МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Самарский государственный социально-педагогический университет»
Кафедра информатики, прикладной математики и методики их преподавания

Утверждаю
Проректор по учебно-методической
работе и качеству образования
_____ Н.Н. Кислова

Тюжина Ирина Викторовна

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
для проведения промежуточной аттестации по дисциплине
«Программирование»

Направление подготовки: 44.03.05 Педагогическое образование (с двумя профилями подготовки)
Направленность (профиль): «Информатика» и «Дополнительное образование (в области информатики и ИКТ)»
Квалификация выпускника
Бакалавр

Рассмотрено
Протокол № 1 от 28.08.2018
Заседания кафедры информатики, прикладной математики и методики их преподавания

Одобрено
Начальник Управления образовательных программ
Н.А. Доманина

Самара 2018

Пояснительная записка

Фонд оценочных средств (далее – ФОС) для промежуточной аттестации по дисциплине «Программирование» разработан в соответствии с требованиями Федерального государственного образовательного стандарта 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 (зарегистрирован Министерством юстиции Российской Федерации 15 марта 2018 г., регистрационный № 50358), с изменениями, внесенными приказами Министерства науки и высшего образования Российской Федерации от 26 ноября 2020 г. № 1456 (зарегистрирован Министерством юстиции Российской Федерации 27 мая 2021 г., регистрационный № 63650) и от 8 февраля 2021 г. № 83 (зарегистрирован Министерством юстиции Российской Федерации 12 марта 2021 г., регистрационный № 62739), основной профессиональной образовательной программой «Информатика» и «Дополнительное образование (в области информатики и ИКТ)» с учетом требований профессионального стандарта «01.001 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), 01.003 «Педагог дополнительного образования детей и взрослых» утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 22 сентября 2012 г. № 652н от 22.09.2021 г. (Зарегистрировано в Минюсте России 17.12.2021 N 66403).

Цель ФОС для промежуточной аттестации – установление уровня сформированности 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.

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

Знает: способы формализации алгоритмов на языках программирования высокого уровня.

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

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

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

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

Требование к процедуре оценки:

Помещение: особых требований нет

Расходные материалы: лист бумаги, ручка

Доступ к дополнительным справочным материалам: не предусмотрен

Нормы времени: 130 мин

Комплект оценочных средств для проведения промежуточной аттестации 3 семестр
Раздел «Процедурное программирование»

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.

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 (A), практические задания (B, C).

Содержание задания:

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?

А3. Запись алгоритма на языке конкретного исполнителя – это...
a) алгоритм;
b) команда;
c) программа;
d) процедура.

А4. Фактические параметры процедуры:
a) описываются в ее заголовке;
b) указываются при описании данных в программе;
c) перечисляются при ее вызове;
d) нигде не указываются.

А5. 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?

А6. При истинности какого условия последовательность переменных A, B, C не является упорядоченной по возрастанию:
a) (A<B) AND (NOT(B>=C));
b) NOT ((A>B) OR (B>C));
c) (A<=B) AND (NOT(B>=C));
d) NOT ((A<=B) AND (B<=C))?

А7. 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?

А8. 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?

А9. 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?

А10. 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?

А11. Служебное слово VAR в программе на языке Паскаль фиксирует начало раздела программы, содержащего:
a) операторы;
b) список меток;
c) перечень констант;
d) описание переменных.

А12. В алфавит языка Паскаль не входит служебное слово:
a) THEN;
b) BEGIN;
c) END;
d) STEP.

А13. Комментарий к тексту программы на языке Паскаль заключается:
a) в фигурные скобки;
b) в круглые скобки;
c) в квадратные скобки;
d) в апострофы.

А14. В качестве имени в языке Паскаль можно использовать
a) 1r;
b) or;
c) перем;
d) sum.

А15. Операторы в программе на языке Паскаль отделяются друг от друга:
a) двоеточием;
b) пробелом;
c) запятой;
d) точкой с запятой.

А16. Чему равен результат выражения 3 – 8 + 21 div 3
a) 2;
b) 5;
c) 12;
d) 18 ?

А17. Какое из приведенных неравенств верно:
a) –4.9876543234 E–02<–0,03;
b) –4.9876543234 E–02<–0,5;
c) –4.9876543234 E–02>–0,03;
d) –4.9876543234 E–02<–0,4?

А18. Какая из перечисленных операций не относится к логическим:
a) OR;
b) MOD;
c) NOT;
d) AND?

