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.

РАБОЧАЯ ПРОГРАММА Дисциплины

|  |
| --- |
| **Императивное программирование** |

Направление подготовки: 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Направленность (профиль): Компьютерные науки и системотехника

Форма обучения: очная

Год обучения: 1 семестр: 1, 2

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Вид деятельности** | **Семестр** | |
| **1** | **2** |
| **1** | Лекции, час. | 32 | 32 |
| **2** | Практические занятия, час. | 32 | 32 |
| **3** | Лабораторные занятия, час. | 32 | 32 |
| **4** | Занятий в контактной форме без учета промежуточной аттестации, час, из них | 96 | 98 |
| **5** | в электронной форме, час. |  |  |
| **6** | из них аудиторных занятий, час. | 96 | 96 |
| **7** | из них в активной и интерактивной форме, час. | 96 | 96 |
| **8** | консультаций, час. |  | 2 |
| **9** | Самостоятельная работа, час. | 82 | 116 |
| **10** | в том числе на выполнение письменных работ, час | 50 | 60 |
| **11** | Форма аттестации (экзамен, зачет, дифференцированный зачет), час | ДЗ 2 | Э 2 |
| **12** | Всего зачетных единиц[[1]](#footnote-1) | 5 | 6 |

Новосибирск 2019

Рабочая программа дисциплины составлена на основании федерального государственного образовательного стандарта (ФГОС) высшего образования - бакалавриат по направлению подготовки 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА.

Федеральный государственный образовательный стандарт (ФГОС) высшего образования - бакалавриат по направлению подготовки 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА введен в действие приказом Минобрнауки от 19.09.2017 № 929.

Место дисциплины в структуре учебного плана: Блок 1 Дисциплины (модули); обязательная часть; обязательная дисциплина.

Рабочая программа дисциплины утверждена решением Ученого совета факультета информационных технологий от 02.07.2019, протокол № 75.

Программу разработал:

Доцент

кафедры систем 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

**Аннотация к рабочей программе дисциплины   
«Императивное программирование»**

Дисциплина «Императивное программирование» реализуется в рамках образовательной программы высшего образования – программы бакалавриата 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА, направленность (профиль): Компьютерные науки и системотехника по очной форме обучения на русском языке.

**Место в образовательной программе:**

Дисциплина «Императивное программирование» реализуется в первом и втором семестрах в рамках обязательной части дисциплин (модулей) Блока 1. Для усвоения дисциплины необходимы знания по предметам математического блока дисциплин общеобразовательной программы для среднего образования.

Освоение данной дисциплины необходимо для дисциплин: «Объектно-ориентированное программирование», «Модели вычислений», «Теория параллелизма», «Проектирование программного обеспечения», «Разработка программно-аппаратного комплекса для решения научных и прикладных задач (групповой проект)».

Дисциплина «Императивное программирование» направлена на формирование компетенций:

Способен разрабатывать алгоритмы и программы, пригодные для практического применения (ОПК-8), в части следующих индикаторов достижения компетенции:

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

**Перечень основных разделов дисциплины:**

Дисциплина «Императивное программирование» предусматривает проведение лекций, практических и лабораторных занятий.

**Лекционные занятия**

Используются традиционные технологии, активные и интерактивные формы проведения занятий;

Используется видео, презентации, на которых представлена работа алгоритмов с использованием элементов анимации;

Обсуждаются программные системы с использованием видеопроектора.

**Практические занятия**

Используются традиционные технологии, активные и интерактивные формы проведения занятий, у доски выполняются задания;

Обсуждаются основные конструкции языка Си и их применение при решении различных задач;

С использованием видеопроектора ведется обсуждение реализации алгоритмов;

По некоторым разделам студенты самостоятельно готовят доклады.

**Лабораторные занятия**

Используются традиционные технологии, реализуются на языке Си изученные алгоритмы;

Устно cдаются задачи, представленные преподавателем;

Используется автоматическое компьютерное тестирование программ, задачи сдаются в автоматизированную систему тестирования NSUts.

Дисциплина «Императивное программирование» имеет своей целью: владение языком программирования Си, техникой программирования, знакомство с различными структурами данных, типовыми задачами программирования, основными моделями и методами их решения.

