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 2019 г.

**Фонд оценочных средств промежуточной аттестации**

**по дисциплине Императивное программирование**

Направление подготовки: 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Направленность (профиль): Компьютерные науки и системотехника

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 «Императивное программирование», реализуемой в рамках образовательной программы высшего образования – программы бакалавриата 09.03.01 Информатика и вычислительная техника, направленность (профиль): Компьютерные науки и системотехника

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

Разработчики:

Доцент

кафедры общей информатики ФИТ, Т.В. Нестеренко

Заведующий кафедрой систем информатики ФИТ,

доктор физико-математических наук М.М. Лаврентьев

Ответственный за образовательную программу:

доцент кафедры систем информатики ФИТ,

кандидат физико-математических наук Д.С. Мигинский

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

Промежуточная аттестация по дисциплине «Императивное программирование» проводится по завершению периодов освоения образовательной программы (семестров) для оценки сформированности компетенций в части следующих индикаторов достижения компетенции (таблица П1.1).

Таблица П1.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Код | Компетенции, формируемые в рамках дисциплины  «Императивное программирование» | Семестр 1 | | Семестр 2 | |
| Портфолио | Дифзачет | Портфолио | Экзамен |
|  | **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 | **+** | **+** | **+** | **+** |

Тематика экзаменационных вопросов соответствует избранным разделам (темам) дисциплины «Императивное программирование»:

1 семестр

* Системы счисления, реализация алгоритмов перевода
* Оценка сложности вычислительных алгоритмов и общие методы решения вычислительных задач
* Перестановки набора, методы поиска всех перестановок
* Задача поиска, методы поиска в массиве и подстроки в строке
* Задача сортировки массивов, простые и улучшенные методы сортировки
* Управление памяти в Си
* Классические модели динамической памяти: список, стэк, очередь; операции, способы реализации, задачи
* Бинарные файлы, cортировка файлов

2 семестр

* Динамическое программирование
* Абстрактные структуры данных: графы, деревья, способы реализации
* Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, реализация
* Элементы теорий вероятностей, информации и кодирования, реализация проекта
* В-деревья, красно-черные деревья, реализация проекта
* Раскраски графа, гамильтоновы и Эйлеровы циклы, алгоритмы и реализация.
* Потоки в сетях, алгоритмы поиска максимального потока, реализация
* Элементы теории языков
  1. **Порядок проведения промежуточной аттестации по дисциплине**

Дифференцированный зачет и экзамен проводятся в аудитории, студентам разрешено пользоваться бумагой для записей и авторучкой. Справочной, учебной и другой литературой пользоваться не разрешается. Использование электронных устройств (телефоны, любые виды компьютеров, т.д.) запрещено. Необходимым условием для прохождения промежуточной аттестации является положительная оценка по результатам всех выполненных и сданных в течение семестра заданий, входящих в рамки портфолио.

1. **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 П1.3.

Таблица П1.3

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование оценочного средства | Краткая характеристика оценочного средства | Представление оценочного средства в фонде |
| Этап 1 - портфолио | | | |
| 1. | Разноуровневые задачи и задания | Различают задачи и задания:  а) репродуктивного уровня, позволяющие оценивать и диагностировать  знание фактического материала (базовые понятия, алгоритмы, факты) и умение правильно использовать специальные термины и понятия, узнавание объектов изучения в рамках определенного раздела дисциплины;  б) реконструктивного уровня, позволяющие оценивать и диагностировать умения синтезировать, анализировать, обобщать фактический и теоретический материал с формулированием конкретных выводов, установлением причинно-следственных связей;  в) творческого уровня, позволяющие оценивать и диагностировать умения, интегрировать знания различных областей, аргументировать собственную точку зрения. | Комплект разноуровневых задач и заданий |
| Этап 2 – дифзачет | | | |
| 2 | Собеседование | Средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п. | Вопросы по темам/разделам дисциплины |
| Этап 3 - портфолио | | | |
| 3 | Разноуровневые задачи и задания | Различают задачи и задания:  а) репродуктивного уровня, позволяющие оценивать и диагностировать  знание фактического материала (базовые понятия, алгоритмы, факты) и умение правильно использовать специальные термины и понятия, узнавание объектов изучения в рамках определенного раздела дисциплины;  б) реконструктивного уровня, позволяющие оценивать и диагностировать умения синтезировать, анализировать, обобщать фактический и теоретический материал с формулированием конкретных выводов, установлением причинно-следственных связей;  в) творческого уровня, позволяющие оценивать и диагностировать умения, интегрировать знания различных областей, аргументировать собственную точку зрения. | Комплект разноуровневых задач и заданий |
| Этап 4 - экзамен | | | |
| 4 | Экзаменационный билет | Комплекс вопросов | Список теоретических вопросов |

