

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Ефанов Алексей Валерьевич  
Должность: Директор Невиномысского технологического института (филиал) СКФУ  
Дата подписания: 10.10.2022 17:22:55  
Уникальный программный ключ:  
49214306dd433e7a1b0f8632f645f9d53c99e3d0

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**УТВЕРЖДАЮ**  
Директор НТИ (филиал) СКФУ  
Ефанов А.В.

\_\_\_\_\_  
Ф.И.О.  
« \_\_\_\_ » \_\_\_\_\_ 2022 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

для проведения текущего контроля успеваемости и промежуточной аттестации по  
дисциплине

Информационные технологии и программирование

Направление подготовки	15.03.02 Технологические машины и оборудование
Направленность (профиль)	Технологическое оборудование химических и нефтехимических производств
Год начала обучения	2022
Форма обучения	Заочная
Реализуется на 2 курсе	

## Введение

1. Назначение: обеспечение методической основы для организации и проведения текущего контроля по дисциплине «Информационные технологии и программирование». Текущий контроль по данной дисциплине – вид систематической проверки знаний, умений, навыков студентов. Задачами текущего контроля являются получение первичной информации о ходе и качестве освоения компетенций, а также стимулирование регулярной целенаправленной работы студентов. Для формирования определенного уровня компетенций.

2. ФОС является приложением к программе дисциплины «Информационные технологии и программирование» и в соответствии с образовательной программой высшего образования по направлению подготовки 15.03.02 Технологические машины и оборудование .

3. Разработчик: Кочеров Юрий Николаевич, доцент базовой кафедры Регионального индустриального парка, кандидат технических наук

4. Проведена экспертиза ФОС.

Члены экспертной группы:

Председатель:

Павленко Е.Н.–зав. кафедрой ХТМиАХП

Члены экспертной группы:

Романенко Е.С. – доцент кафедры ХТМиАХП

Свидченко А.И. – доцент кафедры ХТМиАХП

Представитель организации-работодателя:

Новоселов А.М., начальник отдела технического развития АО «Невинномысский Азот»

Экспертное заключение: фонд оценочных средств соответствует ОП ВО по направлению подготовки 15.03.02 Технологические машины и оборудование и рекомендуется для оценивания уровня сформированности компетенций при проведении текущего контроля успеваемости и промежуточной аттестации студентов по дисциплине «Информационные технологии и программирование».

05 марта 2022 г.

5. Срок действия ФОС определяется сроком реализации образовательной программы.

## 1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Код оцениваемой компетенции, индикатора (ов)	Этап формирования компетенции (№ темы) (в соответствии с рабочей программой дисциплины)	Средства и технологии оценки	Вид аттестация контроля, (текущий/промежуточный)	Тип контроля (устный, письменный или с использованием технических средств)	Наименование оценочного средства
ИД-1 ОПК-4 ИД-1 ОПК-14 ИД-2 ОПК-14 ИД-3 ОПК-14	1-15	Собеседование	Текущий	Устный	Вопросы для собеседования
ИД-1 ОПК-4 ИД-1 ОПК-14 ИД-2 ОПК-14 ИД-3 ОПК-14	1-15	Экзамен	Промежуточный	Устный	Вопросы к экзамену

## 2. Описание показателей и критериев оценивания на различных этапах их формирования, описание шкал оценивания

Уровни сформированности компетенции(ий), индикатора (ов)	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
ОПК-4. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности				
Результаты обучения по дисциплине (модулю): ИД-1 ОПК-4 ИД-1 ОПК-14 ИД-2 ОПК-14 ИД-3 ОПК-14	<b>Не удовлетворительно понимает</b> основы разработки алгоритмов и компьютерных программ <b>Не удовлетворительно разрабатывает</b> алгоритмы и компьютерные программы	<b>Слабо понимает</b> основы разработки алгоритмов и компьютерных программ <b>Слабо разрабатывает</b> алгоритмы и компьютерные программы <b>Слабо применяет навыки</b>	<b>Понимает</b> основы разработки алгоритмов и компьютерных программ <b>Разрабатывает</b> алгоритмы и компьютерные программы <b>Применяет навыки</b>	<b>Понимает</b> основы разработки алгоритмов и компьютерных программ в профессиональной области <b>Разрабатывает</b> алгоритмы и компьютерные программы, пригодные для