Общий объем дисциплины – 11 зачетных единиц (396 часов)

**Правила аттестации по дисциплине.**

В соответствии с учебным планом устанавливаются следующие формы контроля

1 семестр: Портфолио и диф.зачет.

2 семестр Портфолио и экзамен.

Результаты оцениваются по шкале «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

По результатам аттестации выставляется оценка по шкале «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

**Учебно-методическое обеспечение дисциплины.**

Учебно-методические материалы по дисциплине «Императивное программирование» выложены на странице курса в сети Интернет:

<https://classroom.google.com/u/1/c/MTc2OTM4ODgzNTla>

1. Цикоза, Виталий Аркадьевич. Методы программирования : учебное пособие : [для студентов 1 курса ФИТ и 1-2 курсов Высшего колледжа информатики НГУ : в 2 ч.] / В.А. Цикоза, Т.Г. Чурина ; М-во образования Рос. Федерации, Новосиб. гос. ун-т, Фак. информ. технологий, Каф. систем информатики. Новосибирск : Новосибирский государственный университет, 2003-2006. ; 20 см. Ч.1: Представление и кодирование информации. 2003. 47 с. : ил. (27 экз)
2. Цикоза, Виталий Аркадьевич. Методы программирования : учебное пособие : [для студентов 1 курса ФИТ и 1-2 курсов Высшего колледжа информатики НГУ : в 2 ч.] / В.А. Цикоза, Т.Г. Чурина ; М-во образования Рос. Федерации, Новосиб. гос. ун-т, Фак. информ. технологий, Каф. систем информатики. Новосибирск : Новосибирский государственный университет, 2003-2006. ; 20 см. Ч.2: Перестановки, поиск и сортировка. 2006. 57, [1] с. : ил. ISBN 5-94356-370-9. (45 экз)
3. Чурина, Татьяна Геннадьевна. Методы программирования : алгоритмы и структуры данных : учебное пособие : [для студентов физико-математических специальностей вузов] / Т.Г. Чурина, Т.В. Нестеренко ; М-во образования и науки РФ, Новосиб. гос. ун-т, Фак. информ. технологий, Каф. систем информатики. Новосибирск : Редакционно-издательский центр НГУ, 2014-. ; 20 см. Ч.3: Динамические структуры данных, алгоритмы на графах. 2014. 214 с. : ил. URL: <http://e-lib.nsu.ru/dsweb/Get/Resource-719/page001.pdf>. ISBN 978-5-4437-0278-0.

**1. Внешние требования к дисциплине**

Таблица 1.1

|  |  |
| --- | --- |
| ***Компетенция* ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения,** ***в части следующих индикаторов достижения компетенции:*** | |
| 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 |

**2. Требования к результатам освоения дисциплины**

Таблица 2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)** | **Формы организации занятий** | | | |
| **Лекции** | **Практики / семинары** | **Лабораторные** | **Самостоятельная работа** |
| **ОПК-8.1** Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения | | | | |
| 1. Знать: основные структуры данных и способы их реализации; алгоритмы решения типовых задач программирования, уметь реализовывать основные структуры данных представленные алгоритмы на языке Си; тестировать программы; использовать стандартные функции языка Си | + | + | + | + |
| **ОПК-8.2** Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы , интегрировать программные модули | | | | |
| 2. формулировать задачу и осуществлять поиск в интернет-ресурсах и библиографических материалах, быстро и эффективно реализовывать классические алгоритмы, использовать систему тестирования NSUts. | + | + | + | + |
| 3. владеть навыками работы в текстовых редакторах, интегрированных средах разработки программ, системе NSUts. | + | + | + | + |
| **ОПК-8.3** Владеть: языком программирования; навыками отладки и тестирования работоспособности программы | | | | |
| 4. знать классы сложности типовых алгоритмов; способы оценки объема памяти и быстродействия программ; уметь оценивать объем памяти и быстродействие программ; тестировать программы; использовать автоматизированную систему NSUts. | + | + | + | + |