* 1. **Требования к структуре и содержанию оценочных средств  
     аттестации в первом семестре**

В соответствии с учебным планом устанавливаются следующие формы контроля: текущий контроль студентов в течение 2 семестра в форме портфолио (прием задач, тестирование в системе NSUts; контрольная работа, выполнение проекта) и промежуточная аттестация во 2 семестре в виде экзамена

Правила оформления лабораторных работ:

− лабораторные работы выполняются на языке Си;

− программа должна сопровождаться кратким описанием в виде комментария в заголовке программы, если она несложная, или отдельного текста, описывающего, что делает эта программа, какой алгоритм использует и как вызывается, с какими параметрами. Каждая функция должна предваряться комментарием с информацией о назначении функции, типах параметров и возвращаемого значения и с кратким описанием алгоритма, если он нетривиален.

− результатом выполнения лабораторной работа является набор файлов на языке Си, если программа сдается устно;

− результатом выполнения лабораторной работа является один файл на языке Си, если программа сдается в автоматизированной системе тестирования NSUts.

− исполняемая программа должна проходить все тесты, подготовленные преподавателем к лабораторной работе, если программа сдается устно.

− исполняемая программа должна проходить все тесты, загруженные в системе NSUts к лабораторной работе, если программа сдается в этой системе.

Требования к представлению и оформлению файлов для системы тестирования

Для всех задач:

Имя входного файла: input.txt

Имя выходного файла: output.txt

Ограничение по времени: 1 секунда на тест

Ограничение по памяти: 64 Мб

Время сдачи на 11 баллов: в первый день выдачи задания

на 10 баллов: в первую неделю

на 8 баллов: во вторую неделю

на 5 баллов: в третью неделю

на 3 балла:

на 1 балл: с начала четвертой недели до конца месяца со дня выдачи задания

до конца семестра

Критерии оценивания ответа студента:

− знание и понимание основных понятий изучаемой темы

− умение формулировать задачу

− умение формализовать задачу

− умение привести примеры из изучаемой темы

− умение корректно излагать смысл проблемы

− базовые навыки рассуждений о понятиях изучаемой темы

− понимание и обоснование теоретических результатов изучаемой темы

− умение применять теоретические результаты в различных ситуациях

− умение эффективно тестировать программы

Оценка промежуточной аттестации для дифзачета выставляется по результатам заданий, входящих в портфолио:

− по результату устного опроса студента в очной форме;

− на основании выполнения заданий по лабораторным работам;

− по результату потоковой контрольной.

Оценка промежуточной аттестации для экзамена выставляется по результатам заданий, входящих в портфолио:

− по результату устного ответа на экзаменационный билет;

− на основании выполнения заданий по лабораторным работам;

− по результату защиты проекта.

Оценка ответа обучающегося по дисциплине «Императивное программирование» является положительной («удовлетворительно» и выше) только в случае положительных оценок по всем вопросам и задачам.

По результатам освоения дисциплины «Императивное программирование» выставляется оценка «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

Задания и подробная инструкция по сдаче решений в систему выкладываются на странице курса

* 1. **Требования к структуре и содержанию оценочных средств во втором семестре**

В соответствии с учебным планом устанавливаются следующие формы контроля: текущий контроль студентов в течение 2 семестра в форме портфолио (прием задач, тестирование в системе NSUts; контрольная работа, выполнение проекта) и промежуточная аттестация во 2 семестре в виде экзамена

Оценка за портфолио формируется:

* по результату устного ответа на экзаменационный билет;
* на основании выполнения заданий по лабораторным работам;
* по результату защиты проекта.

Промежуточная аттестация по дисциплине производится в форме экзамена.

По результатам освоения дисциплины «Императивное программирование» выставляется оценка «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации

2.2.1.Примеры оценочных средств

*Пример экзаменационного билета*

1. Структуры в Си. Синтаксис описания структур. Обращение к полям структур для объектов и к полям по указателю на объект типа структура. Инициализатор структур. Примеры использования структур в алгоритмах.

2. Система непересекающихся множеств (СНМ). Реализация алгоритма Прима с использованием СНМ.

