Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение   
 высшего образования «Новосибирский национальный исследовательский  
 государственный университет» (Новосибирский государственный университет, НГУ)

**Факультет информационных технологий**

СОГЛАСОВАНО

Декан ФИТ НГУ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.М. Лаврентьев

«03» июля 2019г.

**Фонд оценочных средств промежуточной аттестации**

**по дисциплине Программирование**

Направление подготовки: 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Направленность (профиль): Программная инженерия и компьютерные науки

Квалификация: бакалавр

Форма обучения: очная Год обучения: 1, семестр 1, 2

|  |  |
| --- | --- |
| Форма аттестации | Семестр |
| Дифференцированный зачет | 1 |
| Экзамен | 2 |

Новосибирск 2019

**Фонд оценочных средств** промежуточной аттестации по дисциплине является **Приложением 1** к рабочей программе дисциплины «Программирование», реализуемой в рамках образовательной программы высшего образования – программы бакалавриата 09.03.01 Информатика и вычислительная техника, направленность (профиль): Программная инженерия и компьютерные науки.

Фонд оценочных средств промежуточной аттестации по дисциплине утвержден решением ученого совета факультета информационных технологий, протокол № 75 от 02.07.2019.

Разработчики:

Старший преподаватель кафедры общей информатики ФИТ,

кандидат физико-математических наук Е.С.Петров

Заведующий кафедрой общей информатики ФИТ,

доктор физико-математических наук Д.Е. Пальчунов

Ответственный за образовательную программу:

доцент кафедры систем информатики ФИТ,

кандидат технических наук А.А. Романенко

1. **Содержание и порядок проведения промежуточной аттестации  
   по дисциплине**
   1. **Общая характеристика содержания промежуточной аттестации**

Промежуточная аттестация по дисциплине «Программирование» проводится по завершению периода освоения образовательной программы (семестра) для оценки сформированности компетенций в части следующих индикаторов достижения компетенции (таблица П1.1).

Таблица П1.1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Код | Компетенции, формируемые в рамках дисциплины  «Программирование» | Семестр 1 | | Семестр 2 | | |
| Портфолио | Диф. зачет | Портфолио | Экзамен |
|  | **ОПК-1 Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности:** | | | | | |
| **ОПК-1.1** | Знать: основы математики, физики, вычислительной техники и программирования | **+** | **+** | **+** | **+** |
| **ОПК-1.2** | Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и обще-инженерных знаний, методов математического анализа и моделирования | **+** | **+** | **+** | **+** |
| **ОПК-1.3** | Владеть: навыками теоретического и экспериментального исследования объектов профессиональной деятельности | **+** | **+** | **+** | **+** |
|  | **ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения:** | | | | | |
| **ОПК-8.1** | Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения | **+** | **+** | **+** | **+** |
| **ОПК-8.2** | Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули | **+** | **+** | **+** | **+** |
| **ОПК-8.3** | Владеть: языком программирования; навыками отладки и тестирования работоспособности программы | **+** | **+** | **+** | **+** |
|  | **ОПК-9 Способен осваивать методики использования программных средств для решения практических задач** | | | | | |
| **ОПК-9.1** | Знать: классификацию программных средств и возможности их применения для решения практических задач | **+** | **+** | **+** | **+** |
| **ОПК-9.2** | Уметь: находить и анализировать техническую документацию по использованию программного средства, выбирать и использовать необходимые функции программных средств для решения конкретной задачи | **+** | **+** | **+** | **+** |
| **ОПК-9.3** | Владеть: способами описания методики использования программного средства для решения конкретной задачи в виде документа, презентации или видеоролика | **+** | **+** | **+** | **+** |

Промежуточная аттестация включает 2 этапа. Все компетенций оценивается как на основе портфолио, в которое входят работы, выполненные в рамках дисциплины, так и дифференцированным зачётом или экзаменом.

Экзаменационные задания проверяют знания и умения в рамках тем, перечисленных в программе учебной дисциплины.

* 1. **Порядок проведения промежуточной аттестации по дисциплине**

Промежуточная аттестация по дисциплине «Программирование» проводится по завершению каждого периода ее освоения (семестра). Результаты промежуточной аттестации по дисциплине оцениваются по шкале «неудовлетворительно», «удовлетворительно», «хорошо», «отлично».