**3. Содержание и структура учебной дисциплины**

Таблица 3.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Темы лекций** | **Активные формы, час.** | **Часы** | **Ссылки на результаты обучения** | |
| **Семестр: 1** | | | | |
| Лекция 1. Основные вопросы по Си  Краткий обзор базовых типов языка Си. Преобразования между базовыми типами. Связь понятия указателя и массива.  Глобальные и локальные переменные. Модификаторы области видимости и времени жизни.  Разбор порядка вычисления выражения, приоритеты операций | 2 | 2 | | 1, 3, 4 |
| Лекция 2. Основные вопросы по Си  Понятие типа/преобразование типов и структуры. Инициализатор структур.  Функции. Порядок передачи параметров через стек.  Понятие указателя в Си.Операции над указателями.  Строки в языке Си. Функции работы со строками: определение длины строки, копирование строк, слияние строк. | 2 | 2 | | 1, 3, 4 |
| Лекция 3. Основные вопросы по Си  Основные стандартные функции языка Си для работы с файлами. Текстовые и бинарные файлы.  Динамическая память. Функции работы с ДП. | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 4. Системы счисления  Рекуррентные функции перевода "число-запись" и "запись-число"; вычисления по схеме Горнера с минимальным числом операций; алгоритмы перевода целых чисел из одной b-с.с. в другую. Модификации этих алгоритмов для перевода действительных чисел. | 2 | 2 | | 1, 3, 4 |
| Лекция 5. Оценка сложности вычислительных алгоритмов, общие методы решения вычислительных задач  Размер задачи как характеристика объема входных данных. Временная и емкостная сложность программы как функции размера задачи. Верхняя, нижняя и средняя оценки. Классы эффективности алгоритмов: примеры задач, допускающих решение за константное, логарифмическое, линейное, квадратичное, полиномиальное, экспоненциальное время.  Основные методы программирования: повторение (цикл), ветвление и рекурсия. Рекурсивный переход, правила выхода, ветвящаяся и хвостовая рекурсия. Алгоритмы с возвратом. | 2 | 2 | | 1, 3, 4 |
| Лекция 6. Перестановки набора, методы поиска всех перестановок  Перестановки набора. Подсчет числа перестановок. Инверсии. Число инверсий, как мера сложности упорядочения набора; таблицы инверсий; алгоритм восстановления перестановки по таблице инверсий.  Таблица инверсий как число в особой с.с.; итерационный алгоритм генерации всех таблиц инверсий. Алгоритмы перебора перестановок: рекурсивный, итерационный с лексикографическим упорядочением (Дейкстры). | 2 | 2 | | 1, 3, 4 |
| Лекция 7. Задача поиска, методы поиска в массиве и подстроки в строке  Постановка задач поиска в произвольном наборе данных. Методы простого поиска в массиве: линейный поиск, бинарный поиск, оценки сложности.  Методы поиска подстроки в строке: алгоритм прямого перебора. | 2 | 2 | | 1, 3, 4 |
| Лекция 8. Задача поиска, методы поиска в массиве и подстроки в строке  Алгоритм Бойера-Мура, Рабина-Карпа. Оценки сложности. | 2 | 2 | | 1, 3, 4 |
| Лекция 9. Задача сортировки массивов, простые и улучшенные методы сортировки  Постановка задач сортировки (С) записей в произвольном наборе данных. Внешняя и внутренняя постановка задачи Си. Методы сортировки массива: метод простых вставок; метод бинарных вставок; метод простого выбора; метод "пузырька"; шейкер-сортировка; метод Шелла. Оценки сложности. | 2 | 2 | | 1, 3, 4 |