3. В компании из *N* человек составили список пар знакомых. Пусть *N* = 13. После того, как всех пронумеровали, получились следующие пары знакомых друг с другом:

(1, 2), (2, 7), (3, 11), (5, 3), (7, 10), (8, 4), (2, 9), (6, 5), (8, 12)

Доопределим понятие «знакомы» следующим образом. Будем считать двух человек знакомыми, если они знакомы лично, либо у них есть общий знакомый.

Определить минимальное возможное число людей *K*, такое, чтобы при любом выборе *К* человек из *N* имеющихся среди них гарантированно встретились хотя бы двое знакомых (лично или опосредованно).

Требуется:

* найти *K* для данного примера,
* описать алгоритм решения задачи для любого *N* и произвольного непустого набора пар знакомых.

Примеры задач

*Пример задачи, которую требуется сдать в автоматизированную систему тестирования:*

|  |  |
| --- | --- |
| *Имя входного файла:* | *input.txt* |
| *Имя выходного файла:* | *output.txt* |
| *Ограничение по времени:* | *1 секунда на тест* |
| *Ограничение по памяти:* | *64 Мб* |

**Задача «Спички» (Транзитивное замыкание графа)**

Во время игры какое-то количество спичек рассыпается по столу, и игроки должны убрать их, выбирая по одной, не сдвинув с места остальные спички. Вам нужно определить, касаются ли данные спички друг друга. Вам будет дан список координат концов спичек, и нужно будет определить, соединены ли две данные спички или нет. Заметим, что касание — это соединение, и что две спички могут быть соединены, если они соединены не прямо, а посредством других спичек.

Входные данные: Входной файл сдержит несколько тестов. В первой строке каждого теста находится целое число *n* (1 < *n* < 13), задающее количество спичек на столе. Каждая из следующих n строк содержит 4 положительных целых x1, y1, x2, y2, обозначающих координаты (x1, y1), (x2, y2)концов одной спички. Все координаты меньше 100. (Заметим, что спички могут быть разной длины). Первая введенная спичка будет спичкой номер 1, вторая, соответственно, номер 2 и т.д. остальные строки теста содержат по два целых числа a и b (от 1 до n, включительно). Вам нужно определить, связана ли спичка a со спичкой b, или нет. Конец теста: a = b = 0. Все данные корректны и нет спичек нулевой длины. После каждого теста во входном файле пустая строка. Последняя строка в файле содержит один символ — 0.

Выходные данные: Вам нужно для каждой пары спичек сгенерировать строчку CONNECTED, если данные спички связаны, и NOT CONNECTED, наоборот. Будем считать, что спичка связана сама с собой. Пустые строки между тестами не вставлять.

Пример входных данных:

|  |  |
| --- | --- |
| ***input.txt*** | ***output.txt*** |
| **7**  **1 6 3 3**  **4 6 4 9**  **4 5 6 7**  **1 4 3 5**  **3 5 5 5**  **5 2 6 3**  **5 4 7 2**  **1 4**  **1 6**  **3 3**  **6 7**  **2 3**  **1 3**  **0 0**  **0** | **CONNECTED**  **NOT CONNECTED**  **CONNECTED**  **CONNECTED**  **NOT CONNECTED**  **CONNECTED** |

2.2.2 Форма и перечень вопросов экзаменационного билета

**Форма  экзаменационного билета 2 семестра**

Таблица П1.5

|  |
| --- |
| Новосибирский государственный университет  **Экзамен**  Императивное программирование  наименование дисциплины  09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА  Компьютерные науки и системотехника  наименование образовательной программы    **ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №**  1 Вопрос из категории 1  2 Вопрос из категории 2  3 Задача из категории 3  Составитель        \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.Ю.Гатилов (подпись)  Ответственный за образовательную программу  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Д.С.Мигинский  (подпись)  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20     г. |

Перечень вопросов экзамена, структурированный по категориям, представлен в таблице П1.6