А19. 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 div 2=0;
b) I mod 2<>0;
c) N mod I=0;
d) N div I=1.

А20. 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 div 2=0
b) I mod 2<>0;

c) N mod I=0;
d) N 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

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 В

B1	B2	B3	B4
77	-3	13	79

С1. Составьте оптимальную по времени и использованию памяти программу, проверяющую можно ли представить число n в виде суммы двух простых слагаемых. Если такое возможно, то выводящую эти числа на экран.

Оценочный лист к типовому заданию С (модельный ответ)

С1. 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.

Комплект оценочных средств для проведения промежуточной аттестации 4 семестр

Раздел «Структуры данных»

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.

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 (A), практические задания (B, C).

Содержание задания:

A1. Структура данных – это
a) список данных, в котором порядок элементов списка задан посредством указателей, включенных в их запись;
b) множество элементов данных, объединенных и упорядоченных определенным образом;
c) представление логической организации данных в виде множества типов записей данных и связей между ними
d) список данных, в котором порядок элементов никак не обозначен.

A2. Укажите наиболее полный перечень способов записи алгоритмов
a) словесный;
b) словесный, графический, псевдокод, программный;
c) графический, программный;
d) словесный, программный.

A3. Суть такого свойства алгоритма, как дискретность заключается в том, что
a) алгоритм всегда состоит из последовательности дискретных шагов;
b) для записи алгоритма используются команды, которые входят в систему команд исполнителя;
c) алгоритм обеспечивает решение не одной конкретной задачи, а некоторого класса задач;
d) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов и привести к определенному результату.

A4. Алгоритм называется линейным, если
a) он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
b) последовательность выполнения его команд зависит от истинности тех или иных условий;
c) 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
a) обеспечивают непосредственное решение пользовательских задач;
b) 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 Copy('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 I     f 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.

a) программа подсчитывает количество отрицательных элементов;
b) программа вычисляет сумму элементов;
c) не учитываются элементы со значением «0»;
d) программа не выводит результат.

A16. Физическим именем файла в процедурном языке программирования называют:
a) имя переменной, использующейся в программе при осуществлении операций над файлом;
b) имя программы обработки файла;
c) имя, под которым программа обработки файла хранится на диске;
d) имя файла, под которым он записан на внешнем устройстве.

A17. Открывает существующий файл процедура
a) reset;
b) rewrite;
c) close;
d) 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.
```

a) вывести текстовый файл на экран и подсчитать количество строк в данном текстовом файле;
b) вывести текстовый файл на экран и подсчитать количество символов в данном текстовом файле;
c) создать текстовый файл и подсчитать количество символов в нем;
d) 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.
```

a) 6;
b) 1 2 3 4 5 6;
c) 2 3 5;
d) 4?

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

a) в процедуре допускается описание локальных переменных, а в функции – нет;
b) в программе обращение к процедуре может осуществляться многократно, тогда как к функции только один раз;
c) в процедуре допускается использование глобальных переменных, а в функции – нет;
d) 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

Содержание задания:

B1. В программе используется одномерный целочисленный массив A 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 c после выполнения следующего фрагмента этой программы

```
c := 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

B1	B2	B3
5	9 6 3 1 2 3 1	Диапазон задается двумя точками, а не тремя, индексом не может быть дробное число

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 строчки. 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:

1. Для исходного массива выбрать минимальный элемент.
2. Поменять его местами с последним элементом (после этого самый большой элемент будет стоять на своем месте).
3. 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.

Комплект оценочных средств для проведения промежуточной аттестации 5 семестр

Раздел «Мультипарадигмальное программирование»

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.

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 (A), практические задания (B, C).

Содержание задания:

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 является 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 является результатом выполнения следующих строк кода
b=6
a=b=8
print(a)
a) 6;
b) 8;
c) False;
d) True?

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) ‘рограмма’?

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

Каждый верный ответ оценивается в 2 балла.

B1. 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)
```
B2. 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)
```

B3. 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 B

B1. 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.

B2. 3

B3. 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).

C1. Составьте эффективную по памяти и по времени программу, проверяющую есть ли в числе одинаковые цифры?

C2. 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.

C3. Приведите примеры языков поддерживающий функциональное, логическое и процедурное программирование. Опишите отличия любых двух парадигм.

Оценочный лист к типовому заданию C

C1. 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.

C2. 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.

Комплект оценочных средств для проведения промежуточной аттестации 6 семестр

Раздел «Подключаемые библиотеки»

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.

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 (А), практические задания (В).

Содержание задания:

А1. К библиотекам анализа данных в Python НЕ относятся …
a) tkinter
b) Pandas;
c) NumPy;
d) Matplotlib.

А2. К модулям в Python относят:
a) любой файл с расширением Py;
b) специальные программы, расположенные на официальном сайте Python.org;
c) только элементы стандартной библиотеки;
d) файлы с расширением pip.

А3. Система управления пакетами, которая используется для установки и управления программными пакетами, написанными на Python –
a) import;
b) install;
c) pip;
d) uses.

А4. Модели последовательностей описывают …
a) правила или набор правил в соответствии с которыми можно отнести описание любого нового объекта к одному из классов;

b) функции, которые позволяют прогнозировать изменения непрерывных числовых параметров;
c) функциональные зависимости между зависимыми и независимыми показателями и переменными в понятной человеку форме;
d) группы, на которые можно разделить объекты, данные о которых подвергаются анализа.

A5. До предполагаемых моделей относятся такие модели данных:
a) модели классификации и последовательностей;
b) регрессивные, кластеризации, исключений, итоговые и ассоциации;
c) классификации, кластеризации, исключений, итоговые и ассоциации;
d) модели классификации, последовательностей и исключений.

A6. К описательным моделям относятся следующие модели данных:
a) модели классификации и последовательностей;
b) регрессивные, кластеризации, исключений, итоговые и ассоциации;
c) классификации, кластеризации, исключений, итоговые и ассоциации;
d) модели классификации, последовательностей и исключений.

A7. Модели классификации описывают …
a) правила или набор правил в соответствии с которыми можно отнести описание любого нового объекта к одному из классов;
b) функции, которые позволяют прогнозировать изменения непрерывных числовых параметров;
c) функциональные зависимости между зависимыми и независимыми показателями и переменными в понятной человеку форме;
d) группы, на которые можно разделить объекты, данные о которых подвергаются анализа.

A8. Дана инструкция «from math import e, ceil as c». Выберите верные утверждения.
a) math – модуль, e,ceil – атрибуты, c – псевдоним
b) math, ceil – модуль, e– атрибут, c – псевдоним
c) ceil – атрибут модуля c
d) c псевдоним для атрибута ceil;
e) e атрибут модуля math.

A9. Кластеризация — …
a) это установление зависимости непрерывной выходной переменной от входных переменных;
b) эта группировка объектов (Наблюдений, событий) на основе данных, описывающих свойства объектов;
c) выявление закономерностей между связанными событиями;
d) это установление зависимости дискретной выходной переменной от входных переменных.

A10. Свойство дискретности алгоритмов заключается в том, что…
a) процедура решения задачи распадается на последовательность шагов, а на каждом шаге обрабатывается порция информации конечного объема;
b) величины, с которыми работает программа - дискретны;
c) алгоритм не может работать с аналоговыми данными;
d) выполнение алгоритма должно завершаться получением искомого результата за конечное число шагов.

A11. Регрессивные модели описывают …
a) правила или набор правил в соответствии с которыми можно отнести описание любого нового объекта к одному из классов;
b) функции, которые позволяют прогнозировать изменения непрерывных числовых параметров;
c) функциональные зависимости между зависимыми и независимыми показателями и переменными в понятной человеку форме;
d) группы, на которые можно разделить объекты, данные о которых подвергаются анализа.

A12. Какие способы позволяют отобрать 5 первых строк датафрэйма dataf:
a) dataf.head(5); dataf.iloc[0:5]; df.tail(7);
b) dataf.head(5); dataf.iloc[0:5]; dataf.head(5);
c) dataf.head(5); dataf.head(5); df.tail(7);
d) dataf.iloc[0:5]; dataf.head(5); df.tail(7).

A13. Отметьте верные утверждения:

a) модуль нельзя именовать также, как и ключевое слово
b) мена модулей нельзя начинать с цифры
c) имена модуле должны содержать буквы и цифры
d) модуль необходимо называть также, как встроенные функции

A14 Модуль tkinter предназначен для

a) анализа данных в Python;
b) создания приложений с графическим интерфейсом;
c) построения графиков;
d) машинного обучения в Python.

A15. Определенная последовательность действий, выполнение которой приводит к достижению поставленной цели…

a) программой;
b) функцией;
c) процедурой;
d) алгоритмом.

A16. Выберите верные утверждения:

a) Ошибка IndentationError – связана с неправильными отступами;
b) Ошибка IndentationError дочернее исключение по отношению к SyntaxError;
c) Ошибка IndentationError родительское исключение по отношению к SyntaxError;
d) Ошибка IndentationError дочернее исключение по отношению к OSError;

A17. Исключение ValueError говорит о том, что…

a) функция получает аргумент правильного типа, но некорректного значения;
b) абстрактные методы класса требуют переопределения в дочерних классах;
c) не найдено переменной с таким именем;
d) не удалось импортирование модуля или его атрибута.

A18. При обучении дерева решений на обучающей выборке, получена 100% точность классификатора, однако на тестовых классификатор показал точность около 50%. Что может быть причиной такого результата?

a) обучающая выборка слишком большая;
b) возникла проблема переобучения;
c) дерево решений имеет недостаточную глубину;
d) исходные данные некорректны.

A19. Инструкция int('пятнадцать') вернет..

a) 15
b) ошибку ValueError
c) Nan
d) DevisionbyZerroError

A20. Исключение, которое порождается встроенной функцией next, если в итераторе больше нет элементов называется

a) AttributeError
b) AssertionError
c) StopIteration
d) OverflowError

Оценочный лист к типовому заданию A

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

Каждый верный ответ оценивается в 2 балла.

Задание B

1. Оцените следующий алгоритм. Для чего предназначены переменные lines, words, letters. Опишите, что делает программа в общем случае? Предложите альтернативный вариант решения.