| Лекция 10. Задача сортировки массивов, простые и улучшенные методы сортировки  Быстрая сортировка Хоара; сортировка деревом, пирамидальная сортировка; реализация кучи. Оценки сложности. | 2 | 2 | | 1, 3, 4 |
| Лекция 11. Управление памяти в Си, бинарные файлы  Основные ошибки: утечка памяти, висящие указатели. Стратегии выделения памяти. Внешняя и внутренняя фрагментация. Выделение памяти алгоритмом парных меток. Выделение памяти алгоритмом близнецов. Сборка мусора: область применения, основные методы. | 2 | 2 | | 1, 3, 4 |
| Лекция 12. Классические модели динамической памяти: список, стек, очередь; операции, способы реализации, задачи  Список; как универсальная модель линейно упорядоченных структур данных последовательного доступа; разновидности списков: одно/двусвязные; циклические; иерархические Операции над списками, алгоритмы поиска и включения для списков, анализ их эффективности, способы реализации списков (статика, динамика), реализация алгоритма топологической сортировки на иерархических списках. | 2 | 2 | | 1, 3, 4 |
| Лекция 13. Классические модели динамической памяти: список, стек, очередь; операции, способы реализации, задачи  Стек, преобразование инфиксной формы записи выражения в постфиксную и вычисление значения полученного выражения.  Очередь, Основные операции, способы реализации на различных базовых представлениях. | 2 | 2 | | 1, 3, 4 |
| Лекция 14. Бинарные файлы, сортировка файлов  Бинарные файлы | 2 | 2 | | 1, 3, 4 |
| Лекция 15. Бинарные файлы, сортировка файлов  Cортировка файлов простым двухпутевым слиянием | 2 | 2 | | 1, 3, 4 |
| Потоковая контрольная | 2 | 2 | | 1, 3, 4 |
| **Итого** | **32** | **32** | |  |
| **Семестр: 2** | | | | |
| Лекция 1. Динамическое программирование  Динамическое программирование как решение задач с помощью табличной техники. Примеры задач: о делимости, о гангстерах. Алгоритм Ахо, задача о расстановке скобок в выражении. | 2 | 2 | | 1, 3, 4 |
| Лекция 2. Динамическое программирование  Задачи о рюкзаке, метод ветвей и границ. Задача об умножении матриц. | 2 | 2 | | 1, 3, 4 |
| Лекция 3. Абстрактные структуры данных: графы, деревья, способы реализации  Графы, пути и маршруты, модели представления в ЭВМ: матрицы смежности и инцидентности, динамическая структура со списками дуг, табличное представление. Дерево, как частный вид графа. Дерево двоичного поиска, создание орфографического словаря по заданному набору слов. | 2 | 2 | | 1, 3, 4 |
| Лекция 4. Абстрактные структуры данных: графы, деревья, способы реализации  Уравновешенное дерево. Алгоритмы включения и удаления, сравнение их эффективности с известными алгоритмами на массивах. Левое/правое скобочное представление деревьев. Дерево отрезков, реализация, оценки. | 2 | 2 | | 1, 3, 4 |
| Лекция 5. Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, двусвязные компоненты графа, реализация  Обход вершин графа методом поиска в глубину, реализация. Свойства поиска в глубину, использование стека. Двусвязные компоненты графа, реализация алгоритма поиска двусвязных компонент и точек сочленения графа. Обходы деревьев в глубину: в ре/пост/инфиксном порядке, связь с выражениями. | 2 | 2 | | 1, 3, 4 |