Оценка выставляется на основании взвешенной суммы баллов ВСБ, полученных в течение семестра. При ВСБ в интервале [0, 2.5) выставляется оценка «неудовлетворительно»; в интервале [2.5, 3.5) – «удовлетворительно»; [3.5, 4.5) – «хорошо»; [4.5, ∞) – «отлично».

Оценки «удовлетворительно», «хорошо», «отлично» означают успешное прохождение промежуточной аттестации и соответствуют соответственно пороговому, базовому и продвинутому уровню сформированности компетенций на момент аттестации.

В 1 семестре оценка за освоение дисциплины выставляется на основе портфолио работ студента, которое включает:

1. 3 контрольных работы (1-я и 2-я контрольная неделя в каждой группе, конец семестра потоковая или во время дифференцированного зачета);
2. 7 лабораторных работ по темам, изучаемым в 1 семестре.

Оценка за освоение дисциплины в 1 семестре выставляется на основании взвешенной суммы баллов ВСБ = 0,6 \* БЛР + 0,4 \* БКР, где БКР и БЛР – это соответственно средний балл по пятибалльной шкале за контрольные и лабораторные работы, входящие в портфолио студента.

Дифференцированный зачет заключается в выполнении одной или нескольких контрольных работ с целью замещения эквивалентных контрольных работ в портфолио и повышения БКР.

В 2 семестре промежуточная аттестация по дисциплине включает 2 этапа:

1. подготовку портфолио, которое включает:
   1. 3 контрольных работы (1-я и 2-я контрольная неделя в каждой группе, конец семестра потоковая);
   2. 6 лабораторных работ по темам, изучаемым в 2 семестре;
2. выполнение экзаменационных заданий.

Оценка за освоение дисциплины в 2 семестре выставляется на основании взвешенной суммы баллов ВСБ = 0,24 \* БКР + 0,36 \* БЛР + 0,4 \* БЭЗ, где БКР, БЛР – это соответственно средний балл по пятибалльной шкале за контрольные и лабораторные работы, входящие в портфолио студента, а БЭЗ – это балл, полученный за выполнение экзаменационного задания.

БЭЗ – это второй в порядке возрастания балл за выполнение частей экзаменационного задания (билета), включающего три темы для собеседования и две задачи для письменного решения. Каждая часть экзаменационного задания оценивается по пятибалльной шкале.

Контрольная работа состоит из 1-20 заданий. Выполнение каждого задания оценивается баллами от 0 (не выполнено) до полной стоимости задания (выполнено). Контрольная работа оценивается на основе суммы баллов за выполнение заданий по следующим правилам:

1. контрольная работа отсутствует в портфолио – БКР = 0;
2. менее 50% суммарной стоимости всех заданий – БКР = 2;
3. от 50% до 75% суммарной стоимости всех заданий -- БКР = 3;
4. от 75% до 87,5% суммарной стоимости всех заданий -- БКР = 4;
5. от 87,5% суммарной стоимости всех заданий -- БКР = 5.

Лабораторная работа оценивается по пятибалльной шкале на основе результатов автоматического тестирования и собеседования с преподавателем по теме лабораторной работы по следующим правилам:

1. код лабораторной работы не опубликован на gitlab.ccfit.nsu.ru – БЛР = 0;
2. ошибки или предупреждения во время компиляции или сборки – БЛР <= 1;
3. проходят менее 50% автоматических тестов – БЛР <= 2;
4. проходят менее 100% автоматических тестов – БЛР <= 3;
5. на сайте есть неисправленные замечания преподавателя по оформлению кода – БЛР <= 4;
6. иначе – БЛР = 5.
7. **Требования к структуре и содержанию фонда оценочных средств  
   промежуточной аттестации по дисциплине**

Перечень оценочных средств, применяемых на каждом этапе проведения промежуточной аттестации по дисциплине, представлен в таблице П1.2.