	<b>Не удовлетворительно применяет навыки</b> разработки алгоритмов и компьютерных программ	разработки алгоритмов и компьютерных программ	разработки алгоритмов и компьютерных программ	практического применения <b>Применяет</b> навыки разработки алгоритмов и компьютерных программ, пригодных для практического применения
--	--	---	---	---

### Описание шкалы оценивания

В рамках рейтинговой системы успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации.

### Текущий контроль

Текущий контроль осуществляется в течение семестра - на лабораторных и практических занятиях, по которым рабочими программами дисциплин предусмотрены отчетности

### Промежуточная аттестация

Промежуточная аттестация в форме экзамена предусматривает проведение обязательной экзаменационной процедуры

Процедура зачета (зачета с оценкой) как отдельное контрольное мероприятие не проводится, оценивание знаний обучающегося происходит по результатам защиты лабораторных и практических занятий.

## 3. Типовые контрольные задания и иные материалы, характеризующие этапы формирования компетенций

### Вопросы для собеседования

1. Информационные технологии для разработки алгоритмов со сложной логикой.
2. Что такое информационная среда программы?
3. Что такое программное средство (ПС)?
4. Что такое малая и большая системы?
5. Что такое жизненный цикл программного средства (ПС)?
6. Что такое внешнее описание ПС?
7. Что такое сопровождение ПС?
8. Что такое определение требований к программному средству (ПС)?
9. Что такое спецификации качества ПС?
10. Что такое устойчивость (robustness) ПС?
11. Функции

function F(x, y: integer): integer;

function G(x, y: integer): integer;

function R(x, y: integer): integer;

определены с помощью операционной семантики равенствами:

$$R(x, y) = x*(y - 1),$$

$$F(x, y) = R(x + 1, y) - R(x, y - 1),$$

$$G(x, y) = F(x, R(x, y)).$$

Найти значения G(3, 3).

## 12. Функции

function F(n: integer): integer;

function G(n: integer): integer;

определены с помощью операционной семантики равенствами:

$F(0)=1$ ,

$G(0)=2$ ,

$F(n)=G(n-1)$ ,

$G(n)=F(n-1) + G(n-1)$ .

Найти значения F(3) и G(3).

13. Что такое архитектура программного средства?

14. Что такое программный модуль?

15. Что такое прочность программного модуля?

16. Что такое структурное программирование?

17. Что такое триада Хоора?

18. Что такое отладка программного средства?

19. Что такое тестирование программного средства?

20. Что такое автономная отладка программного средства?

21. Что такое защитное программирование?

22. Какие виды защиты программного средства от искажения информации Вы знаете?

23. Какие задачи приходится решать при обеспечении коммуникабельности ПС?

24. Какие возможности предоставляет пользователю графический пользовательский интерфейс?

25. Как нужно действовать для обеспечения эффективности ПС?

26. Что такое инсталлятор программного средства (ПС)?

27. Что такое менеджер программного средства?

28. Что такое ординарный пользователь программного средства?

29. Что такое администратор программного средства?

30. Что такое управление разработкой ПС?

31. Что такое менеджер программного проекта?

32. Что такое неформальная демократическая бригада разработчиков ПС?

33. В чем заключается сущность объектного подхода к разработке программных средств (ПС)?

34. Какие категории объектов можно выделить с точки зрения разработчиков ПС?

35. Что такое объектная модель ПС?

36. Что такое программный инструмент разработки ПС?

37. Что такое аппаратный инструмент разработки ПС?

38. Что такое инструментальная среда разработки и сопровождения ПС?

39. Что такое инструментально-объектный подход к разработке программного средства?