```
import sys
fname = sys.argv[1]
lines = 0
words = 0
letters = 0

for line in open(fname):
   lines += 1
   letters += len(line)

   pos = 'out'
   for letter in line:
      if letter != ' ' and pos == 'out':
         words += 1
         pos = 'in'
      elif letter == ' ':
         pos = 'out'

print("Lines:", lines)
print("Words:", words)
print("Letters:", letters)
```

2. Даны файлы

   a1.py:

   ```
   b=2
   import a2
   ```

   a2.py:

   ```
   print(b)
   ```

   Что произойдет при запуске файла a1.py. Почему? Что необходимо сделать, чтобы исправить ситуацию?

Оценочный лист к типовому заданию (модельный ответ):

B1. Переменные lines, words, letters отвечают за количество строк, слов и символов соответственно. Программа подчитывает количество строк, слов и символов в файле. Альтернативное решение может быть связано с использованием метода split, однако засчитывается любое альтернативное работоспособное решение. Задача оценивается в 10 баллов

B2. Запуск приведет к ошибке "NameError: name 'b' is not defined". Проблема связана с областями видимости, решается например объявлением переменной.

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 1-5	10	10	5-6	7-8	9-10
УК-1.2	Задание А 6-10	10	10	5-6	7-8	9-10
УК-1.3	Задание А 11-15	10	10	5-6	7-8	9-10

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-1.4	Задание А 16-20	10	10	5-6	7-8	9-10
УК-1.5	Задание В	20	20	10-14	15-16	17-20

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

Комплект оценочных средств для проведения промежуточной аттестации в 7 семестре

Раздел «Объектно-ориентированное программирование»

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.

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 (А), практические задания (В).

Содержание задания:

А1. Выберите правильный вариант объявления константной переменной в C++, где type - тип данных в C++ variable - имя переменной value - константное значение

a) const variable = value;
b) const type variable = value;
c) const type variable := value;

А2. Чтобы подключить заголовочный файл в программу на C++, например iostream необходимо написать:

a) #include <> с iostream внутри скобок
b) #include <>; с iostream.h внутри скобок
c) include #iostream,h;
d) include (iostreamh)

А3. Какой из ниже перечисленных операторов, не является циклом в C++?

a) while
b) repeat until
c) for
d) do while

А4. Чему будет равна переменная a, после выполнения этого кода int a; for(a = 0; a < 10; a++) {}?

a) 9
b) 10
c) 1
d) 17

А5. Укажите объектно-ориентированный язык программирования

a) Все варианты ответов
b) C++
c) Java
d) Eiffel

А6. Какую функцию должны содержать все программы на C++?

a) system()
b) start()
c) main()
d) program()

А7. Какие служебные символы используются для обозначения начала и конца блока кода?

a) { }
b) < >
c) begin end
d) ( )

А8. Цикл с постусловием?

a) while
b) for
c) do while
d) repeat

А9. Какая из следующих записей - правильный комментарий в C++?

a) */ Комментарии */

b) {комментарий}
c) /* комментарий */
d) ** Комментарий **

А10. Программа, переводящая входную программу на исходном языке в эквивалентную ей выходную программу на результирующем языке, называется:
a) интерпретатор
b) сканер
c) транслятор
d) компилятор

А11. Цикл с предусловием?
a) while
b) do while
c) for
d) repeat

А12. Какой оператор не допускает перехода от одного константного выражения к другому?
a) Stop;
b) end;
c) точка с запятой
d) break;

А13. Какими знаками заканчивается большинство строк кода в Си++?
a) ; (точка с запятой)
b) : (двоеточие)
c) , (запятая)
d) . (точка)

А14. Каков результат работы следующего фрагмента кода?