Таблица П1.2

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование оценочного средства | Краткая характеристика оценочного средства | Представление оценочного средства в фонде |
| Семестр 1 | | | |
| 1 | Портфолио | Целевая подборка работ студента, раскрывающая его индивидуальные образовательные достижения в дисциплине | 3 контрольных работы,  7 лабораторных работ по темам, изучаемым в 1 семестре |
| 2 | Диф. зачет | Комплекс заданий по темам 1 семестра для письменного выполнения | Не более 3 контрольных работ для замены контрольных работ в портфолио по желанию студента |
| Семестр 2 | | | |
| 1 | Портфолио | Целевая подборка работ студента, раскрывающая его индивидуальные образовательные достижения в дисциплине | 3 контрольных работы,  6 лабораторных работ по темам, изучаемым в 1 семестре |
| 2 | Экзаменационный билет | Комплекс вопросов и заданий по темам 1 и 2 семестра | Список теоретических вопросов и практических заданий |

* 1. **Требования к структуре и содержанию оценочных средств  
     аттестации**

2.1.1 Требования к структуре и содержанию портфолио 1 семестра

Портфолио состоит из следующих элементов:

1. контрольных работы по языку программирования Си;
2. 7 лабораторных работ по темам, изучаемым в 1 семестре и перечисленным в программе дисциплины.

2.1.2 Требования к структуре и содержанию портфолио 2 семестра

Портфолио состоит из следующих элементов:

1. контрольных работы по языку программирования Си;
2. 6 лабораторных работ по темам, изучаемым в 2 семестре и перечисленным в программе дисциплины.

2.1.3 Форма и перечень вопросов экзаменационного билета

**Форма экзаменационного билета**

Таблица П1.3

|  |  |  |  |
| --- | --- | --- | --- |
|  | Новосибирский государственный университет  **Экзамен Семестр 2** | |  |
|  | Программирование | |  |
|  | наименование дисциплины  09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА  Программная инженерия и компьютерные науки | |  |
|  | наименование образовательной программы    **ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №**   1. Вопрос из категории «Алгоритмы и структуры данных» 2. Вопрос из категории «Язык Си» 3. Вопрос из категории «Язык Си» 4. Задача на написание кода на языке Си 5. Задача на составление алгоритма | |  |
|  | Составитель |  |  |
|  |  | Е.С. Петров |  |
|  | (подпись) |  |  |
|  | Ответственный за образовательную программу | |  |
|  |  | А.А. Романенко |  |
|  | (подпись) |  |  |
|  | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20     г. |  |  |

Перечень вопросов экзамена, структурированный по категориям, представлен в таблице П1.4

