

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Ефанов Алексей Валерьевич

Должность: Директор Невиномысского технологического института (филиал) СКФУ

Дата подписания: 19.06.2025 13:58:20

Уникальный программный ключ:

49214306dd433e7a1b0f8632f645f9d57c89e3d8

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

**УТВЕРЖДАЮ**

Директор НТИ (филиал) СКФУ  
канд. техн. наук, доцент Ефанов А.В.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

Информационные технологии и программирование

Направление подготовки/специальность	09.03.02	Информационные системы и технологии
Направленность (профиль)/специализация		Информационные системы управления технологическими и сервисными процессами
Год начала обучения		2026
Форма обучения	очная	заочная      очно-заочная
Реализуется в семестре		2 3

## Предисловие

1. Назначение: данный фонд оценочных средств предназначен для оценивания уровня сформированности компетенций при проведении текущего контроля успеваемости и промежуточной аттестации обучающихся по дисциплине «Информационные технологии и программирование».
2. ФОС является приложением к программе дисциплины «Информационные технологии и программирование».
3. Разработчик: Кочеров Юрий Николаевич, доцент кафедры информационных систем, электропривода и автоматики, канд. техн. наук, доцент
4. Проведена экспертиза ФОС.

Члены экспертной группы:

Председатель: Кочеров Ю.Н., кандидат технических наук, доцент, доцент кафедры информационных систем, электропривода и автоматики

Члены комиссии:

Колдаев А.И., заведующий кафедрой информационных систем, электропривода и автоматики, кандидат технических наук, доцент

Евдокимов А.А., кандидат технических наук, доцент, доцент кафедры информационных систем, электропривода и автоматики

Представитель организации-работодателя:

Остапенко Н.А., кандидат технических наук, ведущий инженер-конструктор ООО «Корпоративный институт электротехнического приборостроения «Энергомера» филиала АО «Электротехнические заводы «Энергомера»

Экспертное заключение: ФОС рекомендуется для оценивания уровня сформированности компетенций при проведении текущего контроля успеваемости и промежуточной аттестации студентов по дисциплине «Информационные технологии и программирование».

5. Срок действия ФОС определяется сроком реализации образовательной программы.

## Описание критериев оценивания компетенции на различных этапах их формирования, описание шкал оценивания

Компетенция (ии), индикатор (ы)	Уровни сформированности компетенци(ий),			
	Минимальный уровень не достигнут (Неудовлетворит ельно) 2 балла	Минимальный уровень (удовлетворительно ) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий				
<p>Результаты обучения по дисциплине (модулю): Индикатор: ИД-1 ОПК-6 анализирует задачу и проектирует алгоритм или структуру базы данных (в зависимости от дисциплины), необходимые для создания программного продукта. ИД-2 ОПК-6 Реализует разработанный проект в виде исходного кода на языке программирования, используя современные инструменты и среду разработки. ИД-3 ОПК-6 Проводит тестирование и отладку программы, оценивая её готовность к практическому использованию и соответствие исходным требованиям.</p>	<p>Студент не может проанализировать постановку прикладной задачи, не выделяет исходные данные и требуемый результат. Алгоритм решения либо отсутствует, либо разработан неверно, выбранные структуры данных (массивы, списки) не соответствуют условию задачи. Программный код не создан или создан не самостоятельно. Демонстрирует незнание синтаксиса языка C# и базовых управляющих конструкций (ветвление, циклы). Среда разработки не используется или используется не по назначению. Отладка и тестирование программы не</p>	<p>Студент анализирует задачу поверхностно, но в целом верно выделяет исходные данные и результат. Блок-схема или алгоритм решения разработаны с существенными замечаниями (например, не учтены частные случаи, структура выбрана неоптимально, но логика в целом верна). Программный код на языке C# создан, но содержит грубые синтаксические или логические ошибки. Базовые управляющие конструкции (ветвление, циклы) используются не всегда уместно или с ошибками. Инструменты среды разработки используются минимально (только для компиляции). Отладка кода проводилась формально либо с помощью множественных правок «наугад». Тестирование проведено не в</p>	<p>Студент уверенно анализирует прикладную задачу, четко определяет исходные данные и требуемый результат. Разрабатывает корректный алгоритм (блок-схему), осознанно выбирает подходящие структуры данных (массивы, списки, записи) для реализации. Программный код написан грамотно, с корректным использованием управляющих конструкций языка C#. Студент свободно ориентируется в интегрированной среде разработки (Visual Studio / Rider), использует базовые инструменты для написания и редактирования кода. Возможны незначительные стилистические недочеты или недочеты в оптимизации. Выполняет отладку программы с</p>	<p>Демонстрирует глубокий анализ задачи, способен предложить несколько вариантов алгоритмов и структур данных, аргументированно выбирая оптимальный (с точки зрения производительности, читаемости, использования памяти). Блок-схема и алгоритм полностью соответствуют задаче и не содержат ошибок. Программный код написан на высоком профессиональном уровне: используется эффективный синтаксис C#, корректно применяются современные конструкции, соблюдаются принципы</p>