40. Какие признаки классификации инструментальных сред разработки и сопровождения ПС Вы знаете?

41. Что такое интегрированность инструментальной среды разработки и сопровождения ПС?

42. Какие виды интегрированности инструментальной среды разработки и сопровождения ПС Вы знаете?

43. Что такое репозиторий инструментальной среды разработки и сопровождения ПС?

44. Что такое инструментальная среда программирования?

45. Что такое языково-ориентированная инструментальная среда программирования?

46. Что такое жизненный цикл ПС?

47. Основное назначение моделей ЖЦ ПС?

48. Перечислите основные процессы ЖЦ ПС.

49. Назовите вспомогательные процессы ЖЦ ПС.

50. Опишите структуру стандарта ГОСТ ISO/IEC 12207.
51. Перечислите основные подходы организации процессов создания ПС и назовите основные виды моделей ЖЦ ПС.
52. Языки программирования высокого и низкого уровня
53. Что такое ошибка в ПС?
54. Что такое надежность ПС?
55. Что такое технология программирования?
56. Что такое простая и сложная системы?
57. Что такое качество ПС?
58. Что такое смежный контроль?
59. Что такое защищенность (defensiveness) ПС?
60. Что такое коммуникабельность (communicativeness) ПС?
61. Что такое функциональная спецификация ПС?
62. Что такое ручная имитация внешнего описания ПС?
63. Формальные языки E и T определены над алфавитом  $\{ 'a', '*', '&', '<', '>' \}$   
с помощью денотационной семантики равенствами  
 $E = T \cup ' * ' T \cup E ' & ' T,$   
 $T = ' a ' \cup ' a * ' \cup ' < ' E ' > '$   
 Какие из следующих строк  
 $' * a & * a * & a * ' ,$   
 $' * a & < a & a * > ' ,$   
 $' * < * a * & a > & < * a * > * '$   
 принадлежат языку E и какие из них не принадлежат языку E.
64. Тип R определён с помощью следующей аксиоматической семантики.  
 Описания:  
 type R = record P1, P2, P3: CHAR end;  
 function READ(S: R): CHAR; {READ: R → CHAR}  
 function SHIFT(S: R): R; {SHIFT: R → R}  
 function ADD(S: R, C: CHAR): R; {ADD: R \* CHAR → R}  
 function REMOVE(S: R): R; {REMOVE: R → R}  
 var X, Y, Z: CHAR;  
 U: R;  
 Аксиомы:  
 $SHIFT(ADD(ADD(ADD(U, X), Y), Z)) =$   
 $ADD(ADD(ADD(U, Y), Z), X);$   
 $REMOVE(U) = SHIFT(ADD(U, '#'));$   
 $READ(SHIFT(ADD(U, X))) = X;$   
 Найти значение:  
 $READ(SHIFT(SHIFT(REMOVE(ADD(ADD(U, 'a'), 'b'))))) =$
65. Что такое архитектурная функция?
66. Что такое сцепление программного модуля?
67. Что такое пошаговая детализация программного модуля?
68. Что такое псевдокод?
69. Что такое свойство программы?
70. Что такое комплексная отладка программного средства?
71. Что такое ведущий отладочный модуль?
72. Что такое отладочный имитатор программного модуля?
73. Какие требования предъявляются к компьютеру, чтобы можно было обеспечить защиту программы от отказов другой программы в мультипрограммном режиме?
74. Что такое компьютерная подпись?
75. Что такое компьютерная печать?