Таблица П1.4

|  |  |
| --- | --- |
| Категория | Формулировка вопроса |
| Алгоритмы и структуры данных (ОПК-8.1 – ОПК-8.3; ОПК-1.1 – ОПК-1.3) | 1. Алгоритм Беллмана-Форда построения кратчайшего пути в графе. Оценка сложности. 2. Алгоритм Бойера-Мура поиска подстроки в строке без суффиксной эвристики. Оценка сложности. 3. Алгоритм Рабина-Карпа поиска подстроки в строке. Оценка сложности. 4. Алгоритм топологической сортировки. Оценка сложности. 5. Алгоритмы записи целых чисел в произвольной с.с. 6. Алгоритмы обхода графа в глубину и в ширину. Оценки сложности. 7. Алгоритмы обхода деревьев (в ширину, в глубину: в пре/пост/инфиксном порядке). 8. Алгоритмы перевода записи вещественных чисел из с.с. в с.с. 9. Алгоритмы перевода записи целых чисел из с.с. в с.с. 10. Алгоритмы с возвратом на примере расстановки ферзей. 11. Алгоритмы сортировки массива простым включением, простым выбором, «пузырьком». Оценка сложности. 12. Алгоритм Краскала построения каркаса графа (без СНМ). Оценка сложности. 13. Алгоритм Прима каркаса графа. Оценка сложности. 14. Алгоритм быстрой сортировки Хоара. Оценка сложности. 15. Алгоритм обхода графа в ширину с использованием очереди. Оценка сложности. 16. Алгоритм пирамидальной сортировки. Оценка сложности. 17. Алгоритм Флойда-Уоршелла вычисления длин кратчайших путей между всеми парами вершин. Оценка сложности. 18. Алгоритм Дейкстры поиска кратчайших путей в графе. Оценка сложности. 19. Алгоритм вставки элемента в АВЛ-дерево. Длинный и короткий повороты. Оценка сложности. 20. Метод динамического программирования на примере конкретной задачи. Обратный ход. Оценка сложности. 21. Метод Хаффмана построения оптимального префиксного двоичного кода. 22. Методы поиска в массиве: линейный поиск, бинарный поиск, хэш-таблицы. Оценка сложности. 23. Алгоритм построения постфиксной записи арифметического выражения. Алгоритм вычисления выражения по постфиксной записи. 24. Граф. Представление в ЭВМ: матрица смежности, матрица инцидентности, список дуг, упакованный список дуг. 25. Дерево. Дерево двоичного поиска. Способы представления деревьев в ЭВМ: скобочная запись, список предков, указатели. 26. Стек и очередь как абстрактные типы данных. Способы реализации на языке Си. Реализация очереди с помощью циклического буфера. 27. Список как абстрактный тип данных. Одно- и двусвязные, циклические списки. Способы реализации на языке Си. 28. Система непересекающихся множеств как абстрактный тип данных. Реализация на языке Си через дерево с рангом и сжатием путей. |
| Язык Си (ОПК-8.1 – ОПК-8.3) | 1. Язык Си. Вещественные типы. Операции над вещественными числами. Представление в памяти, распределение на числовой прямой. Преобразования между вещественными типами языка Си. 2. Язык Си. Выражения. Классы операций по числу операндов, размещению относительно операндов. Приоритет, ассоциативность. Связь приоритета и класса операции. 3. Язык Си. Динамически распределяемая память. Стандартные функции для работы с динамической памятью. 4. Язык Си. Понятие времени жизни, области видимости и связывания переменных. Глобальные и локальные переменные. Модификаторы области видимости, времени жизни и связывания. 5. Язык Си. Понятие указателя. Операции над указателями. 6. Язык Си. Понятие функции. Граф вызовов, стек вызовов. Описание функций. Передаваемые параметры, возвращаемое значение. 7. Язык Си. Препроцессор. Понятие макроподстановки. Синтаксис директив define и undef. 8. Язык Си. Препроцессор. Понятие словной компиляции. Синтаксис директив include, if/else/endif. 9. Язык Си. Массивы. Многомерные массивы. Инициализаторы массивов. Расположение элементов в памяти. Связь с указателями. 10. Язык Си. Строки. Длина строки. Инициализаторы строк. Функции работы со строками. Связь с массивами. 11. Язык Си. Функции с переменным числом параметров. Доступ к значениям фактических параметров. Особенности использования. 12. Язык Си. Структуры. Описание структур. Операции над структурами. Инициализатор структур. Расположение элементов в памяти. 13. Язык Си. Целочисленные типы. Операции над целыми числами. Представление в памяти. Преобразования между целочисленными типами. |