	<p>проводились. Программа неработоспособна или выдает некорректные результаты, не соответствующие исходным требованиям. Студент не может объяснить причины ошибок.</p>	<p>полном объеме (например, только на «идеальных» входных данных), программа работает корректно только в частных случаях, часть требований не выполнена.</p>	<p>использованием встроенных средств (точки останова, пошаговое выполнение) для поиска ошибок. Проводит функциональное тестирование на основе подготовленного набора тестов. Программа в целом соответствует исходным требованиям, результат корректен, но возможны небольшие недочеты в обработке граничных случаев.</p>	<p>читаемости кода. Студент уверенно использует расширенные возможности среды разработки (рефакторинг, шаблоны кода, сниппеты) для ускорения и улучшения качества разработки. В совершенстве владеет инструментами отладки, эффективно выявляя и исправляя скрытые логические ошибки. Тестирование проведено исчерпывающе, включает проверку граничных условий и некорректных входных данных. Программа полностью готова к практическому применению, полностью соответствует исходным требованиям, результат стабильно корректен.</p>
--	--	--	---	---

Оценивание уровня сформированности компетенции по дисциплине осуществляется на основе «Положения о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры - в

федеральном государственном автономном образовательном учреждении высшего образования «северо-кавказский федеральный университет» в актуальной редакции.

## ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
		<b>Форма обучения очная Семестр 1 Форма обучения заочная Семестр 2</b>	
1.	3.      .	Логическая операция "ИЛИ" на языке C# обозначается 1.    //; 2.    \\; 3.      .	ОПК-14
2.	4.    указания мелких деталей процесса обработки данных.	Языки низкого уровня требуют ... 1.    указания средних деталей процесса обработки данных; 2.    указания крупных деталей процесса обработки данных; 3.    описания алгоритмов; 4.    указания мелких деталей процесса обработки данных.	ОПК-14
3.	2.    количество выполняемых операций в худшем случае.	Чтобы оценить быстродействие программы, можно посчитать 1.    объем памяти приложения; 2.    количество выполняемых операций в худшем случае.	ОПК-14
4.	транслятор	Программа или техническое средство, выполняющее трансляцию программы называется _____ (все символы ответа в нижнем регистре)	ОПК-14
5.	компилятор	Программа, переводящая написанный на языке программирования текст в набор машинных кодов называется _____ (все символы ответа в нижнем регистре)	ОПК-14
6.		Как выводится информация в консоль на языке C#?	ОПК-14
7.		Графическое изображается блока условия в блок схемах алгоритмов?	ОПК-14
8.		Обозначение логических элементов И, ИЛИ, НЕ, ХОР на языке C#	ОПК-14
9.		Обозначение условного оператора на языке C#?	ОПК-14
10.		Цикл с предусловием и постусловием на языке C#	ОПК-14
11.		Итерационный цикл на языке C#?	ОПК-14
12.		Операторы управления циклами в C#?	ОПК-14
13.		Вывод на консоль массива и его элементов?	ОПК-14
14.		Что такое «сортировка»?	ОПК-14
15.		В чем изморятся сложность алгоритма сортировки?	ОПК-14
16.		Метод split на языке C#	ОПК-14