| Лекция 6. Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, двусвязные компоненты графа, реализация  Обход графа методом в ширину. Реализация с помощью очереди, примеры задач. Кратчайшие ипути в лабиринте. Обходы деревьев в ширину слева направо. | 2 | 2 | | 1, 3, 4 |
| Лекция 7. Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, двусвязные компоненты графа, реализация  Топологическая сортировка, реализация на матрице смежности и на еирархических списках. Поиск кратчайших путей в графе, алгоритмы Дейкстры, Беллмана-Форда, Флойда-Уоршелла. Транзитивное замыкание графа. Алгоритм Флойда. Реализация алгоритмов в том числе на куче, оценки сложности. | 2 | 2 | | 1, 3, 4 |
| Лекция 8. Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, двусвязные компоненты графа, реализация  Каркас графа минимальной стоимости, алгоритмы Краскала, Прима, Прима-Краскала. Система непересекающихся множеств, реализация, оценки сложности. | 2 | 2 | | 1, 3, 4 |
| Лекция 9. Элементы теорий вероятностей, информации и кодирования, реализация проекта  Модель информационной системы связи Шеннона. Кодирование как преобразование сообщения при переходе в среду с другим алфавитом. Префиксный код. Однозначная декодируемость префиксного кода. Примеры непрефиксного однозначно декодируемого кода. | 2 | 2 | | 1, 3, 4 |
| Лекция 10. Элементы теорий вероятностей, информации и кодирования, реализация проекта  Формулы Шеннона, Хартли. Избыточность кодирования. Сжатие по Хаффману. Реализация проекта проекта по созданию архиватора. | 2 | 2 | | 1, 3, 4 |
| Лекция 11. В-деревья, красно-черные деревья, реализация проекта  В-деревья, основные определения. Примеры. Реализация на файлах. Алгоритм поиска вершины. | 2 | 2 | | 1, 3, 4 |
| Лекция 12. В-деревья, красно-черные деревья, реализация проекта  Алгоритмы разбиения вершины, добавления и удаления элемента, реализация, оценки. Реализация проекта. | 2 | 2 | | 1, 3, 4 |
| Лекция 13. Раскраски графа, гамильтоновы и Эйлеровы циклы, алгоритмы и реализация.  Раскраска графа. Задача о составлении расписаний, о распределении ресурсов. Алгоритм последовательной раскраски. Эйлеров граф. Алгоритм Флери, реализация, оценки. Поиск Эйлерова цикла за время O(n+m), реализация. Гамильтонов цикл, реализация перебором. | 2 | 2 | | 1, 3, 4 |
| Лекция 14. Потоки в сетях, алгоритмы поиска максимального потока, реализация  Максимальный поток в сети, примеры, сокращение потока, несколько истоков, остаточные сети, дополняющие пути, разрезы. Метод Форда-Фалкерсона, реализация, оценки. | 2 | 2 | | 1, 3, 4 |
| Лекция 15. Потоки в сетях, алгоритмы поиска максимального потока, реализация  Задача о максимальном паросочетании, реализация. Алгоритм проталкивания предпотока, реализация, оценки. | 2 | 2 | | 1, 3, 4 |
| Лекция 16. Элементы теории языков  Символьные формализмы для описания синтаксиса языков: БНФ, РБНФ. Формальные грамматики. Синтаксический анализатор. Нисходящий и восходящий разбор. Полный перебор правил подстановки. | 2 | 2 | | 1, 3, 4 |
| **Итого** | **32** | **32** | |  |