Перечень задач, представлен в таблице П1.4.1

Таблица П1.4.1

|  |  |
| --- | --- |
| Категория | Формулировка задания |
| Алгоритмы и структуры данных (ОПК-1.1 - ОПК-1.3) | 1. Односвязный список задан указателем на первый элемент. Написать функцию, меняющую порядок элементов на обратный за счёт изменения связей элементов. 2. Простой способ шифровки текста, состоящего из строчных латинских букв и знаков препинания, состоит в замене каждой буквы на букву с заданным циклическим сдвигом n (если сдвиг 1, то "a" заменяется на "b", "b" на "c", "z" на "a"; если сдвиг 2, то "a" заменяется на "c", "y" на "a", "z" на "b"). Написать функцию зашифровки и расшифровки текста. Исходный текст берется из файла, результаты помещаются в другой файл. 3. Многочлен от двух переменных задается в виде матрицы вещественных коэффициентов, в позиции (i, j) стоит коэффициент при x^i \* y^j. Написать функцию, умножающую многочлены, заданные таким образом. 4. Дана непустая последовательность слов, составленных из строчных латинских букв; между соседними словами пробел, в конце -- точка. Написать функцию, которая печатает в алфавитном порядке все согласные буквы, которые входят только в одно слово. 5. Написать функцию для перевода числа из одной позиционной системы счисления в другую. Цифры, большие 9, обозначить латинскими буквами А, В, С, ... Основания систем счисления не больше 36. 6. Простое число называется числом Мерсенна, если оно может быть представлено в виде 2^р-1, где р -- простое число. Написать функцию, печатающую все числа Мерсенна, меньшие данного n. 7. Время суток представлено в виде структуры, содержащей информацию о часе, минутах и секундах. Написать функцию, которая увеличивает значение времени на n секунд (после 23:59:59 идет 00:00:00) 8. Дана последовательность ненулевых целых чисел a[0], a[1], a[2], ... длины не менее 3. Триплетом называются три последовательных элемента последовательности. Знаки чисел в триплете могут образовывать 8 комбинаций (+++, ++-, +-+, +--, -++, -+-, --+, ---). Написать функцию, записывающую в массив из 8 элементов число триплетов, имеющих каждую из перечисленных комбинаций знаков. 9. Написать функцию для печати распределения слов во входном файле по их длине (сколько слов из одной буквы, сколько из двух и т.д.). Файл содержит строчные латинские буквы и пробелы. 10. Ориентированный граф задан матрицей смежности. Написать функцию, находящую все вершины, в которые есть путь из данной. 11. Задано двоичное дерево, элементами которого являются целые числа. Написать функцию для нахождения наибольшего элемента дерева. 12. Написать функцию, подсчитывающая в файле целых чисел количество элементов равных минимальному за один просмотр файла. 13. Написать функцию, которая объединяет два упорядоченных списка целых чисел в один упорядоченный список. 14. Написать функцию, печатающую названия и количество деталей, которых нет на складе или не хватает для удовлетворения заявки. Описание деталей на складе и деталей в заявке хранятся в двух файлах. Для каждой детали задано название (не более 10 символов), код (целое число), и количество (целое число). Каждый файл упорядочен по возрастанию кода. 15. Гамма-функция Г(x) обладает свойством: Г(x+1) = x\*Г(x). Дана таблица приближенных значений гамма-функции на отрезке от 1.00 до 1.99 с шагом 0.01. Написать функцию, приближенно вычисляющую Г(x) с помощью этой таблицы. 16. Дана матрица A целых чисел. Написать функцию, которая отметит единицами в массиве B все симметричные строки А и нулями все несимметричные строки А. 17. Дана матрица A. Написать функцию, которая построит матрицу B, такую что элемент B[i][j] равен произведению элементов матрицы A, показанных на рисунке.   j  \_\_\_\_\_\_\_\_\_\_\_\_\_\_  | |  | |  i | \*\*\*\*\*\*\*\*\*\*|  | \*\*\*\*\*\*\*\*\*\*|  | \*\*\*\*\*\*\*\*\*\*|   1. Анкета студента содержит: фамилию, номер группы и набор слушаемых курсов лекций (не более 10). Написать функцию, считывающую анкеты из одного файла и записывающую в другой файл списки фамилий студентов, слушающих каждый курс лекций, разделяя их пустыми строками. 2. Написать функцию, вычисляющую высоту двоичного дерева. 3. Дана последовательность вещественных чисел. Написать функцию, которая строит по этой последовательности два односвязных списка, первый из которых содержит все неповторяющиеся положительные числа последовательности, а второй -- все неповторяющиеся отрицательные. 4. Анкета студента содержит: фамилию, номер группы и набор слушаемых курсов лекций (не более 10). Написать функцию, считывающую анкеты из одного файла и записывающую в другой файл все пары студентов, которые не слушают одинаковые лекции. 5. Написать функцию, проверяющую, является ли строка арифметическим выражением, составленным из однобуквенных идентификаторов и двуместных операций + и -. 6. Арифметическое выражение представлено двоичным деревом, в узлах которого находятся знаки операций, а в листьях – аргументы операций. Написать функцию, вычисляющую по заданному дереву значение выражения. 7. Два натуральных числа представлены в k-ричной системе счисления (1 < k <= 10) как последовательности цифр из интервала [0, k-1]. Написать функцию, которая построит аналогичное представление для разности данных чисел. 8. В каждой вершине бинарного дерева хранится вещественное число. Написать функцию, которая находит разницу между максимальным и минимальным числом в дереве. 9. Дана строка, содержащая запись выражения, принадлежащего грамматике, заданной следующими БНФ:   <выражение> ::= <цифра> | (<выражение> <знак> <цифра>)  <знак> :: = + | – | \*  <цифра> :: = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9  Написать функцию, которая по записи выражения в текстовом файле вычислит его значение.   1. Граф задан списком дуг в файле. Написать функцию, подсчитывающую число его вершин. 2. Граф задан списком дуг в файле. Написать функцию, которая запишет в другой файл представление этого графа списками смежности. 3. Написать функцию, выполняющую подсчет числа листьев произвольного бинарного дерева. 4. Дано произвольное целое число K и массив. Написать функцию, перемещающую каждый элемент на K позиций вперед, считая, что за последним элементом массива снова идет 0-й. Каждый элемент разрешается перемещать только один раз. Использовать дополнительный массив нельзя. 5. Написать функцию, которая сосчитает число пар рядом стоящих единиц в двоичной записи целого числа без знака. 6. Написать функцию, вычисляющую квадратный корень из данного числа, используя только операции +, /, \*. 7. Дан массив целых чисел без знака. Написать функцию, которая найдет в массиве число, в двоичной записи которого больше всего единиц. 8. Дано дерево, вершины которого имеют произвольное число потомков и хранят символы. Написать функцию, которая печатает все элементы дерева по уровням, разделяя уровни переводом строки: сначала -- из корня дерева, затем -- из вершин, дочерних по отношению к корню, затем -- из вершин, дочерних по отношению к этим вершинам, и т. д. 9. Дано дерево, вершины которого имеют произвольное число потомков и хранят целые числа. Написать функцию, которая находит в дереве длину кратчайшего пути от корня до вершины с данным значением. 10. Написать функцию, которая подсчитывает число вершин на данном уровне дерева, вершины которого имеют произвольное число потомков. 11. Написать функцию, которая изменит порядок слов в строке на обратный. Слова разделяются пробелами. |

