>;
- d) класс переменной будет зависеть от введенных данных?

A7. Какая библиотека используется для работы с регулярными выражениями в Python?

- a) request;
- b) requests;
- c) re;
- d) random?

A8. С помощью какой инструкции в языке Python определяются функции

- a) function;
- b) procedure;
- c) def;
- d) proc?

A9. Функция open ('text.txt', 'r') открывает файл text.txt...

- a) На запись;
- b) На чтение;
- c) На дозапись;
- d) На чтение и запись.

A10. Что будет являться результатом выполнения следующих строк кода

```
b=6  
a=b+8  
print(a)
```

- a) 6;
- b) 8;
- c) False;
- d) True?

Тестовые задания по дисциплине «Программирование»

(для выявления порогового уровня сформированности компетенции СКИ-4)

- A11. Чему будет равна переменная `sum` результате выполнения следующего участка кода
- ```
sum = 0
for i in range(1, n + 1):
 sum += i
```
- a) `n!`;  
b) `(n+1)!`;  
c) `n`;  
d) `n+1`?
- A12. `S='программа'`. Результатом выполнения команды `list(S)` будет
- a) `['п', 'р', 'о', 'г', 'р', 'а', 'м', '']`;  
b) `['п', 'р', 'о', 'г', 'р', 'а', 'м', 'м', 'а']`;  
c) `['программа']`;  
d) возникнет ошибка.
- A13. `S='5'`. Каким будет результат выполнения команды `S*3`
- a) `15`;  
b) `'15'`;  
c) `'555'`;  
d) возникнет ошибка?
- A14. Какая из следующих конструкций вернёт длину слова `s`
- a) `s.length`;  
b) `s.length()`;  
c) `s.len`;  
d) `len(s)`?
- A15. `S='программа'`. Каким будет результат выполнения команды `S[::-1]`
- a) `['п', 'р', 'о', 'г', 'р', 'а', 'м', 'м', 'а']`  
b) `'аммаргорп'`  
c) `'программ'`  
d) `'рограмм'?`
- A16. Что будет выведено на экран в результате выполнения следующего участка кода
- ```
a=6.7
b = a // 2
pt=rint(a)
```
- a) `3`;
b) `3.35`;
c) `Syntax Error`;
d) `0.7`?
- A17. Что будет выведено на экран в результате выполнения следующего участка кода
- ```
a = [1,2, None,(),[],]
print(len(a))
```
- a) `3`;  
b) `5`;  
c) `Syntax Error`;  
d) `Infinity`?
- A18. Что будет выведено на экран в результате выполнения следующего участка кода
- ```
x=[1,2,3,4,5,6]
Print(x[3])
```
- a) `3`;
b) `4`;
c) `Syntax Error`;
d) `[1,2,3]`?
- A19. Что будет выведено на экран в результате выполнения следующего участка кода
- ```
x = [1, 2, 7]
y = x
y[-1] = ['a', 'b']
print x
```
- a) `[1, 2, 'a', 'b']`;  
b) `[1, 2, ['a', 'b']]`;  
c) `['a', 'b', 1, 2]`;  
d) `[[1, 2], 'a', 'b']`?
- A20. Что будет выведено на экран в результате выполнения следующего участка кода?
- ```
a=6
while a<10:
    print(str(a)*3,end='')
    a+=2
```

- a) 1824;
- b) 18 24;
- c) 666888;
- d) 666 888.

Оценочный лист к типовому заданию А

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
d	d	b	a	b	d	c	c	b	b
A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
a	b	c	d	b	a	b	b	b	c

Примеры заданий для выявления продвинутого уровня сформированности компетенции УК-1

В1. Что будет выведено на экран в результате выполнения следующего участка кода, если на вход были поданы числа 15,16,17? Приведите пример входных данных, при которых на экран будет выведено число 3.

```
a,b,c=(int(input()) for i in range(3))
print(max(a,b,c)%4)
```

В2. Ниже записаны две рекурсивные функции (процедуры): F и G. Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F(11)?

```
Python def F(n):
if n > 0: G(n - 1)
def G(n):
print("*")
if n > 1: F(n - 3)
```