```
int x = 0;

  switch(x)
  {

   case 1: cout << "Один";

   case 0: cout << "Нуль";

   case 2: cout << "Привет мир";

  }
```

a) НульПривет мир
b) Привет мир
c) Нуль
d) Один

А15. Какое значение, по умолчанию, возвращает программа операционной системе в случае успешного завершения?
a) 0
b) 1
c) -1
d) Программа не возвращает значение.

А16. Простые типы данных в С++.
a) целые – int, вещественные – float или double, символьные – string
b) целые – int, вещественные – float или double, символьные – char
c) целые – int, вещественные – float или real, символьные – char
d) целые – bool, вещественные – float или double, символьные – string

А17. Какой служебный знак ставится после оператора case ?
a) :
b) ;
c) .

d) -

18. Укажите правильное определение функции main в соответствии со спецификацией стандарта ANSI
a) int main(void)
b) int main()
c) void main()
d) void main(void)

19. Какому зарезервированному слову программа передаёт управление в случае, если значение переменной или выражения оператора switch не совпадает ни с одним константным выражением?
a) default
b) all
c) other
d) contingency

20. Какие среды программирования (IDE) предназначены для разработки программных средств?
a) MVS, NetBeans, QT Creator, RAD Studio, Dev-C++
b) MVS, Code::Blocks, QT Creator, RAD Studio, MathCAD
c) MVS, Code::Blocks, QT Creator, AutoCAD, Eclipse

Оценочный лист к типовому заданию А

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

Каждый верный ответ оценивается в 2 балла.

Задание В

В1. Оцените следующий алгоритм. Что вернет программа при вводе num5. Опишите, что делает программа в общем случае? Предложите альтернативный вариант решения.

```
#include <iostream>
#include <locale.h>

using namespace std;

int main()
{
  int number;
  setlocale(LC_CTYPE,"Russian");
  cout << "Введите число: ";
  cin >> number;
  cin.ignore();
  cout << "Вы ввели: "<< number <<"\n";
  cin.get();
}
```

В2. Последовательность состоит из натуральных чисел и завершается числом 0. Всего вводится не более 10000 чисел (не считая завершающего числа 0). Определите, сколько элементов этой последовательности равны ее наибольшему элементу.

Оценочный лист к типовому заданию (модельный ответ):

В1. Программа вернет 5. Программа возвращает введенное число, игнорируя остальные символы. Засчитывается любое альтернативное работоспособное решение. Задача оценивается в 10 баллов

В2. Оценивается работоспособность программы и скорость работы алгоритма. Задача оценивается в 10 баллов

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 1-5	10	10	5-6	7-8	9-10
УК-1.2	Задание А 6-10	10	10	5-6	7-8	9-10
УК-1.3	Задание А 11-15	10	10	5-6	7-8	9-10
УК-1.4	Задание А 16-20	10	10	5-6	7-8	9-10
УК-1.5	Задание В	20	20	10-14	15-16	17-20

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

Комплект оценочных средств для проведения промежуточной аттестации в 8 семестре

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.

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 (А), практические задания (В).

Содержание задания:

А1. Оператор вывода cout может печатать несколько значений или переменных в одной команде, используя следующий синтаксис:

a) cout << "Привет", name, "n";
b) cout << "Привет" + name + "n";
c) cout << ("Привет" & name & "n");
d) cout << "Привет" << name << "n";

А2. Какое значение будет напечатано, в результате выполнения программы?

