

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Попов Анатолий Николаевич
Должность: директор
Дата подписания: 16.05.2024 10:57:53
Уникальный программный ключ:
1e0c38dcaae73cee1e5e09c1d5873fc7497ba8

Приложение 2
к рабочей программе дисциплины

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Объектно-ориентированное программирование *(наименование дисциплины(модуля))*

Направление подготовки / специальность

27.03.05 Инноватика
(код и наименование)

Направленность (профиль)/специализация

Управление инновациями на транспорте
(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
ОПК-10.1. Разрабатывает алгоритмы и программы на языке программирования высокого уровня	ОПК-10.1.1. Знать: <ul style="list-style-type: none"> – понятие алгоритма и классификацию алгоритмических языков и систем программирования; – язык программирования высокого уровня - Си;
	ОПК-10.1.2. Уметь: <ul style="list-style-type: none"> – применять на практике современные технологии разработки программ, языки программирования, методы тестирования, отладки и решения задач на ЭВМ; – Разрабатывать программы в соответствии с выбранной моделью жизненного цикла;
	ОПК-10.1.3. Владеть: <ul style="list-style-type: none"> – навыками алгоритмизации и программной реализации на языке высокого уровня решений практических задач; – навыками разработки программ в водопадной модели жизненного цикла
ПК-4.1 Проводит модульное тестирование программного обеспечения ИС, интеграционное тестирование	ПК-4.1.1. Знать: преимущества использования объектно-ориентированного программирования при создании сложных проектов
	ПК-4.1.2. Уметь: пользоваться различными средствами разработки ООП
	ПК-4.1.3. Владеть: навыками практической работы в объектно-ориентированных средах (в том числе визуальных)

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

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине	Оценочные материалы
ОПК-10.1. Разрабатывает алгоритмы и программы на языке программирования высокого уровня	ОПК-10.1.1. Знать: <ul style="list-style-type: none"> – понятие алгоритма и классификацию алгоритмических языков и систем программирования; – язык программирования высокого уровня - Си; 	тест №6 - №9
	ОПК-10.1.2. Уметь: <ul style="list-style-type: none"> – применять на практике современные технологии разработки программ, языки программирования, методы тестирования, отладки и решения задач на ЭВМ; – Разрабатывать программы в соответствии с выбранной моделью жизненного цикла; 	Задания 4-7
	ОПК-10.1.3. Владеть: <ul style="list-style-type: none"> – навыками алгоритмизации и программной 	Задания 12-15

	<p>реализации на языке высокого уровня решений практических задач;</p> <ul style="list-style-type: none"> – навыками разработки программ в водопадной модели жизненного цикла 	
ПК-4.1 Проводит модульное тестирование программного обеспечения ИС, интеграционное тестирование	ПК-4.1.1. Знать: преимущества использования объектно-ориентированного программирования при создании сложных проектов	Задания 16-20
	ПК-4.1.2. Уметь: пользоваться различными средствами разработки ООП	Задания 21-30
	ПК-4.1.3. Владеть: навыками практической работы в объектно-ориентированных средах (в том числе визуальных)	Задания 31-40

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

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

2. Типовые¹ 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
ОПК-10.1.1.	Обучающийся знает: <ul style="list-style-type: none">– понятие алгоритма и классификацию алгоритмических языков и систем программирования;– язык программирования высокого уровня - Си;
<p>Примерные вопросы</p> <p>1. Значение выражения $S := 3 * 7 / 2 \% 7 / 3$ 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 <= 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 :int A [6] имеют значения: \$20, \$10, 8, 4, 2, 1. Чему равно значение выражения: $A[A[5]] + A[1] \% A[4]$? 1) 0 2) 34 3) 36 4) 22</p> <p>4. Дана программа <code>include<stdio.h></code> <code>int nod (int a, int b)</code> { while (a!=b) if (a>b) a= a – b; else b= b - a; return a; }; <code>void main()</code> { <code>int n,m,a;</code> m=8; n=6; a=nod (m,n)); <code>printf(“%d”,a)</code> }</p>	

¹ 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.

Что будет выведено на экран?

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

ОПК-10.1.2

. Уметь:

- применять на практике современные технологии разработки программ, языки программирования, методы тестирования, отладки и решения задач на ЭВМ;
- Разрабатывать программы в соответствии с выбранной моделью жизненного цикла;

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?