76. Что такое управление конфигурацией ПС?
77. Что такое ядро ПС?
78. Что такое оболочка ПС?
79. Что такое руководство по инсталляции программного средства?
80. Что такое руководство по управлению программным средством?
81. Что такое руководство по сопровождению программного средства?
82. Что такое бригада ведущего программиста?
83. Что такое смотр программной компоненты (программного документа)?
84. Что такое аттестация ПС?
85. Что такое динамическая модель ПС?
86. Что такое диаграмма состояний класса?
87. Что такое функциональная модель ПС?
88. Что такое компонент ПС?
89. Что такое компьютерная технология (CASE-технология) разработки ПС?
90. Какие отличия жизненного цикла ПС при компьютерной технологии программирования от жизненного цикла ПС при традиционной (ручной) технологии программирования (при водопадном подходе)?
91. Что такое рабочее место компьютерной технологии разработки и сопровождения ПС?
92. Что такое инструментальная система технологии программирования?
93. Что такое языково-зависимая инструментальная система технологии программирования?
94. Что такое ядро инструментальной системы технологии программирования?
95. Что такое встроенный инструмент инструментальной системы технологии программирования?
96. Что такое импортируемый инструмент инструментальной системы технологии программирования?
97. Что такое оболочка инструментальной системы технологии программирования?
98. Опишите суть водопадного подхода разработки ПС.
99. Опишите суть исследовательского программирования.
100. Опишите суть прототипирования при разработке ПС.
101. Опишите основные черты подходов формальных преобразований и сборочного программирования при разработке ПС.
102. Какие общие черты имеют инкрементная и эволюционная модели?
103. Как построить новую модель ЖЦ на основе стандарта ISO/IEC 12207?

### **1. Критерии оценивания компетенций**

Оценка «отлично» выставляется студенту, если он:

Имеет высокие начальные сведения о стандартах проектирования, разработки, тестирования и эксплуатации программного обеспечения;

На высоком уровне выбирает технологию программирования и инструментальные программные средства высокого уровня

На высоком уровне использует основные алгоритмические структуры; знает основные способы записи алгоритмов и конструирования программ с использованием различных алгоритмических языков;

На высоком уровне использует стандартные и собственные структуры данных, базовые и собственные алгоритмы их обработки; использует современные методы и средства разработки алгоритмов при решении задач профессиональной деятельности

На высоком уровне использует современные методы и средства разработки программного обеспечения систем управления

Оценка «хорошо» выставляется студенту, если он:

Получил начальные сведения о стандартах проектирования, разработки, тестирования и эксплуатации программного обеспечения;

Выбирает технологию программирования и инструментальные программные средства высокого уровня

Использует основные алгоритмические структуры; знает основные способы записи алгоритмов и конструирования программ с использованием различных алгоритмических языков;

Использует стандартные и собственные структуры данных, базовые и собственные алгоритмы их обработки; использует современные методы и средства разработки алгоритмов при решении задач профессиональной деятельности

Использует современные методы и средства разработки программного обеспечения систем управления

Оценка «удовлетворительно» выставляется студенту, если он:

Получил слабые начальные сведения о стандартах проектирования, разработки, тестирования и эксплуатации программного обеспечения;

На низком уровне выбирает технологию программирования и инструментальные программные средства высокого уровня

Слабо использует основные алгоритмические структуры; знает основные способы записи алгоритмов и конструирования программ с использованием различных алгоритмических языков;

Слабо использует стандартные и собственные структуры данных, базовые и собственные алгоритмы их обработки; использует современные методы и средства разработки алгоритмов при решении задач профессиональной деятельности

Слабо использует современные методы и средства разработки программного обеспечения систем управления

Оценка «неудовлетворительно» выставляется студенту, если он:

Недостаточно получил начальных сведений о стандартах проектирования, разработки, тестирования и эксплуатации программного обеспечения;

На недостаточном уровне выбирает технологию программирования и инструментальные программные средства высокого уровня

Недостаточно использует основные алгоритмические структуры; знает основные способы записи алгоритмов и конструирования программ с использованием различных алгоритмических языков;

Недостаточно использует стандартные и собственные структуры данных, базовые и собственные алгоритмы их обработки; использует современные методы и средства разработки алгоритмов при решении задач профессиональной деятельности

Недостаточно использует современные методы и средства разработки программного обеспечения систем управления

## **2. Описание шкалы оценивания**

Текущая аттестация в форме собеседования предусматривает защиту выполненных лабораторных и практических работ и оценивается в соответствии с критериями оценивания компетенций

## **3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Пример:

Процедура проведения данного оценочного мероприятия включает в себя:

Предлагаемые студенту вопросы позволяют проверить ИД-1 ОПК-14, ИД-2 ОПК-14, ИД-3 ОПК-14 компетенции.

