

Документ подписан простой электронной подписью

Информация о владельце:

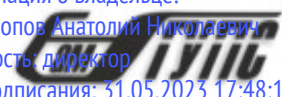
ФИО: Попов Анатолий Николаевич

Должность: директор

Дата подписания: 31.05.2023 17:48:15

Уникальный программный ключ:

1e0c38dccc0aee71c2e1e5c09d1d5875tc7497bc8



МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

САМАРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ

Приложение 2
к рабочей программе дисциплины

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Системное программирование

(наименование дисциплины(модуля))

Направление подготовки / специальность

09.03.03 Прикладная информатика

(код и наименование)

Направленность (профиль)/специализация

Прикладная информатика на железнодорожном 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 |
|---|
| ПК-1 Способен разрабатывать требования и проектировать программное обеспечение |
| ПК-1.1 Разрабатывает программный код и осуществляет отладку на языках программирования низкого уровня |
| ПК-1.2 Разрабатывает программный код и осуществляет отладку на языках программирования высокого уровня |
| ПК-4 Способен разрабатывать компоненты системных программных продуктов |
| ПК-4.1 Разрабатывает компоненты системных программных продуктов с использованием современных инструментальных средств |
| ПК-4.2 Разрабатывает документацию на системные утилиты |

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 программный код и осуществляет отладку на языках программирования низкого уровня | ПК-1.1.1 Знать: понятие алгоритма и классификацию алгоритмических языков и систем программирования; | Тест 1 - 5 |
| | 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 |
| ПК-4.1 Разрабатывает компоненты системных программных продуктов с использованием современных инструментальных средств | ПК-4.1.1 Обучающийся знает: основные модели системных процессов | Тест 10-19 |
| | ПК-4.1.2 Обучающийся умеет: разрабатывать программы на уровне моделей системных | Задания 16-18 |

| | | |
|--|--|---------------|
| | процессов | |
| | ПК-4.1.3 Обучающийся владеет: шаблонами проектирования системного программного обеспечения | Задания 19-22 |
| ПК-4.2 Разрабатывает документацию на системные утилиты | ПК-4.2.1 Обучающийся знает: Методы функционального проектирования. Технические требования к пользовательскому интерфейсу. Архитектуру целевой аппаратной платформы, для которой разрабатывается программное обеспечение. | Тест 20-29 |
| | ПК-4.2.2 Обучающийся умеет: планировать проектные работы; выбирать методики разработки требований к системе и шаблоны документов требований к системе; формулировать цели, исходя из анализа проблем, потребностей и возможностей. Разрабатывать графический дизайн интерфейсов. | Задания 23-25 |
| | ПК-4.2.3 Обучающийся владеет: Навыками создания графических документов в программах растровых и векторных изображений. Разработки блок-схемы разрабатываемых систем. Реинжиниринга разработанного программного обеспечения для решения технических задач | Задания 26-28 |

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

- 1) ответ на билет, состоящий из теоретических вопросов и практических заданий;
- 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 знаниевого образовательного результата

| Код и наименование индикатора достижения компетенции | Образовательный результат |
|--|--|
| ПК-1.1.1 | Обучающийся знает: <ul style="list-style-type: none"> - понятие алгоритма и классификацию алгоритмических языков и систем программирования; |
| <p>Примерные вопросы</p> <p>1. Значение выражения $S := 3 * 7 / 2 \% 7 / 3$ равно:</p> <p>1) 10 2) 1 3) 1.5 4) 0.5</p> <p>2. Вычисляемое в программе</p> <pre>s:= 1; n = 1; for (i= 2; i<=n; i++) s= s + 1/i;</pre> <p>значение переменной S равно:</p> | |

¹Приводятся 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) 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()
{
    int n,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-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?

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

ПК-4.1.1

Обучающийся знает: основные модели системных процессов

Вопрос к заданиям 10 -19. Дан фрагмент программы на ассемблере:

```
push BX
push CX
push DX
push AX
mov AX,3508h
add BX,AX
pop AX
pop DX
mov DX,CX
shlBX,1
notAX
```

В каждом задании даны начальные значения регистров AX, BX, CX, DX. Определить значения этих регистров после выполнения данного фрагмента.