```
#include
 int main()
{
 int x = 3;
  switch(x)
 {
  case 0:
   int x = 1;
   std::cout << x << std::endl;
  break;
  case 3:
   std::cout << x << std::endl;
  break;
  default:
   x = 2;
   std::cout << x << std::endl;
 }
 return 0;
}
```

a) 3
b) 2
c) ничего не напечатается, программа вообще не будет работать
d) 1
e) 0

А3. Тело оператора выбора if, будет выполняться, если его условие:

a) ложно (false)
b) истинно (true)
c) выполнение не зависит от условия

A4. Укажите блок кода, в котором переменная y доступна.

```
int main(int argc, char** argv)
{
 if ( argc > 10 )
{
}
 else if (int y = argc - 1 )
 {
   }
 else
 {
 }
 return 0;
 }
```

a) строки 8 -11
b) строки 8 -17
c) строки 8 -15
d) строки 4 -17
e) строки 4 -15

A5. Что появится на экране, после выполнения этого фрагмента кода?

```
int a = 1, b =2;
if (a == b);
cout << a << " = " << b << endl;
```

a) синтаксическая ошибка
b) 1 = 2
c) a = b
d) вывод на экран не выполнится

A6. Результат выполнения следующего фрагмента кода: !((1 || 0) && 0)

a) результат не может быть заранее определен
b) 1
c) 0
d) 4

A7. Какое из следующих значений эквивалентно зарезервированному слову true?

a) 0.1
b) 1
c) -1
d) 66
e) Все варианты ответов

A8. Это значение 5.9875e17 может быть сохранено в переменной, типа

a) Bool
b) Float
c) Int
d) Long
e) Short

A9. Вывод данных в C++

a) cout « <переменная >,< “< строка выводится на экран>” ,<выражение > ,endl;
b) cout « <переменная >,< “< строка выводится на экран>” ,<выражение > ,endl;
c) cout « <переменная > « “< строка выводится на экран>” « <выражение > « endl;

A10. В каком случае лучше всего использовать приведение типов данных?

a) во всех выше указанных случаях
b) чтобы разрешить программе использовать только целые числа
c) чтобы изменить тип возвращаемого значения функции
d) при делении двух целых чисел, для того, чтобы вернуть результат с плавающей точкой

A11. Какой тип данных имеет переменная ARGV?

a) это не переменная
b) char **
c) char *

d) int

A12. Что будет напечатано на экране, после выполнения этого кода?

```
#include
 int foo(int y);
int foo(int x)
{
  return x+1;
}
 int main(int argc, char** argv)
{
  int x = 3;
  int y = 6;
   std::cout << foo(x) << std::endl;
   return 0;
}
```

a) 4
b) ошибка компиляции
c) 3
d) 9

A13. Какая строка содержит зарезервированные слова языка программирования C++?

a) sizeof, const, typedef, static, voided, enum, struct, union
b) char, int, float, doubled, short, long, unsigned, signed
c) if, else, for, while do, switch, continue, break
d) defaulted, goto, return, extern, private, public, protected

A14. Что будет напечатано, после выполнения этого кода: cout << (5 << 3); ?

a) 35
b) 53
c) 40
d) 56

A15. Укажите неправильно записанную операцию отношения

a) >=
b) <=
c) =!
d) все операторы записаны правильно

A16. Результат выполнения следующего фрагмента кода: cout << 22 / 5 * 3;

a) 12
b) 13.2
c) 1
d) 1.47
e) Другое

A17. В каком случае можно не использовать фигурные скобочки в операторе выбора if?

a) если в теле оператора if всего один оператор
b) если в теле оператора if два и более операторов
c) нет правильного ответа
d) если в теле оператора if нет ни одного оператора

A18. Ввод данных в C++

a) cin » <выражение1> » <выражение2>…;
b) cin » <выражение1>,<выражение2>,…;
c) cin » <выражение1> » <выражение2> » endl »…;

A19. Какое ключевое слово указывает, что целая переменная не может принимать отрицательные значения?

a) Unsigned
b) Positive
c) нет такого зарезервированного слова
d) long
e) другое

A20. Преобразование целочисленной переменной value в ASCII эквивалент

a) cout << value

b) atoi(value)
c) (char) value
d) char (value)

Оценочный лист к типовому заданию А

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

Каждый верный ответ оценивается в 2 балла.

Задание В

В1. Оцените следующий алгоритм. Что вернет программа при вводе числа 105. Опишите, что делает программа в общем случае? Предложите альтернативный вариант решения.