17.		Какого типа переменную лучше использовать для хранения одного символа на языке C#?	ОПК-14
18.		Вывод строковой переменной на консоль с переходом на новую строку и без?	ОПК-14
19.		Применение Console.Read(); Console.ReadLine(); Console.ReadKey();	ОПК-14
20.		Структура IF(если- то- иначе) и ее программирование. Примеры. На языке C#	ОПК-14
		<b>Форма обучения очная Семестр 2 Форма обучения заочная Семестр 3</b>	
21.	return	Какой оператор возвращает значение из метода языке C# (все символы ответа в нижнем регистре)?	ОПК-14
22.		Этим символом обозначается побитовое ИЛИ на языке C#	ОПК-14
23.	2	Дан фрагмент программы, написанной на языке C#: int a = 10; int b = 4; int c = a % b; Console.Write(c); Console.ReadKey(); Какой символ / группа символов выведется на консоль	ОПК-14
24.	break	Этим ключевым словом прерывается цикл на языке C#	ОПК-14
25.	15AND7=1111AND111=111=7	Для представленного кода программы, написанного на языке C рассчитайте, что выдаст компилятор если пользователь ввел последовательно числа 15 и 7: int a = Convert.ToInt16(Console.ReadLine()); int b = Convert.ToInt16(Console.ReadLine()); Console.WriteLine(a & b); Console.ReadKey();	ОПК-14
26.	15OR7=1111OR111=1111=15	Для представленного кода программы, написанного на языке C# рассчитайте, что выдаст компилятор если пользователь ввел последовательно числа 15 и 7: int a = Convert.ToInt16(Console.ReadLine()); int b = Convert.ToInt16(Console.ReadLine()); Console.WriteLine(a   b); Console.ReadKey();	ОПК-14
27.	15XOR7=1111XOR111=1000=8	Для представленного кода программы, написанного на языке C# рассчитайте, что выдаст компилятор если пользователь ввел последовательно числа 15 и 7: int a = Convert.ToInt16(Console.ReadLine()); int b = Convert.ToInt16(Console.ReadLine());	ОПК-14

		<pre>Console.WriteLine(a ^ b); Console.ReadKey();</pre>	
28.	1*2*3*4*5=120	<p>Для представленного кода программы, написанного на языке C# рассчитаете, что выдаст компилятор если пользователь ввел последовательно число 5</p> <pre>int product = 1; int n = Convert.ToInt16(Console.ReadLine()); for (int i = 1; i &lt;= n; i++)     product = product * i; Console.WriteLine(product); Console.ReadKey();</pre>	ОПК-14
29.	0+1+2+3+4+5+6=21	<p>Для представленного кода программы, написанного на языке C# рассчитаете, что выдаст компилятор если пользователь ввел последовательно число 5</p> <pre>int rez = 0, i = 0; int n = Convert.ToInt16(Console.ReadLine()); while (i &lt;= n) {     i++;     rez += i; } Console.WriteLine(rez); Console.ReadKey();</pre>	ОПК-14
30.	0+3+6+9=18	<p>Для представленного кода программы, написанного на языке C# рассчитаете, что выдаст компилятор если пользователь ввел последовательно число 7</p> <pre>int rez = 0, i = 0; int n = Convert.ToInt16(Console.ReadLine()); do {     i += 3;     rez += i; } while (i &lt;= n); Console.WriteLine(rez); Console.ReadKey();</pre>	ОПК-14

31.		Одномерные массивы. Задание массивам первоначальных значений. На языке C#	ОПК-14
32.		Двумерные массивы. Задание массивам первоначальных значений. На языке C#	ОПК-14
33.		Ввод-вывод матриц.	ОПК-14
34.		Вывод элементов главной диагонали матрицы. На языке C#	ОПК-14
35.		Нахождение максимального (минимального) элемента массива	ОПК-14
36.		Языки программирования высокого и низкого уровня	ОПК-14
37.		Основные математические функции на языке C#	ОПК-14
38.		Строки как последовательность символов на языке C#	ОПК-14
39.		Алгоритм линейной структуры на языке C#	ОПК-14
40.		Генерация случайных чисел на языке C#	ОПК-14