В3. На обработку поступает положительное целое число, не превышающее 109. Нужно написать программу, которая выводит на экран сумму цифр этого числа, меньших 7. Если в числе нет цифр, меньших 7, требуется на экран вывести 0. Программист написал программу неправильно.

```
N = int(input())
sum = 0
while N > 0:
digit = N % 10
if digit < 7:
sum = sum + 1
N = N // 10
print(digit)
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 358.
2. Приведите пример такого трёхзначного числа, при вводе которого программа выдаёт верный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки: 1) выпишите строку, в которой сделана ошибка; 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Оценочный лист к типовому заданию В

В1. 1; 0,0,3

Программа выдаёт остаток от целочисленного деления максимального из числе a,b и c на 4. В примере это 17 и результат целочисленного деления - 1. Подходящим ответом на вторую часть задания будет любой набор чисел, в котором максимальное число при делении на 4 в остатке дает 3.

В2. 3

В3. Программа работает неправильно из-за неверной выводимой на экран переменной и неверного увеличения суммы. Соответственно, программа будет работать верно, если в числе старшая цифра (крайняя левая) равна сумме цифр, меньших 7.

1. Программа выведет число 3.
2. Пример числа, при вводе которого программа выдаёт верный ответ: 862.
3. В программе есть две ошибки.
 - 1) Неверное увеличение суммы. Строка с ошибкой: `sum = sum + 1`; Верное исправление: `sum = sum + digit`.
 - 2) Неверный вывод ответа на экран. Строка с ошибкой: `print(digit)` Верное исправление: `print(sum)`.

Примеры заданий для выявления высокого уровня сформированности компетенции УК-1.

С1. Составьте эффективную по памяти и по времени программу, проверяющую есть ли в числе одинаковые цифры?

С2. В физической лаборатории проводится долговременный эксперимент по изучению гравитационного поля Земли. По каналу связи каждую минуту в лабораторию передаётся положительное целое число – текущее показание

прибора «Сигма 2015». Количество передаваемых чисел в серии известно и не превышает 10 000. Все числа не превышают 1000. Временем, в течение которого происходит передача, можно пренебречь. Необходимо вычислить «бета-значение» серии показаний прибора – минимальное чётное произведение двух показаний, между моментами передачи которых прошло не менее 6 минут. Если получить такое произведение не удаётся, ответ считается равным –1. Напишите программу для решения поставленной задачи.

С3. Приведите примеры языков поддерживающий функциональное, логическое и процедурное программирование. Опишите отличия любых двух парадигм.

Оценочный лист к типовому заданию С

С1. Пример верно решенной задачи:

```
If len(x)=len(set(x)):
print ('нет одинаковых цифр')
else:
print ('есть одинаковые цифры')
```

Допускаются вариации, не влияющие на конечный результат вычисления, и не увеличивающие код существенно.

С2. Пример верно решенной задачи:

```
n=int(input())
l=[float(input())for i in range(n)]
sum_max=0
imax=0;
for i in range(5,n):
    if l[i-5]>sum_max:
        imax=l[i-5];
    if l[i]+imax>sum_max:
        sum_max=l[i]+imax;
print(sum_max)
```

Допускаются вариации, не влияющие на конечный результат вычисления, и не увеличивающие код существенно.

С3. Один из вариантов ответа:

Функциональное программирование: Logo, Haskell, Erlang, F#, Python

Логическое программирование: Prolog

Процедурное программирование: Паскаль, Python, Basic, Си, Go и др.

Процедурное программирование подразумевает последовательность изменений состояния программы, а переменные служат для хранения этого состояния. Функциональное программирование, наоборот, предусматривает последовательность действий над данными.

Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Код контролируемой компетенции (индикаторы)	Наименование оценочного средства	Максимальное количество баллов	Всего баллов	Уровень освоения компетенции (в баллах)		
				Пороговый (56-70%)	Продвинутый (71-85%)	Высокий (86-100%)
УК-1.1	Задание А 1-10	10	10	5-6	7-8	9-10
УК-1.2	Задание А 11-20	10	10	5-6	7-8	9-10
УК-1.3	Задание В 1	10	10	5-6	7-8	9-10
УК-1.4	Задание В 2-3	20	20	10-14	15-16	17-20
УК-1.5	Задание С	10	10	5-6	7-8	9-10

Полученное число баллов (30-60) выставляется в графу «Промежуточная аттестация» балльно-рейтинговой карты дисциплины.