```
void prime_divisors(int n)
{
    int t,k;
    cout<<n<<" =";
    if (n<=1)
    {
            cout<<" "<<n;
            return ;
    }
    t=sqrt(n);
    k=2;
    while (k<=t)
    {
            if (!(n%k))
            {
                    n/=k;
                    cout<<" "<<k;
                    while (!n%k)
                    {
                            n/=k;
                            cout<<" "<<k;
                }
                t=sqrt(n);
            }
            k++;
     }
     if (n>1)
            cout<<" "<<n<<endl;
}
```

В2. На электронную почту Вам пришло письмо, подписанное аббревиатурой (первыми буквами фамилии, имени и отчества (далее - ФИО) отправителя). Аббревиатура оказалась Вам незнакома. У Вас есть список всех предполагаемых отправителей, взятый из ранее полученных писем, среди которых различных людей с такой аббревиатурой не больше 10.
Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая определит всех вероятных адресатов – людей, ФИО которых можно сократить до нужной аббревиатуры. ФИО следует выдать в порядке убывания частоты их встречаемости в списке, если несколько человек встречаются с одинаковой частотой, то вывести их в алфавитном порядке.

Оценочный лист к типовому заданию (модельный ответ):
В1. Программа вернет 3*5*7. Программа раскалывает введенное число на простые сомножители. Засчитывается любое альтернативное работоспособное решение. Задача оценивается в 10 баллов
В2. Оценивается работоспособность программы и скорость работы алгоритма. Задача оценивается в 10 баллов

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 1-5	10	10	5-6	7-8	9-10
УК-1.2	Задание А 6-10	10	10	5-6	7-8	9-10
УК-1.3	Задание А 11-15	10	10	5-6	7-8	9-10
УК-1.4	Задание А 16-20	10	10	5-6	7-8	9-10
УК-1.5	Задание В	20	20	10-14	15-16	17-20

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

Комплект оценочных средств для проведения промежуточной аттестации в 9 семестре

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.

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 (А), практические задания (В).

Содержание задания:

А1. Функциональное программирование — парадигма программирования, в которой
- a) процесс вычисления трактуется как вычисление значений функций в математическом понимании последних;
- b) функции трактуются как подпрограммы;
- c) программы представляются в виде иерархической структуры блоков;
- d) программы представлены в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.

А2. Какая особенность функционального подхода дает путь к достижению надежности программ?
- a) математическая основа исходных понятий;
- b) доказательность основных построений при разработке универсальных функций;
- c) разнообразие встроенных функций и библиотек;
- d) высокий уровень языковых средств.

А3. Суть такого свойства алгоритма, как результативность, заключается в том, что
- a) алгоритм всегда состоит из последовательности дискретных шагов;
- b) для записи алгоритма используются команды, которые входят в систему команд исполнителя;
- c) алгоритм обеспечивает решение не одной конкретной задачи, а некоторого класса задач;
- d) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов и привести к определенному результату

А4. Суть такого свойства алгоритма, как понятность, заключается в том, что:
- a) алгоритм всегда состоит из последовательности дискретных шагов;
- b) для записи алгоритма используются команды, которые входят в систему команд исполнителя;
- c) алгоритм обеспечивает решение не одной конкретной задачи, а некоторого класса задач;
- d) алгоритм должен состоять из команд, однозначно понимаемых исполнителем.

А5. Алгоритм называется циклическим, если
- a) он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
- b) последовательность выполнения его команд зависит от истинности тех или иных условий;
- c) его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;
- d) он включает в себя вспомогательный алгоритм.

А6. Характерным признаком линейной программы является:
- a) выполнение операторов в порядке их записи;
- b) наличие в каждой программной строке только одного оператора;
- c) использование в ней исключительно операторов присваивания;
- d) присутствие в ней операторов условного и безусловного перехода.

А7. В каких из нижеперечисленных языков используется функциональное программирование
- a) Lisp;
- b) Python;

c) Prolog;
d) Logo?

A8. Как работает команда «останов»
a) использованная во вложенной процедуре команда «останов» возвращает управление в головную процедуру;
b) прекращает выполнение всех процедур и возвращает управление в поле команд;
c) останавливает работу черепашки;
d) пропускает следующая после команды «останов» инструкция?

A9. Что из перечисленного является словом в Logo
a) “слово;
b) “слово”;
c) ‘слово;
d) :слово?

A10. Что из перечисленного является списком в системе Logo
a) [1 2 3];
b) [];+
c) 1,2,3;
d) [[1 2 3] [3 4]]?

A11. Команда ПО…
a) выводит на экран текущую версию программного обеспечения;
b) опускает перо черепашки;
c) заставляет черепашку сделать поворот на 180 градусов;
d) поднимает перо черепашки?

A12. Чему будет равна переменная N после выполнения следующих команд
ПУСТЬ “N 0
ПОВТОРИ 5 [ПУСТЬ “N :N + 1]]
a) 0;
b) 1;
c) 5;
d) 10?

A13. Какая фигура будет изображена на экране после выполнения следующей серии команд: вп 20 пр 90 вп 20 лв 90 вп 20 пр 90?
a) квадрат;
b) прямоугольник;
c) треугольник;
d) ступенька.

A14. На какой угол относительно своего начального положения повернётся черепаха после выполнения команд пр 120 лв 30 пр 45 лв 90
a) 90;
b) 45;
c) 30;
d) 120?

A15. На каком расстоянии от своего начального положения будет находиться черепаха после выполнения команд нд 30 вп 90 нд 40
a) 50;
b) 20;
c) 40;
d) 30?

A16. Какой из наборов команд приведёт к вычерчиванию пятиконечной звезды
a) по повтори 5 [вп 40 пр 144];
b) пп повтори 5 [вп 40 пр 144];
c) по повтори 5 [вп 144 пр 40];
в) пп повтори 5 [вп 144 пр 40]?

A17. Команда «штамп»…
a) меняет форму исполнителя;
b) оставляет оттиск исполнителя (черепашки) на экране;
c) загружает окно редактора формы;
d) активизирует исполнителя.

A18. Чему равна переменная у после выполнения команд: пусть “x “программа пусть “t псл :x пусть “z кпсл :x пусть “y слово :t :z

a) Программап;
b) Апрограмм;
c) Программа;
d) Аммаргорп?

A19. С помощью какой команды можно отсечь от слова первую букву?

a) прв;
b) кпсл;
c) псл;
d) кпрв.

A20. Чему будет равна переменная Z после выполнения ряда команд:

пусть “x 0
пусть “I 0
повтори 4 [если остаток :I 2 = 0 [пусть “x : x + :i] пусть “i :i + 1]
Еслииначе :x > 4 [пусть “z “1]
[еслииначе :x > 1 [пусть“z “2][пусть“z “3]]

a) 0;
b) 1;
c) 2;
d) 3.

Оценочный лист к типовому заданию А

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

Каждый верный ответ оценивается в 2 балла.

Задания В

B1. Ниже приведена программа на языке Logo. Укажите через запятую, каковы будут результаты программы для слов «отвага», «программирование» и «крот»