Таблица П1.6

|  |  |
| --- | --- |
| Категория | Формулировка вопроса |
| Категория 1 | Системы счисления, реализация алгоритмов перевода |
| Оценка сложности вычислительных алгоритмов и общие методы решения вычислительных задач |
| Перестановки набора, методы поиска всех перестановок |
| Задача поиска, методы поиска в массиве и подстроки в строке |
| Задача сортировки массивов, простые и улучшенные методы сортировки |
| Управление памяти в Си |
| Классические модели динамической памяти: список, стэк, очередь; операции, способы реализации, задачи |
| Бинарные файлы, cортировка файлов |
| Категория 2 | Динамическое программирование |
| Абстрактные структуры данных: графы, деревья, способы реализации |
| Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, реализация |
| Элементы теорий вероятностей, информации и кодирования, реализация проекта |
| В-деревья, красно-черные деревья, реализация проекта |
| Раскраски графа, гамильтоновы и Эйлеровы циклы, алгоритмы и реализация. |
| Потоки в сетях, алгоритмы поиска максимального потока, реализация |
| Элементы теории языков |
|  |
|  |
| Категория 3 | Написать программу, реализующую подсчет суммы десятичных цифр в введенной последовательности символов, заканчивающейся точкой. |
| Написать функцию, осуществляющую печать латинского алфавита. Привести пример вызова этой функции. |
| Для введенной строки создать новую строку, состоящую только из маленьких букв латинского алфавита, входящих в нее. |
| Напиcать функцию char \* my\_strcpy(char \*ad, const char \*as)**,** аналогичную *strcpy* из стандартной библиотеки, которая копирует байты строки *as* в строку *ad*. Привести примеры вызова этой функции. |
| Напиcать функцию int my\_strcmp(const char \*al  > 0  < 0  = 0 |
| Напиcать функцию int my\_strcmp(const char \*al, const char \*ar), аналогичную *strcmp* из стандартной библиотеки, которая сравнивает строки *al* и *ar* и выдает значение:  > 0, если *al* > *ar*  < 0, если *al* < *ar*   = 0, если *al* = *ar* |
|  | Односвязный список задан указателем на первый элемент. Написать функцию, переставляющую элементы списка в обратном порядке. |

Набор экзаменационных билетов формируется и утверждается в установленном порядке в начале учебного года при наличии контингента обучающихся, осваивающих дисциплину «Императивное программирование» в текущем учебном году.

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

Таблица П1.7

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Шифр компетенций** | **Структурные элементы оценочных средств** | **Показатель сформированности** | **Не сформирован** | **Пороговый уровень** | **Базовый уровень** | **Продвинутый**  **уровень** |
| ОПК-8 | Портфолио (этап 1, этап3),  Дифзачет (этап 2),  Вопросы экзаменационного билета (этап 4) | ОПК-8.1 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения | Демонстрирует слабые знания структур данных, не умеет использовать стандартные библиотеки языка Си | Может реализовать основные структуры данных и алгоритмы на языке Си | Может реализовать основные и усложненные структуры данных и алгоритмы на языке Си; имеет навыки работы языка Си и со стандартными библиотеками | Может реализовать основные и усложненные структуры данных и алгоритмы на языке Си; имеет навыки работы языка Си и со стандартными библиотеками |
| ОПК-8 | Портфолио (этап 1, этап3), Дифзачет (этап 2),  Вопросы экзаменационного билета (этап 4) | ОПК-8.2 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы , интегрировать программные модули | Не знает классические алгоритмы и способы их применения | Знает классические алгоритмы, способы их решения, авторов, литературу | Знает классические алгоритмы, способы их решения, авторов, литературу; основы устройства и функционирования операционных систем | Знает классические алгоритмы, способы их решения, авторов, литературу; основы устройства и функционирования операционных систем |
| ОПК-8 | Портфолио (этап 1, этап3), Дифзачет (этап 2), Вопросы экзаменационного билета (этап 2, этап 4) | ОПК-8.3 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы | Не имеет навыка работы в вреде тестирования | имеет навыки работы в системе тестирования NSUts. | имеет хорошие навыки работы в системе тестирования NSUts | имеет хорошие навыки работы в системе тестирования NSUts; придерживается выбранного стиля программирования |

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

В соответствии с учебным планом устанавливаются следующие формы контроля:

в 1 семестре - текущий контроль студентов в течение семестра в форме портфолио и промежуточная аттестация в 1 семестре в виде дифференцированного зачета;

во 2 семестре - текущий контроль студентов в течение семестра в форме портфолио и промежуточная аттестация во 2 семестре в виде экзамена.

Оценка «отлично» соответствует продвинутому уровню сформированности компетенции.

Оценка «хорошо» соответствует базовому уровню сформированности компетенции.

Оценка «удовлетворительно» соответствует пороговому уровню сформированности компетенции.

Оценка «неудовлетворительно» выставляется при неудовлетворительном прохождении одного или двух этапов промежуточной аттестации.

**Лист актуализации фонда оценочных средств промежуточной аттестации**

**по дисциплине  
«Императивное программирование»**

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