10. AX=0000h, BX=0CE5h, CX=0007h, DX=0023h

11. AX=0427h, BX=5ACEh, CX=0000h, DX=000Fh

12. AX=1010h, BX=0000h, CX=0DACH, DX=9990h

13. AX=0DEDh, BX=0444h, CX=1649h, DX=0000h

14. AX=0754h, BX=0431h, CX=0001h, DX=43Ach

15. AX=5325h, BX=0A11h, CX=9999h, DX=0101h

16. AX=0002h, BX=09ABh, CX=3333h, DX=0407h

17. AX=0AAAh, BX=3DEFh, CX=6109h, DX=0C0Ch

18. AX=35ADh, BX=1000h, CX=2100h, DX=0C03h

ПК-4.2.1

Обучающийся знает: Методы функционального проектирования. Технические требования к пользовательскому интерфейсу. Архитектуру целевой аппаратной платформы, для которой разрабатывается программное обеспечение.

20 Легкость применения программного обеспечения это:

- а) характеристики ПО, позволяющие минимизировать усилия пользователя по подготовке исходных данных, применению ПО;
- б) отношение уровня услуг, предоставляемых ПО пользователю при заданных условиях, к объему используемых ресурсов;
- в) характеристики ПО, позволяющие минимизировать усилия по внесению изменений для устранения в нем ошибок и по его модификации.

21 Мобильность программного обеспечения это:

- а) способность ПО выполнять набор функций, которые удовлетворяют потребности пользователей;
- б) способность ПС безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени;
- в) способность ПО быть перенесенным из одной среды (аппаратного / программного) в другое.

22 Что из приведенного является критериями оценки удобства интерфейсов?

- а) скорость обучения;
- б) адаптация к стилю работы пользователя;
- в) все ответы правильные.

23 Артефакт — это

- а) любой продукт деятельности специалистов по разработке программного обеспечения;
- б) результат ошибок разработчика во входных или проектных спецификациях;
- в) графическое представление элементов моделирования системы.

24 Укажите правильную последовательность этапов при каскадной модели жизненного цикла:

- а) Определение требований -> Тестирование -> Реализация;
- б) Проектирование -> Реализация -> Тестирование;
- в) Проектирование -> Определение требований -> Реализация.

25 Устойчивость программного обеспечения — это:

- а) свойство, характеризующее способность ПС завершать автоматически корректное функционирование ПК, несмотря на неправильные (ошибочные) входные данные;
- б) свойство, способна противостоять преднамеренным или непреднамеренным деструктивным действиям пользователя;
- в) свойство, характеризующее способность ПС продолжать корректное функционирование, несмотря на неправильные (ошибочные) входные данные.

26 UML — это:

- а) язык программирования, имеющий синтаксис схож с C ++;
- б) унифицированный язык визуального моделирования, использует нотацию диаграмм;
- в) набор стандартов и спецификаций качества программного обеспечения.

27 При конструировании программного обеспечения на этапе разработки или выбора алгоритма решения реализуется следующее:

- а) архитектурное обработки программы;
- б) выбор языка программирования;
- в) совершенствование программы.

28 Проектирование ПО в основном рассматривается как

- а) архитектурное проектирование;
- б) коммуникационные методы;
- в) детальные методы.

29 На этапе тестирования пользователь выполняет следующее:

- а) синтаксическое отладки;
- б) выбор тестов и метода тестирования;
- в) определение формы выдачи результатов.

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">- применять на практике современные технологии разработки алгоритмов, языки программирования, методы тестирования, отладки и решения задач на ЭВМ; | | | | | |
| | <ol style="list-style-type: none">1. Найти все совершенные числа, меньшие заданного числа n. Число называется совершенным, если равно сумме всех своих положительных делителей, кроме самого этого числа. Например, 28 – совершенно, т.к. $28 = 1 + 2 + 4 + 7 + 14$.2. Определить, является ли введенная пользователем строка записью натурального числа. Пример: '123' – является, а '34_4f' – не является.3. Дан массив данных об автомобилях заводского гаража (формирование происходит путем чтения массива из текстового файла): марка, номер, год выпуска, пробег, фамилия шофера. Вывести:<ul style="list-style-type: none">а) список машин, подготовленных на списание (старше заданного года выпуска и с пробегом больше заданной величины) и их количество;б) сообщение о том, есть ли автомобиль заданной марки (да или нет);с) марки автомобилей (без учета модификаций) с наибольшей величиной среднего пробега. В программе должен быть предусмотрен диалог с пользователем. | | | | | |
| ПК-1.1.3 | Обучающийся владеет: <ul style="list-style-type: none">- навыками алгоритмизации и программной реализации на языке высокого уровня решений практических задач; | | | | | |
| | <p>4. Написать программу, выводящую:</p> <table border="1" data-bbox="979 1491 1099 1753"><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> <ol style="list-style-type: none">5. Найти n-ое по порядку положительное число кратное 7 и меньшее либо равное введенного числа X.6. Даны два целочисленных массива, состоящие из одинакового числа элементов. Получить третий массив той же размерности, каждый элемент которого равен сумме соответствующих элементов данных массивов. | 1 | 21 | 321 | 4321 | 54321 |
| 1 | | | | | | |
| 21 | | | | | | |
| 321 | | | | | | |
| 4321 | | | | | | |
| 54321 | | | | | | |

