

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Попов Анатолий Николаевич  
Должность: директор  
Дата подписания: 31.05.2023 17:30:45  
Уникальный программный ключ:  
1e0c38dcaae73cee1e5e09c1d5873fc7497ba8

Приложение 2  
к рабочей программе дисциплины

## **ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

### **Основы программирования** *(наименование дисциплины(модуля))*

Направление подготовки / специальность

#### **27.03.05 Инноватика** *(код и наименование)*

Направленность (профиль)/специализация

#### **Управление инновациями** *(наименование)*

## Содержание

1. Пояснительная записка.
2. Типовые контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций.
3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации.

## 1. Пояснительная записка

Цель промежуточной аттестации – оценивание промежуточных и окончательных результатов обучения по дисциплине, обеспечивающих достижение планируемых результатов освоения образовательной программы.

### Перечень компетенций, формируемых в процессе освоения дисциплины

Код и наименование компетенции	Код индикатора достижения компетенции
ПК-1.1 Разрабатывает алгоритмы пригодные для практического применения	ПК-1.1.1 Знать: <ul style="list-style-type: none"> <li>– понятие алгоритма и классификацию алгоритмических языков и систем программирования;</li> <li>– язык программирования высокого уровня - Си;</li> </ul>
	ПК-1.1.2 Уметь: <ul style="list-style-type: none"> <li>– применять на практике современные технологии разработки алгоритмов, языки программирования, методы тестирования, отладки и решения задач на ЭВМ;</li> <li>– Разрабатывать программы в соответствии с выбранной моделью жизненного цикла;</li> </ul>
	ПК-1.1.3 Владеть: <ul style="list-style-type: none"> <li>– навыками алгоритмизации и программной реализации на языке высокого уровня решений практических задач;</li> <li>– навыками разработки программ в водопадной модели жизненного цикла</li> </ul>
ПК-1.2 Разрабатывает программный код и осуществляет отладку на языках программирования высокого уровня	ПК-1.2.1 Знать: <ul style="list-style-type: none"> <li>– понятие алгоритма и классификацию алгоритмических языков и систем программирования;</li> <li>– язык программирования высокого уровня - Си;</li> </ul>
	ПК-1.2.2 Уметь: <ul style="list-style-type: none"> <li>– применять на практике современные технологии разработки программ, языки программирования, методы тестирования, отладки и решения задач на ЭВМ;</li> <li>– Разрабатывать программы в соответствии с выбранной моделью жизненного цикла;</li> </ul>
	ПК-1.2.3 Владеть: <ul style="list-style-type: none"> <li>– навыками алгоритмизации и программной реализации на языке высокого уровня решений практических задач;</li> <li>– навыками разработки программ в водопадной модели жизненного цикла</li> </ul>

### Результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине	Оценочные материалы
ПК-1.1 Разрабатывает	ПК-1.1.1 Знать:	тест №1 - №5

алгоритмы пригодные для практического применения	– понятие алгоритма и классификацию алгоритмических языков и систем программирования;	
	ПК-1.1.2 Уметь: – применять на практике современные технологии разработки алгоритмов, языки программирования, методы тестирования, отладки и решения задач на ЭВМ;	Задания 1-3
	ПК-1.1.3 Владеть: – навыками алгоритмизации и программной реализации на языке высокого уровня решений практических задач;	Задания 8-11
ПК-1.2 Разрабатывает программный код и осуществляет отладку на языках программирования высокого уровня	ПК-1.2.1 Знать: – язык программирования высокого уровня - Си;	тест №6 - №9
	ПК-1.2.2 Уметь: – Разрабатывать программы в соответствии с выбранной моделью жизненного цикла;	Задания 4-7
	ПК-1.2.3 Владеть: – навыками разработки программ в водопадной модели жизненного цикла	Задания 12-15

Промежуточная аттестация (зачет) проводится в одной из следующих форм:

- 1) собеседование;
- 2) выполнение заданий в ЭИОС СамГУПС.

## 2. Типовые<sup>1</sup> контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций

### 2.1 Типовые вопросы (тестовые задания) для оценки знаний образовательного результата

Проверяемый образовательный результат:

Код и наименование индикатора достижения компетенции	Образовательный результат
ПК-1.1.1	Обучающийся знает: – понятие алгоритма и классификацию алгоритмических языков и систем программирования;
<p>Примерные вопросы</p> <p>1. Значение выражения <math>S := 3 * 7 / 2 \% 7 / 3</math> равно: 1) 10 2) 1 3) 1.5 4) 0.5</p> <p>2. Вычисляемое в программе <code>s:= 1; n = 1;</code> <code>for ( i= 2; i&lt;=n; i++) s= s + 1/i;</code> значение переменной S равно: 1) 3.5 2) 3 3) 1 4) 1.5</p> <p>3. Элементы массива A :<code>int A [6]</code> имеют значения: \$20, \$10, 8, 4, 2, 1. Чему равно значение выражения: <code>A[A[5]] + A[1] % A[4]</code>? 1) 0 2) 34 3) 36 4) 22</p> <p>4. Дана программа <code>include&lt;stdio.h&gt;</code> <code>int nod (int a, int b )</code>     {         while (a!=b)             if (a&gt;b)                 a= a – b;             else                 b= b - a;         return a;     }; <code>void main()</code>     { <code>int n,m,a;</code>         <code>m=8; n=6;</code>         <code>a=nod (m,n);</code> <code>printf(“%d”,a)</code>     } Что будет выведено на экран?</p>	

<sup>1</sup>Приводятся типовые вопросы и задания. Оценочные средства, предназначенные для проведения аттестационного мероприятия, хранятся на кафедре в достаточном для проведения оценочных процедур количестве вариантов. Оценочные средства подлежат актуализации с учетом развития науки, образования, культуры, экономики, техники, технологий и социальной сферы. Ответственность за нераспространение содержания оценочных средств среди обучающихся университета несут заведующий кафедрой и преподаватель – разработчик оценочных средств.

1) 8 2) 6 3) 4 4) 2

5. Укажите значение переменной s после выполнения операторов:

```
s = 0; i = 2;
while (i > 0)
{
    s = s + (s + 1)/i;
    i++;
};
```

1) 0.5 2) 1.5 3) 4 4) 2

ПК-1.2.1

Обучающийся знает:

– язык программирования высокого уровня - Си;

6. После вычисления присваивания  $a = b^{--} + c^{--}$ , при  $b=2, c=4$ . значения переменных будут:

1) a=5 2) a=5 3) a=4 4) a=6

b=1 b=2 b=1 b=1

c=3 c=3 c=3 c=3

7. Чему равно значение выражения  $(A[1] + A[3]) / cnt$ ?

```
const int A[] = { 1, 2, 3, 4, 5 };
int cnt = sizeof(A) / sizeof(*A);
```

```
1) 1 2) 1.4 3) 0.8 4) 9
```

3. Вычисляемое в программе

```
s = 1; n = 1;
for (i = 2; i >= n; i--)
```

```
s := s + 1.0/i;
```

```
значение переменной S равно:
```

```
1) 3.5 2) 3 3) 1 4) 1.5
```

8. Дан массив `int a [4]`; Элементы массива вычисляются по формуле:

```
 $a[i] = (i * 6) \% 4 + 1.$ 
```

Чему равен последний элемент массива?

1) -1 2) 1 3) 10 4) 3

9. Программа имеет:

```
#include <stdio.h>
```

```
double v=2;
```

```
double f (double x)
```

```
{
```

```
    v = v * x;
```

```
return v*v + x;
```

```
}
```

```
void main()
```

```
{
```

```
double y;
```

```
y=f(v)+v;
```

```
printf("Y=%le",y);
```

```
}
```

Что будет выведено на экран?

## 2.2 Типовые задания для оценки навыкового образовательного результата

### Проверяемый образовательный результат

Код и наименование индикатора достижения компетенции	Образовательный результат
ПК-1.1.2	Обучающийся умет: <ul style="list-style-type: none"> <li>– применять на практике современные технологии разработки алгоритмов, языки программирования, методы тестирования, отладки и решения задач на ЭВМ;</li> </ul>
	<ol style="list-style-type: none"> <li>1. Найти все совершенные числа, меньшие заданного числа <math>n</math>. Число называется совершенным, если равно сумме всех своих положительных делителей, кроме самого этого числа. Например, 28 – совершенно, т.к. <math>28 = 1 + 2 + 4 + 7 + 14</math>.</li> <li>2. Определить, является ли введенная пользователем строка записью натурального числа. Пример: '123' – является, а '34_4f' – не является.</li> <li>3. Дан массив данных об автомобилях заводского гаража (формирование происходит путем чтения массива из текстового файла): марка, номер, год выпуска, пробег, фамилия шофера. Вывести:               <ol style="list-style-type: none"> <li>а) список машин, подготовленных написание (старше заданного года выпуска и с пробегом больше заданной величины) и их количество;</li> <li>б) сообщение о том, есть ли автомобиль заданной марки (да/нет);</li> <li>с) марки автомобилей (без учета модификаций) с наибольшей величиной среднего пробега. В программе должен быть предусмотрен диалог с пользователем.</li> </ol> </li> </ol>
ПК-1.2.2	Обучающийся умеет: <ul style="list-style-type: none"> <li>– Разрабатывать программы в соответствии с выбранной моделью жизненного цикла;</li> </ul>
	<ol style="list-style-type: none"> <li>4. Найти все плоские числа, меньшие заданного числа <math>n</math>. Составное число называется плоским, если оно представимо в виде произведения двух сомножителей. Например: 4, 6, 8, 9, 10, 12, 14, 15, ...</li> <li>5. Найти в строке подстроку максимальной длины, состоящей из одного символа.</li> <li>6. Дан массив данных о состоянии пациентов палаты больницы (формирование происходит путем чтения массива из текстового файла): фамилия, дата поступления (число, месяц), температура, верхнее и нижнее артериальное давление, вес. Вывести:               <ol style="list-style-type: none"> <li>а) фамилии и температуру больных, у кого ее значение за пределами нормы (норма: <math>36 &lt; t &lt; 37</math>);</li> <li>б) сообщение о том, поступил ли кто-нибудь в палату в заданный день (да/нет);</li> <li>в) количество больных, которые лежат дольше 10 дней (на данную дату).</li> </ol> </li> <li>7. Написать рекурсивную программу перевода числа из десятичной системы счисления в двоичную.</li> </ol>