Для подготовки к данному оценочному мероприятию необходимо 10 минут.



При подготовке к ответу студенту предоставляется право отчетами о выпиленных лабораторных и практических занятиях

При проверке задания, оцениваются последовательность и логика ответа и др.

Бланк оценочного листа собеседования

№ п/п	ФИО студента	Критерий оценивания			Итого
		правильность ответа	полнота раскрытия вопроса	умение аргументировать свой ответ	
1					
2					
...					

### Вопросы к экзамену\*

Вопросы (задача, задание) для проверки уровня обученности

1. Информационные технологии для разработки алгоритмов со сложной логикой.
2. Исторический и социальный контекст программирования
3. Программа как формализованное описание процесса обработки данных.
4. Неконструктивность понятия правильной программы.
5. Интеллектуальные возможности человека.
6. Неправильный перевод как причина ошибок в программных средствах.
7. Специфика разработки программных средств.
8. Жизненный цикл программного средства.
9. Понятие качества программного средства.
10. Обеспечение надежности - основной мотив разработки программных средств.
11. Назначение внешнего описания программного средства и его роль в обеспечении качества программного средства.
12. Определение требований к программному средству.
13. Спецификация качества программного средства.
14. Основные подходы к спецификации семантики функций.
15. Метод таблиц решений.
16. Операционная семантика.
17. Понятие архитектуры программного средства.
18. Основные классы архитектур программных средств.
19. Цель модульного программирования.
20. Основные характеристики программного модуля.
21. Порядок разработки программного модуля.
22. Структурное программирование.
23. Обоснования программ. Формализация свойств программ.
24. Свойства простых операторов.
25. Основные понятия.
26. Принципы и виды отладки программного средства.
27. Заповеди отладки программного средства.
28. Функциональность и надежность как обязательные критерии качества программного средства.
29. Обеспечение завершенности программного средства.
30. Обеспечение точности программного средства.
31. Общая характеристика процесса обеспечения качества программного средства.
32. Обеспечение легкости применения программного средства.
33. Документация, создаваемая и используемая в процессе разработки программных средств.

34. Пользовательская документация программных средств.
35. Назначение и процессы управления разработкой программного средства.
36. Структура управления разработкой программных средств.
37. Особенности объектного подхода на этапе конструирования программного средства.
38. Особенности объектного подхода на этапе кодирования программного средства.
39. Инструменты разработки программных средств.
40. Инструментальные среды разработки и сопровождения программных средств и принципы их классификации.
41. Основные классы инструментальных сред разработки и сопровождения программных средств.
42. Водопадный подход разработки ПС. Каскадная модель ЖЦ ПС
43. Исследовательское программирование. Инкрементная модель ЖЦ ПС
44. Языки программирования высокого и низкого уровня
45. Надежность программного средства.
46. Технология программирования как технология разработки надежных программных средств.
47. Технология программирования и информатизация общества.
48. Модель перевода.
49. Основные пути борьбы с ошибками.
50. Методы борьбы со сложностью.
51. Обеспечение точности перевода.
52. Преодоление барьера между пользователем и разработчиком.
53. Контроль принимаемых решений.
54. Функциональная спецификация программного средства.
55. Методы контроля внешнего описания программного средства.
56. Денотационная семантика.
57. Аксиоматическая семантика.
58. Языки спецификаций.
59. Архитектурные функции.
60. Контроль архитектуры программных средств.
61. Методы разработки структуры программы.
62. Контроль структуры программы.
63. Пошаговая детализация и понятие о псевдокоде.
64. Контроль программного модуля.
65. Завершимость выполнения программы.
66. Пример доказательства свойства программы.
67. Автономная отладка программного средства.
68. Комплексная отладка программного средства.
69. Обеспечение автономности программного средства.
70. Обеспечение устойчивости программного средства.
71. Обеспечение защищенности программных средств.
72. Обеспечение эффективности программного средства.
73. Обеспечение сопровождаемости программного средства.
74. Обеспечение мобильности.
75. Документация по сопровождению программных средств.
76. Планирование и составление расписаний по разработке ПС.
77. Аттестации программного средства.
78. Объекты и отношения в программировании. Сущность объектного подхода к разработке программных средств.
79. Особенности объектного подхода к разработке внешнего описания программного средства.
80. Инструментальные среды программирования.