## **2. Описание шкалы оценивания**

В рамках рейтинговой системы успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации. Рейтинговая система оценки знаний студентов основана на использовании совокупности контрольных мероприятий по проверке пройденного материала (контрольных точек), оптимально расположенных на всем временном интервале изучения дисциплины. Принципы рейтинговой системы оценки знаний студентов основываются на положениях, описанных в Положении об организации образовательного процесса на основе рейтинговой системы оценки знаний студентов в ФГАОУ ВО «СКФУ».

*Рейтинговая система оценки не предусмотрено для студентов, обучающихся на образовательных программах уровня высшего образования магистратуры, для обучающихся на образовательных программах уровня высшего образования бакалавриата заочной и очно-заочной формы обучения.*

## **3. Критерии оценивания компетенций\***

Оценка «отлично» выставляется студенту, если он:

демонстрирует глубокий анализ прикладной задачи, предлагает и аргументированно обосновывает выбор оптимального алгоритма и структур данных (массивы, списки, записи) с учётом эффективности и читаемости кода;

создаёт программный код на языке C# высокого профессионального уровня, корректно применяет современные синтаксические конструкции, управляющие операторы (ветвление, циклы) и активно использует расширенные возможности интегрированной среды разработки (рефакторинг, шаблоны, сниппеты) для повышения качества и скорости разработки;

в совершенстве владеет средствами отладки (точки останова, пошаговое выполнение), эффективно выявляет и устраняет логические ошибки;

проводит исчерпывающее функциональное тестирование, включая проверку граничных условий и некорректных входных данных; итоговая программа полностью соответствует исходным требованиям, стабильно работает и готова к практическому применению.

Оценка «хорошо» выставляется студенту, если он:

уверенно анализирует постановку задачи, правильно выделяет исходные данные и требуемый результат, разрабатывает корректный алгоритм (блок-схему) и осознанно выбирает подходящие структуры данных;

пишет грамотный код на C# с использованием всех необходимых управляющих конструкций, свободно ориентируется в среде разработки для написания, редактирования и компиляции программы (допускаются незначительные стилистические недочёты или неоптимальные решения, не влияющие на работоспособность);

применяет встроенные средства отладки (точки останова, пошаговое выполнение) для поиска ошибок;

выполняет функциональное тестирование на основе подготовленного набора тестов; программа в целом соответствует исходным требованиям и выдаёт корректные результаты, однако возможны мелкие недочёты в обработке граничных случаев.

Оценка «удовлетворительно» выставляется студенту, если он:

выполняет поверхностный, но в целом верный анализ задачи, выделяя основные исходные данные и результат; разработанный алгоритм (блок-схема) содержит существенные замечания (например, не учтены частные случаи), но общая логика решения прослеживается;

создаёт программный код на C#, однако допускает грубые синтаксические или логические ошибки, некорректно использует базовые управляющие конструкции; инструменты среды разработки применяет минимально (только для компиляции); проводит отладку формально либо методом «проб и ошибок»; тестирование выполняется не в полном объёме (например, только на «идеальных» данных); программа работает корректно лишь в отдельных случаях и не полностью удовлетворяет исходным требованиям.

Оценка «неудовлетворительно» выставляется студенту, если он:

не может проанализировать постановку задачи, выделить исходные данные и требуемый результат; алгоритм решения отсутствует или разработан неверно, выбранные структуры данных не соответствуют условию;

не создал программный код либо представил несамостоятельно выполненную работу; демонстрирует незнание синтаксиса C# и базовых управляющих конструкций;

не проводил отладку и тестирование; программа неработоспособна или выдаёт результаты, не соответствующие условиям задачи, студент не может объяснить причины ошибок.