```
Это пример :X
    Пусть “Y “аеёиоуэюя
    Пусть “Lсколько :X
    Пусть “P 0
    Повтори :L
    [Пусть “T первый :X
    Если входит? :T :Y
    [Пусть “P :P + 1]
    Пусть “X кпрв :X]
    Покажи :P
Конец
```

B2. Что появится на экране в результате выполнения следующих команд «черепашьей графики»
Повтори 6 [вп 20 пр 60]?

B3. Что появится на экране в результате выполнения следующих команд «черепашьей графики»
по
повтори 90 [вп 1 пр 1]?

B4. Ниже приведены три процедуры. Какое изображение появится на экране после выполнения процедуры «итог»

```
Это примитив
Повтори 4 [вп 20 пр 90]
Конец

Это примитив_2
Повтори 3 [вп 20 пр 120]
Конец

Это итог
Примитив
```

Вп 20 пр 30
Примитив_2
Лв 30
Конец?

Оценочный лист к типовому заданию В (модельный ответ):

B1	B2	B3	B4
3,7,1	шестиугольник	четверть окружности	домик

Каждая задача оценивается в 5 баллов

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 1-5	10	10	5-6	7-8	9-10
УК-1.2	Задание А 6-10	10	10	5-6	7-8	9-10
УК-1.3	Задание А 11-15	10	10	5-6	7-8	9-10
УК-1.4	Задание А 16-20	10	10	5-6	7-8	9-10
УК-1.5	Задание В	20	20	10-14	15-16	17-20

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