Набор экзаменационных билетов формируется и утверждается в установленном порядке в начале учебного года при наличии контингента обучающихся, завершающих освоение дисциплины «Программирование» в текущем учебном году.

1. **Критерии оценки сформированности компетенций в рамках промежуточной аттестации по дисциплине**

Таблица П1.5

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Шифр компетенций** | **Структурные элементы оценочных средств** | **Показатель сформированности** | **Не сформирован** | **Пороговый уровень** | **Базовый уровень** | **Продвинутый уровень** |
| ОПК-1 | Портфолио  Вопросы и задачи экзаменационного билета[[1]](#footnote-1) | ОПК-1.1 Знать: основы математики, физики, вычислительной техники и программирования  ОПК-1.2 Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и обще-инженерных знаний, методов математического анализа и моделирования  ОПК-1.3 Владеть: навыками теоретического и экспериментального исследования объектов профессиональной деятельности | Испытывает трудности с записью простых алгоритмов (просмотр массива, обмен значений переменных и т.п.) | Уверенно записывает алгоритмы, включенные в программу дисциплины | Дополнительно к пороговому уровню: верно оценивает время работы и расход памяти алгоритмов на заданном множестве входных данных | Дополнительно к базовому уровню: верно указывает способ устранения узких мест алгоритма в зависимости от множества входных данных |
| ОПК-8 | Портфолио  Вопросы и задачи экзаменационного билета[[2]](#footnote-2) | ОПК-8.1 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения  ОПК-8.2 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули  ОПК-8.3 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы | Неверно объясняет на примерах смысл конструкций языка Си | Верно объясняет в общем виде смысл конструкций языка Си | Дополнительно к пороговому уровню: знает количественные характеристики конструкций языка Си (точность, время работы, расход памяти и т.п.) | Дополнительно к базовому уровню: знает и применяет методики создания качественных программ |
| ОПК-9 | Портфолио  Вопросы и задачи экзаменационного билета2 | ОПК-9.1 Знать: классификацию программных средств и возможности их применения для решения практических задач  ОПК-9.2 Уметь: находить и анализировать техническую документацию по использованию программного средства, выбирать и использовать необходимые функции программных средств для решения конкретной задачи  ОПК-9.3 Владеть: способами описания методики использования программного средства для решения конкретной задачи в виде документа, презентации или видеоролика | Код лабораторной работы отсутствует на сайте github.com, либо ошибки или предупреждения во время компиляции или сборки, либо проходит менее 50% автоматических тестов | Код лабораторной работы есть на сайте github.com, собирается без предупреждений компилятора и линкера, проходит от 50% до 99% тестов | Дополнительно к пороговому уровню: проходит 100% автоматических тестов, но на сайте github.com есть неисправленные замечания преподавателя по оформлению кода | Дополнительно к базовому уровню: исправлены все замечания преподавателя по оформлению кода на сайте github.com |

1. **Критерии выставления оценок по результатам промежуточной аттестации по дисциплине**

Промежуточная аттестация по дисциплине «Программирование» проводится по завершению каждого периода ее освоения (семестра). Результаты промежуточной аттестации по дисциплине оцениваются по шкале «неудовлетворительно», «удовлетворительно», «хорошо», «отлично».