ПК-1.1.3	Обучающийся владеет: – навыками алгоритмизации и программной реализации на языке высокого уровня решений практических задач;									
<p>8. Написать программу, выводящую:</p> <table border="1" data-bbox="884 1563 1007 1827"> <tr><td>1</td></tr> <tr><td>21</td></tr> <tr><td>321</td></tr> <tr><td>4321</td></tr> <tr><td>54321</td></tr> </table> <p>9. Найти <math>n</math>-ое по порядку положительное число кратное 7 и меньшее либо равное введенного числа <math>X</math>.</p> <p>10. Даны два целочисленных массива, состоящие из одинакового числа элементов. Получить третий массив той же размерности, каждый элемент которого равен сумме соответствующих элементов данных массивов.</p> <p>11. Заполнить массив <math>A</math> размером <math>6 \times 4</math> следующим способом:</p> <table border="1" data-bbox="609 2072 1083 2112"> <tr> <td>1</td> <td>7</td> <td>13</td> <td>19</td> </tr> </table>		1	21	321	4321	54321	1	7	13	19
1										
21										
321										
4321										
54321										
1	7	13	19							



2	8	14	20
3	9	15	21
4	10	16	22
5	11	17	23
6	12	18	24

ПК-1.2.3	Обучающийся владеет: – навыками разработки программ в водопадной модели жизненного цикла
----------	---

12. Написать программу, выводящую следующее:

9
88
777
6666
55555

13. Найти  $n$ -ное по порядку простое число, следующее за введенным числом  $X$ .

14. Даны два целочисленных массива, состоящие из одинакового числа элементов. Получить третий массив той же размерности, каждый элемент которого равен большему из соответствующих элементов данных массивов.

15. Заполнить массив А размером  $6 \times 4$  следующим способом:

1	12	13	24
2	11	14	23
3	10	15	22
4	9	16	21
5	8	17	20
6	7	18	19

### **2.3. Перечень вопросов для подготовки обучающихся к промежуточной аттестации**

1. Язык программирования TurboPascal: общая структура программы, алфавит языка.
2. Базовые математические операции и стандартные математические функции в TurboPascal. Запись математических выражений.
3. Простые типы данных в системе программирования TurboPascal. . Переменные и константы.
4. Операторы в языке программирования TurboPascal. Простой и составной оператор. Оператор присваивания.
5. Логические выражения и операции отношения в языке программирования TurboPascal.
6. Процедуры ввода - вывода данных в языке программирования TurboPascal.
7. Операторы переходов: условный оператор IF.
8. Оператор выбора CASE в языке программирования TurboPascal.
9. Оператор цикла FOR... в языке программирования TurboPascal.
10. Оператор цикла WHILE... в языке программирования TurboPascal.
11. Оператор цикла REPEAT... в языке программирования TurboPascal.
12. Организация работ с процедурами в языке программирования TurboPascal.
13. Функции пользователя в TurboPascal.
14. Перечисляемый и интервальный типы данных: описание, диапазон значений, примеры использования.
15. Массивы и их реализация в языке программирования TurboPascal. Линейные массивы.
16. Стандартные алгоритмы обработки линейных массивов: подсчет суммы (произведения) элементов массива.
17. Стандартные алгоритмы обработки линейных массивов: определение максимального (минимального) элемента.
18. Поиск элемента в неупорядоченном массиве. Последовательный поиск.
19. Поиск элемента в упорядоченном массиве. Бинарный поиск.
20. Простые методы сортировки. Сортировка методом «пузырька».
21. Двумерные массивы: объявление, формирование, обработка.
22. Строки в TurboPascal. Операции над строками.
23. Функции обработки строк в языке программирования TurboPascal. .
24. Процедуры обработки строк в языке программирования TurboPascal. .