1) 18 2) 22 3) 8 4) 20

ОПК-10.1.3	Обучающийся владеет: – навыками разработки программ в водопадной модели жизненного цикла																													
<p>1. Написать программу, выводящую следующее:</p> <table border="1" data-bbox="197 244 367 510"> <tr><td>9</td></tr> <tr><td>88</td></tr> <tr><td>777</td></tr> <tr><td>6666</td></tr> <tr><td>55555</td></tr> </table> <p>2. Найти <i>n</i>-ное по порядку простое число, следующее за введенным числом <i>X</i>.</p> <p>3. 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.</p> <p>4. Заполнить массив <i>A</i> размером <i>b</i> × <i>4</i> следующим способом:</p> <table border="1" data-bbox="609 703 1082 987"> <tr><td>1</td><td>12</td><td>13</td><td>24</td></tr> <tr><td>2</td><td>11</td><td>14</td><td>23</td></tr> <tr><td>3</td><td>10</td><td>15</td><td>22</td></tr> <tr><td>4</td><td>9</td><td>16</td><td>21</td></tr> <tr><td>5</td><td>8</td><td>17</td><td>20</td></tr> <tr><td>6</td><td>7</td><td>18</td><td>19</td></tr> </table>		9	88	777	6666	55555	1	12	13	24	2	11	14	23	3	10	15	22	4	9	16	21	5	8	17	20	6	7	18	19
9																														
88																														
777																														
6666																														
55555																														
1	12	13	24																											
2	11	14	23																											
3	10	15	22																											
4	9	16	21																											
5	8	17	20																											
6	7	18	19																											

2.2 Типовые задания для оценки навыкового образовательного результата

Проверяемый образовательный результат

Код и наименование индикатора достижения компетенции	Образовательный результат
ОПК-10.1.2.	Уметь: применять на практике современные технологии разработки программ, языки программирования, методы тестирования, отладки и решения задач на ЭВМ; – Разрабатывать программы в соответствии с выбранной моделью жизненного цикла;
5. Найти все совершенные числа, меньшие заданного числа <i>n</i> . Число называется совершенным, если равно сумме всех своих положительных делителей, кроме самого этого числа. Например, 28 – совершенно, т.к. $28 = 1 + 2 + 4 + 7 + 14$. 6. Определить, является ли введенная пользователем строка записью натурального числа. Пример: '123' – является, а '34_4f' – не является. 7. Дан массив данных об автомобилях заводского гаража (формирование происходит путем чтения массива из текстового файла): марка, номер, год выпуска, пробег, фамилия шофера. Вывести: а) список машин, подготовленных написание (старше заданного года выпуска и с пробегом больше заданной величины) и их количество; б) сообщение о том, есть ли автомобиль заданной марки (да/нет); в) марки автомобилей (без учета модификаций) с наибольшей величиной среднего пробега. В программе должен быть предусмотрен диалог с пользователем.	
ОПК-10.1.1.	Знает: – понятие алгоритма и классификацию алгоритмических языков и систем программирования; – язык программирования высокого уровня - Си;

8. Найти все плоские числа, меньшие заданного числа n . Составное число называется плоским, если оно представимо в виде произведения двух сомножителей.

Например: 4, 6, 8, 9, 10, 12, 14, 15, ...

9. Найти в строке подстроку максимальной длины, состоящей из одного символа.

10. 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:

- а) фамилии и температуру больных, у кого ее значение за пределами нормы (норма: $36 < t < 37$);
- б) сообщение о том, поступил ли кто-нибудь в палату в заданный день (да или нет);
- в) количество больных, которые лежат дольше 10 дней (на данную дату).

11. Написать рекурсивную программу перевода числа из десятичной системы счисления в двоичную.

ОПК-10.1.3.	Обучающийся владеет: <ul style="list-style-type: none"> – навыками алгоритмизации и программной реализации на языке высокого уровня решений практических задач; – навыками разработки программ в водопадной модели жизненного цикла 																													
12. Написать программу, выводящую:	<table border="1" style="margin-left: auto; margin-right: auto;"> <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>13. Найти n-ое по порядку положительное число кратное 7 и меньшее либо равное введенному числу X.</p> <p>14. 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>15. Заполнить массив A размером 6×4 следующим способом:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>1</td><td>7</td><td>13</td><td>19</td></tr> <tr><td>2</td><td>8</td><td>14</td><td>20</td></tr> <tr><td>3</td><td>9</td><td>15</td><td>21</td></tr> <tr><td>4</td><td>10</td><td>16</td><td>22</td></tr> <tr><td>5</td><td>11</td><td>17</td><td>23</td></tr> <tr><td>6</td><td>12</td><td>18</td><td>24</td></tr> </table>	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
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																											
	–																													

Код и наименование индикатора достижения компетенции	Образовательный результат
ПК-4.1.1.	ПК-4.1.1. Знать: преимущества использования объектно-ориентированного программирования при создании сложных проектов

Примерные вопросы

1. Значение выражения $S := 3 * 7 / 2 \% 7 / 3$ равно:

1) 10 2) 1 3) 1.5 4) 0.5

2. Вычисляемое в программе

```
s:= 1; n = 1;
```

```
for ( i= 2; i<=n; i++) s= s + 1/i;
```

значение переменной S равно:

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

3. Элементы массива A :int A [6] имеют значения: \$20, \$10, 8, 4, 2, 1.