81. Понятие компьютерной технологии разработки программных средств и ее рабочие места.
82. Инструментальные системы технологии программирования.
83. Прототипирование
84. Основное назначение моделей ЖЦ ПС
85. Структура стандарта ГОСТ ISO/IEC 12207

### **1. Критерии оценивания компетенций**

Оценка «отлично» выставляется студенту, если он:

Имеет высокие начальные сведения о стандартах проектирования, разработки, тестирования и эксплуатации программного обеспечения;

На высоком уровне выбирает технологию программирования и инструментальные программные средства высокого уровня

На высоком уровне использует основные алгоритмические структуры; знает основные способы записи алгоритмов и конструирования программ с использованием различных алгоритмических языков;

На высоком уровне использует стандартные и собственные структуры данных, базовые и собственные алгоритмы их обработки; использует современные методы и средства разработки алгоритмов при решении задач профессиональной деятельности

На высоком уровне использует современные методы и средства разработки программного обеспечения систем управления

Оценка «хорошо» выставляется студенту, если он:

Получил начальные сведения о стандартах проектирования, разработки, тестирования и эксплуатации программного обеспечения;

Выбирает технологию программирования и инструментальные программные средства высокого уровня

Использует основные алгоритмические структуры; знает основные способы записи алгоритмов и конструирования программ с использованием различных алгоритмических языков;

Использует стандартные и собственные структуры данных, базовые и собственные алгоритмы их обработки; использует современные методы и средства разработки алгоритмов при решении задач профессиональной деятельности

Использует современные методы и средства разработки программного обеспечения систем управления

Оценка «удовлетворительно» выставляется студенту, если он:

Получил слабые начальные сведения о стандартах проектирования, разработки, тестирования и эксплуатации программного обеспечения;

На низком уровне выбирает технологию программирования и инструментальные программные средства высокого уровня

Слабо использует основные алгоритмические структуры; знает основные способы записи алгоритмов и конструирования программ с использованием различных алгоритмических языков;

Слабо использует стандартные и собственные структуры данных, базовые и собственные алгоритмы их обработки; использует современные методы и средства разработки алгоритмов при решении задач профессиональной деятельности

Слабо использует современные методы и средства разработки программного обеспечения систем управления

Оценка «неудовлетворительно» выставляется студенту, если он:

Недостаточно получил начальных сведений о стандартах проектирования, разработки, тестирования и эксплуатации программного обеспечения;

На недостаточном уровне выбирает технологию программирования и инструментальные программные средства высокого уровня

Недостаточно использует основные алгоритмические структуры; знает основные способы записи алгоритмов и конструирования программ с использованием различных алгоритмических языков;

Недостаточно использует стандартные и собственные структуры данных, базовые и собственные алгоритмы их обработки; использует современные методы и средства разработки алгоритмов при решении задач профессиональной деятельности

Недостаточно использует современные методы и средства разработки программного обеспечения систем управления

## **2. Описание шкалы оценивания**

Промежуточная аттестация в форме экзамена предусматривает проведение обязательной экзаменационной процедуры и оценивается в соответствии с критериями оценивания компетенций

## **3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Процедура проведения экзамена осуществляется в соответствии с Положением о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры - в СКФУ.

В экзаменационный билет включаются 2 вопроса

Для подготовки по билету отводится 30 минут.