7. Заполнить массив A размером 6x4 следующим способом:

| | | |
|---|----|----|
| 1 | 7 | 13 |
| 2 | 8 | 14 |
| 3 | 9 | 15 |
| 4 | 10 | 16 |
| 5 | 11 | 17 |
| 6 | 12 | 18 |

ПК-1.2.2

Обучающийся умеет:

- Разрабатывать программы в соответствии с выбранной моделью жизненного цикла;

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/нет);

в) количество больных, которые лежат дольше 10 дней (на данную дату).

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

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 A размером 6x4 следующим способом:

| | | |
|---|----|----|
| 1 | 12 | 13 |
| 2 | 11 | 14 |

| | | | |
|---|---|----|----|
| | 3 | 10 | 15 |
| | 4 | 9 | 16 |
| | 5 | 8 | 17 |
| | 6 | 7 | 18 |
| ПК-4.1.2 | Обучающийся умеет: разрабатывать программы на уровне моделей системных процессов | | |
| <p>Задание 16. Провести анализ проблемы. На этапе анализа проблемы провести анализ предметной области, для которой разрабатывается ПО.</p> <p>1) определить границы, или контура, системы;</p> <p>2) описать объектов автоматизации и/или формализации знаний об этих объектах;</p> <p>3) выявить или определить потребностей заказчика ПО.</p> <p>Задание 17. Для выявления потребностей заказчика и описания объектов автоматизации необходимо провести анкетирование заказчиков ПО.</p> <p>Задание 18. Постановка задачи.</p> <p>На этапе постановки задачи разработать:</p> <p>1) перечень заинтересованных лиц;</p> <p>2) список потребностей заинтересованных лиц в разрабатываемом ПО.</p> | | | |
| ПК-4.1.3 | Обучающийся владеет: шаблонами проектирования системного программного обеспечения | | |
| <p>Задание 19. Разработать модели вариантов использования и их спецификаций.</p> <p>Задание 20. Оформить технического задания в соответствии с ГОСТ 34.602–89.</p> <p>Задание 21. Реализовать архитектуру ПО на базе объектно-реляционного отображения.</p> <p>Задание 22. Разработать простое MDA-приложения.</p> | | | |
| ПК-4.2.2 | Обучающийся умеет: планировать проектные работы; выбирать методики разработки требований к системе и шаблоны документов требований к системе; формулировать цели, исходя из анализа проблем, потребностей и возможностей. Разрабатывать графический дизайн интерфейсов. | | |
| <p>Задания:</p> <p>23 Тема «Разбиение на лексемы транслируемой команды»</p> <p>Задание: определить лексический набор команды как упакованный массив или строковый тип. Разделить строку на лексемы по указанной группе разделителей</p> <p>24 Тема «Построение бинарного несбалансированного дерева»</p> <p>Задание: определить структуру записи листа дерева с учетом идентификатора лексемы, ее системного признака и полей организации доступа к потомкам; построить дерево и организовать вывод ассоциативной таблицы связей.</p> <p>25 Тема «Одномерный поток»</p> <p>Задание: конструктор потока, запуск потока, индикация этапов задачи до потока во время его работы и после завершения потока.</p> | | | |
| ПК-4.2.3 | Обучающийся владеет: Навыками создания графических документов в программах растровых и векторных изображений. Разработки блок-схемы разрабатываемых систем. Реинжиниринга разработанного программного обеспечения для решения технических задач | | |
| <p>Задания:</p> <p>26 Вывод потоком на консоль собственного и системного идентификатора; вывод имени потока и основных характеристик его системных ресурсов (адрес, приоритет и т.п.)</p> <p>Содержание задания:</p> <ul style="list-style-type: none"> - подготовить конструкторы асинхронных потоков - передать не менее 2-х параметров в процедуры потоков посредством объекта-структуры - в составе параметров определить имя или идентификатор потока - считать системный идентификатор потока, приоритет потока. Вывести информационное сообщение | | | |

на консоль.

27 Организация потоков в массив: регистрация потоков в динамической коллекции с распознаванием идентификатора или имени потока при прохождении по коллекции независимым итератором

28 Тема «Параллельные асинхронные потоки»

Задание: конструктор фиксированного числа потоков, запуск потоков, идентификация асинхронной работы потоков; передача структуры с параметрами потока по нетипированному адресу.