25. Тип запись в TurboPascal.
26. Файлы данных в TurboPascal. Файлы последовательного доступа.
27. Файлы данных в TurboPascal. Файлы произвольного доступа.
28. Стандартные процедуры и функции обработки файлов
29. ЭВМ как инструмент решения задач. Основные этапы решения задачи.
30. Генераторы случайных чисел. Функция RANDOM. Примеры использования.
31. Модули в TurboPascal. Стандартный модуль CRT.
32. Работа в среде Турбо-Паскаль в графическом режиме. Назначение модуля GRAPH.
33. Процедуры построения точек, линий, прямоугольников. Примеры использования.
34. Процедуры изображения дуг, окружностей, эллипсов. Примеры использования.
35. Процедуры выбора палитры, стиля заполнения. Примеры использования.
36. Алгоритм построения графика функций в декартовой системе координат.

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

#### Критерии формирования оценок по ответам на вопросы, выполнению тестовых заданий

- оценка **«отлично»** выставляется обучающемуся, если количество правильных ответов на вопросы составляет 100 – 90% от общего объёма заданных вопросов;
- оценка **«хорошо»** выставляется обучающемуся, если количество правильных ответов на вопросы – 89 – 76% от общего объёма заданных вопросов;
- оценка **«удовлетворительно»** выставляется обучающемуся, если количество правильных ответов на тестовые вопросы – 75–60 % от общего объёма заданных вопросов;
- оценка **«неудовлетворительно»** выставляется обучающемуся, если количество правильных ответов – менее 60% от общего объёма заданных вопросов.

#### Критерии формирования оценок по результатам выполнения заданий

##### **«Зачтено»:**

- ставится за работу, выполненную полностью без ошибок и недочетов.
- ставится за работу, выполненную полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета, не более трех недочетов.
- ставится за работу, если обучающийся правильно выполнил не менее 2/3 всей работы или допустил не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более трех негрубых ошибок, одной негрубой ошибки и двух недочетов.

**«Не зачтено»** – ставится за работу, если число ошибок и недочетов превысило норму для оценки «удовлетворительно» или правильно выполнено менее 2/3 всей работы.

##### *Виды ошибок:*

- *грубые ошибки: незнание основных понятий, правил, норм; незнание приемов решения задач; ошибки, показывающие неправильное понимание условия предложенного задания.*
- *негрубые ошибки: неточности формулировок, определений; нерациональный выбор хода решения.*
- *недочеты: нерациональные приемы выполнения задания; отдельные погрешности в формулировке выводов; небрежное выполнение задания.*

#### Критерии формирования оценок по зачету с оценкой

**«Отлично/зачтено»** – студент приобрел необходимые умения и навыки, продемонстрировал навык практического применения полученных знаний, не допустил логических и фактических ошибок

**«Хорошо/зачтено»** – студент приобрел необходимые умения и навыки, продемонстрировал навык практического применения полученных знаний; допустил незначительные ошибки и неточности.

**«Удовлетворительно/зачтено»** – студент допустил существенные ошибки.

**«Неудовлетворительно/не зачтено»** – студент демонстрирует фрагментарные знания изучаемого курса; отсутствуют необходимые умения и навыки, допущены грубые ошибки.

## Экспертный лист

оценочных материалов для проведения промежуточной аттестации по дисциплине «Основы программирования»

по направлению подготовки/специальности

**27.03.05 Инноватика**  
(код и наименование)

Направленность (профиль)/специализация

**Управление инновациями**  
(наименование)

**Бакалавр**  
квалификация выпускника

1. Формальное оценивание			
Показатели	Присутствуют	Отсутствуют	
Наличие обязательных структурных элементов:			
– титульный лист	+		
– пояснительная записка	+		
– типовые оценочные материалы	+		
– методические материалы, определяющие процедуру и критерии оценивания	+		
Содержательное оценивание			
Показатели	Соответствует	Соответствует частично	Не соответствует
Соответствие требованиям ФГОС ВО к результатам освоения программы	+		
Соответствие требованиям ОПОП ВО к результатам освоения программы	+		
Ориентация на требования к трудовым функциям ПС (при наличии утвержденного ПС)	+		
Соответствует формируемым компетенциям, индикаторам достижения компетенций	+		

Заключение: ФОС рекомендуется/ не рекомендуется к внедрению; обеспечивает/ не обеспечивает объективность и достоверность результатов при проведении оценивания результатов обучения; критерии и показатели оценивания компетенций, шкалы оценивания обеспечивают/ не обеспечивают проведение всесторонней оценки результатов обучения.

Эксперт, должность, ученая степень, ученое звание \_\_\_\_\_ /

(подпись)