Чему равно значение выражения: $A[A[5]] + A[1] \% A[4]$?

1) 0 2) 34 3) 36 4) 22

4. Дана программа

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

```
int nod (int a, int b )
```

```
{
```

```
    while (a!=b)
```

```
        if (a>b)
```

```
            a= a - b;
```

```
        else
```

```
            b= b - a;
```

```
        return a;
```

```
    };
```

```
void main()
```

```
{
```

```
intn,m,a;
```

```
    m=8; n=6;
```

```
    a=nod (m,n));
```

```
printf(“%d”,a)
```

```
}
```

Что будет выведено на экран?

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-4.1.2

ПК-4.1.2. Уметь: пользоваться различными средствами разработки ООП

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?

1) 18 2) 22 3) 8 4) 20

ПК-4.1.3

– ПК-4.1.3. Владеть: навыками практической работы в объектно-ориентированных средах (в том числе визуальных)

16. Написать программу, выводящую следующее:

```
9
88
777
6666
55555
```

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

18. 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.

19. Заполнить массив A размером $b \times 4$ 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. Создание динамических списков, очередей, стеков, бинарных деревьев, пирамид
2. Понятие хеширование, хеш-таблица, хеш-функция, коллизия. Открытое и закрытое хеширование.
3. Создание массивов: треугольного, разреженного.
4. Понятие программного обеспечения (ПО), его жизненный цикл. Методы разработки ПО. Требование к программе. Источники ошибок. Качество ПО.
5. Принцип ООП. Свойства ООП (абстрагирование, инкапсуляция, наследование, полиморфизм).
Понятие класс, объект.
6. Методы класса (конструкторы, деструкторы и др. методы). Перегрузка методов.
7. Дружественные функции. Дружественные классы
- 8 Локальные, глобальные, статические переменные. Статические методы.
9. Массивы объектов
10. Указатели на объекты и на члены класса.
11. Константы в классе: методы, объекты. Указатель на константу. Константный указатель.
10. Операторы унарные и бинарные, их перегрузка. Ограничения. Операторная функция.
11. Перегрузка операторов с помощью дружественной функции. Ограничения для дружественных функций.
12. Управление доступом к параметрам базового класса. Восстановление уровня доступа при закрытом наследовании.
- 13 Виды наследования: иерархическое, один ко многим, многие к одному, ромбовидное (виртуальные базовые классы).
- 14 Вызов конструкторов и деструкторов при наследовании. Передача параметров конструкторам базовых классов, через конструкторы производных классов.
15. Полиморфизм: статический, динамический. Виртуальные функции. Требование к виртуальным функциям, их наследование. Виртуальные деструкторы. Запрет наследования.
16. Абстрактные классы. Интерфейсы. Классы реализации.
17. Ранее и позднее связывание. Механизм.
18. Динамическая идентификация типов.
19. Операторы приведения типов.
20. Понятие «Поток». Классы, реализующие поток. преимущества, недостатки. Стандартные потоки.
21. Форматирование данных: флаги, методы, манипуляторы.
22. Неформатированные методы обмена с потоком.
23. Файловые потоки (открытие, чтение/запись).
24. Параметрический полиморфизм. Обобщенные классы и функции. Стандартные типы в обобщенных классах.
25. Аргументы по умолчанию в обобщенных классах. Явная специализация.
26. Исключительная ситуация. Операторы try, catch, throw. Класс exception, метод what.
27. Принцип построения оконного приложения с классическим окном.
28. Графические окна, их свойства
29. Управляющие элементы. Создание.
30. Контекст устройства
31. Объекты оконной графики. Виртуальное окно.
32. Таймер
33. Встроенный ассемблер, преимущества, недостатки. Соглашения о вызовах подпрограмм. Формирование стека.
- 34 Понятие case-технологии.
35. Универсальный язык моделирования. Диаграммы классов.
36. Понятия: информационная технология, информационная система, информационно-

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 «Программирование»

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)