Оценка выставляется на основании взвешенной суммы баллов ВСБ, полученных в течение семестра. При ВСБ в интервале [0, 2.5) выставляется оценка «неудовлетворительно»; в интервале [2.5, 3.5) – «удовлетворительно»; [3.5, 4.5) – «хорошо»; [4.5, ∞) – «отлично».

Оценки «удовлетворительно», «хорошо», «отлично» означают успешное прохождение промежуточной аттестации и соответствуют соответственно пороговому, базовому и продвинутому уровню сформированности компетенций на момент аттестации.

В 1 семестре оценка за освоение дисциплины выставляется на основе портфолио работ студента, которое включает:

1. 3 контрольных работы (1-я и 2-я контрольная неделя в каждой группе, конец семестра потоковая или во время дифференцированного зачета);
2. 7 лабораторных работ по темам, изучаемым в 1 семестре.

Оценка за освоение дисциплины в 1 семестре выставляется на основании взвешенной суммы баллов ВСБ = 0,6 \* БЛР + 0,4 \* БКР, где БКР и БЛР – это соответственно средний балл по пятибалльной шкале за контрольные и лабораторные работы, входящие в портфолио студента.

Дифференцированный зачет заключается в выполнении одной или нескольких контрольных работ с целью замещения эквивалентных контрольных работ в портфолио и повышения БКР.

В 2 семестре промежуточная аттестация по дисциплине включает 2 этапа:

1. подготовку портфолио, которое включает:
   1. 3 контрольных работы (1-я и 2-я контрольная неделя в каждой группе, конец семестра потоковая);
   2. 6 лабораторных работ по темам, изучаемым в 2 семестре;
2. выполнение экзаменационных заданий.

Оценка за освоение дисциплины в 2 семестре выставляется на основании взвешенной суммы баллов ВСБ = 0,24 \* БКР + 0,36 \* БЛР + 0,4 \* БЭЗ, где БКР, БЛР – это соответственно средний балл по пятибалльной шкале за контрольные и лабораторные работы, входящие в портфолио студента, а БЭЗ – это балл, полученный за выполнение экзаменационного задания.

БЭЗ – это второй в порядке возрастания балл за выполнение частей экзаменационного задания (билета), включающего три темы для собеседования и две задачи для письменного решения. Каждая часть экзаменационного задания оценивается по пятибалльной шкале.

Контрольная работа состоит из 1-20 заданий. Выполнение каждого задания оценивается баллами от 0 (не выполнено) до полной стоимости задания (выполнено). Контрольная работа оценивается на основе суммы баллов за выполнение заданий по следующим правилам:

1. контрольная работа отсутствует в портфолио – БКР = 0;
2. менее 50% суммарной стоимости всех заданий – БКР = 2;
3. от 50% до 75% суммарной стоимости всех заданий -- БКР = 3;
4. от 75% до 87,5% суммарной стоимости всех заданий -- БКР = 4;
5. от 87,5% суммарной стоимости всех заданий -- БКР = 5.

Лабораторная работа оценивается по пятибалльной шкале на основе результатов автоматического тестирования и собеседования с преподавателем по теме лабораторной работы по следующим правилам:

1. код лабораторной работы не опубликован на github.com, gitlab.com, bitbucket.org – БЛР = 0;
2. ошибки или предупреждения во время компиляции или сборки – БЛР <= 1;
3. проходят менее 50% автоматических тестов – БЛР <= 2;
4. проходят менее 100% автоматических тестов – БЛР <= 3;
5. на сайте есть неисправленные замечания преподавателя по оформлению кода – БЛР <= 4;
6. иначе – БЛР = 5.

**Лист актуализации фонда оценочных средств промежуточной аттестации**

**по дисциплине  
«Программирование»**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Характеристика внесенных изменений (с указанием пунктов документа) | Дата и №  протокола Ученого совета ФИТ | Подпись  ответственного |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Выбор показателя сформированности компетенции (укрупненной характеристики компетенции) из представленных для оценки осуществляется случайным образом [↑](#footnote-ref-1)
2. Выбор показателя сформированности компетенции (укрупненной характеристики компетенции) из представленных для оценки осуществляется случайным образом [↑](#footnote-ref-2)