2.3. Перечень вопросов для подготовки обучающихся к промежуточной аттестации

1. Понятие системного программного продукта.
2. Стадии разработки программного обеспечения.
3. Технология инкрементального программирования.
4. Понятие объекта: состояние, поведение, методы. Понятие класса и идентичность объектов.
5. Принципы объектной модели.
6. Проблема создания и уничтожения объектов.
7. Понятие ссылки. Операции над ссылками и способы применения ссылок.
8. Статические и постоянные члены класса. Встраиваемые методы.
9. Дружественные функции.
10. Понятие полиморфизма в программировании. Виды полиморфизма в среде разработки Delphi.
11. Иерархические отношения между классами: композиция.
12. Иерархические отношения между классами: наследование.
13. Виртуальные методы и позднее связывание.
14. Понятие абстрактного класса.
15. Использование потокового ввода-вывода.
16. Порождаемые функции в обобщённом программировании.
17. Порождаемые классы в обобщённом программировании.
18. Использование основных контейнеров стандартной библиотеки.
19. Классы функциональных объектов: создание и применение.
20. Использование алгоритмов стандартной библиотеки.
21. Парадигма защитного программирования.
22. Понятие операционной системы, ее назначение и функции.
23. Утилиты операционной системы GNU/Linux.
24. Понятие файла. Виртуальная файловая система ОС GNU/Linux.
25. Разграничение доступа к файлам. Доступ к устройствам.
26. Организация взаимодействия пользователя с системой.
27. Организация ввода-вывода системного уровня.
28. Концепция процесса. Состояния процесса и диаграмма смены состояний. Операции над процессами.
29. Планирование процессов. Политики планирования.
30. Сигналы и их использование для управления процессами.
31. Использование времени в программных системах. Часы POSIX. Таймеры.
32. Понятие потока выполнения. Сравнительный анализ потоков и процессов.
33. Асинхронные параллельные потоки.
34. Задача «разделение доступа к ресурсу». Способы ее решения.
35. Задача «синхронизация по готовности данных». Способы ее решения.
36. Системные средства синхронизации потоков.
37. Мониторы Хоара.
38. Проектирование и модели многопоточных приложений.
39. Иерархия памяти. Виртуальная память: концепция и инструменты.
40. Взаимодействие процессов: проблематика и средства. Каналы.
41. Сокеты и их применение для взаимодействия процессов.
42. Проблематика разработки многопоточных приложений.

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

Критерии формирования оценок по результатам выполнения заданий

«Отлично/зачтено» – ставится за работу, выполненную полностью без ошибок и недочетов.

«Хорошо/зачтено» – ставится за работу, выполненную полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета, не более трех недочетов.

«Удовлетворительно/зачтено» – 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.

«Неудовлетворительно/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.

Критерии формирования оценок по экзамену

«Отлично» (5 баллов) – обучающийся демонстрирует знание всех разделов изучаемой дисциплины: содержание базовых понятий и фундаментальных проблем; умение излагать программный материал с демонстрацией конкретных примеров. Свободное владение материалом должно характеризоваться логической ясностью и четким видением путей применения полученных знаний в практической деятельности, умением связать материал с другими отраслями знания.

«Хорошо» (4 балла) – обучающийся демонстрирует знания всех разделов изучаемой дисциплины: содержание базовых понятий и фундаментальных проблем; приобрел необходимые умения и навыки, освоил вопросы практического применения полученных знаний, не допустил фактических ошибок при ответе, достаточно последовательно и логично излагает теоретический материал, допуская лишь незначительные нарушения последовательности изложения и некоторые неточности. Таким образом данная оценка выставляется за правильный, но недостаточно полный ответ.

«Удовлетворительно» (3 балла) – обучающийся демонстрирует знание основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем. Однако знание основных проблем курса не подкрепляется конкретными практическими примерами, не полностью раскрыта сущность вопросов, ответ недостаточно логичен и не всегда последователен, допущены ошибки и неточности.

«Неудовлетворительно» (0 баллов) – выставляется в том случае, когда обучающийся демонстрирует фрагментарные знания основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем. У экзаменуемого слабо выражена способность к самостоятельному аналитическому мышлению, имеются затруднения в изложении материала, отсутствуют необходимые умения и навыки, допущены грубые ошибки и незнание терминологии, отказ отвечать на дополнительные вопросы, знание которых необходимо для получения положительной оценки.

дисциплине «Системное программирование»

Направление подготовки / специальность

09.03.03 Прикладная информатика

(код и наименование)

Направленность (профиль)/специализация

Прикладная информатика на железнодорожном транспорте

(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)