Таблица 3.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Темы практических занятий** | **Активные формы, час.** | **Часы** | **Ссылки на результаты обучения** | **Учебная деятельность** |
| **Семестр: 1** | | | | | |
| Тема 1. Основные вопросы по Си | 2 | 2 | 1, 2, 3, 4 | Основные типы данных (целые и вещественные) и операции над ними. Выражения. Операторы присваивания, условный, выбора. Условное выражение. Вид программы на Си. Решение задач. |
| Тема 2. Основные вопросы по Си | 2 | 2 | 1, 2, 3, 4 | Циклы for, while, do-while. Операторы выхода из цикла break и continue. Решение задач. |
| Тема 3. Основные вопросы по Си | 2 | 2 | 1, 2, 3, 4 | Линейные массивы. Описание одномерных массивов, инициализация. Операции над одномерными массивами. Задание массива случайным образом. Прямой поиск в массиве. Поиск методом деления пополам |
| Тема 4. Основные вопросы по Си | 2 | 2 | 1, 2, 3, 4 | Функции. Описание функций. Параметры, области видимости переменных. Виды переменных (локальные, глобальные, статические, автоматические, регистровые, внешние). Передача массива параметром в функцию. Рекурсивные функции (вычисление факториала, Ханойские башни). Алгоритм Евклида (итеративный и рекурсивный). Реализация. |
| Тема 5. Основные вопросы по Си | 2 | 2 | 1, 2, 3, 4 | Литерный тип. Описание литерного типа, литерные константы. Таблица ASCII. Функции ввода-вывода символов. Решение задач |
| Тема 6. Основные вопросы по Си | 2 | 2 | 1, 2, 3, 4 | Строки. Описание строк, инициализация. Функции ввода-вывода. Пример реализации некоторых стандартных функций: strlen, strcmp, strcpy |
| Тема 7. Системы счисления | 2 | 2 | 1, 2, 3, 4 | Перевод из одной системы счисления в другую. Кратные системы счисления, особенности реализации на строках |
| Тема 8. Основные вопросы по Си | 2 | 2 | 1, 2, 3, 4 | Представление числовых данных в Си, байты и биты, битовые операции, реализация битовых массивов, битовые поля. |
| Тема 9. Основные вопросы по Си | 2 | 2 | 1, 2, 3, 4 | Многомерные массивы. Описание многомерного массива, операции над ним, инициализация. Ввод массива из файла. Передача массива в качестве параметра. Решение задач |
| Тема 10. Основные вопросы по Си | 2 | 2 | 1, 2, 3, 4 | Файлы, описание, виды файлов, работа с текстовыми файлами, решение задач |
| Тема 11. Основные вопросы по Си | 2 | 2 | 1, 2, 3, 4 | Структуры в Си, описание, инициализация, способы работы со структурами, решение задач |
| Тема 12. Управление памятью в Си | 2 | 2 | 1, 2, 3, 4 | Динамическая память, функции работы с динамической памятью, выделение памяти, реализация динамических массивов, решение задач |
| Тема 13. Классические модели динамической памяти: список, стек, очередь; операции, способы реализации, задачи | 2 | 2 | 1, 2, 3, 4 | Динамические списки, способы создания и обходов. Преобразование списков. |
| Тема 14. Классические модели динамической памяти: список, стек, очередь; операции, способы реализации, задачи | 2 | 2 | 1, 2, 3, 4 | Линейные списки, стеки, очереди. Способы их реализации на динамических списках и на массивах |
| Тема 15. Основные вопросы по Си, управление памятью | 4 | 4 | 1, 2, 3, 4 | Описание аргументов функции main. Создание проекта. |
| **Итого:** | **32** | **32** |  |  |
| **Семестр: 2** | | | | |
| Тема 1. Динамическое программирование | 2 | 2 | 1, 2, 3, 4 | Функции с переменным числом параметров (доклад студента).  Решение задач на списки |
| Тема 2. Динамическое программирование | 2 | 2 | 1, 2, 3, 4 | Динамическое программирование. Решение простых задач, обсуждение задач для самостоятельного решения |
| Тема 3 Абстрактные структуры данных: графы, деревья, способы реализации | 2 | 2 | 1, 2, 3, 4 | Деревья, представление деревьев, обходы, способы реализации.  Задачи на создание и обходы деревьев. |
| Тема 4. Абстрактные структуры данных: графы, деревья, способы реализации | 2 | 2 | 1, 2, 3, 4 | Способы представления графов. Перевод из одного представления в другое. Способы построения сбалансированного дерева. |
| Тема 5. Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, реализация | 2 | 2 | 1, 2, 3, 4 | Метод поиска в глубину. Задача – проверить, является ли граф, заданный матрицей смежности, деревом. |
| Тема 6. Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, реализация | 2 | 2 | 1, 2, 3, 4 | Обход дерева в ширину с использованием очереди. Обход графа в ширину |
| Тема 7. Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, реализация | 2 | 2 | 1, 2, 3, 4 | Топологическая сортировка, три метод реализации: на матрице смежности, на иерархических списках, методом поиска в глубину. |
| Тема 8. Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, реализация | 2 | 2 | 1, 2, 3, 4 | Поиск кратчайших путей в графе. Алгоритм Дейкстры. Реализация кучи. |
| Тема 9. Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, реализация | 2 | 2 | 1, 2, 3, 4 | Транзитивное замыкание графа и алгоритм Флойда. Методы реализации. |
| Тема 10. Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, реализация | 2 | 2 | 1, 2, 3, 4 | Система непересекающихся множеств. Использование СНМ в алгоритме Краскала построения минимального остовного дерева. Использование кучи в алгоритме Прима. |
| Тема 11. Элементы теорий вероятностей, информации и кодирования, реализация проекта | 2 | 2 | 1, 2, 3, 4 | Обсуждение реализации проекта Архиватор. Построение кодирующего дерева. Работа с бинарными файлами (доклад студента). Способы записи и воспроизведения закодированной информации. |
| Тема 12. В-деревья, красно-черные деревья, реализация проекта | 2 | 2 | 1, 2, 3, 4 | Обсуждение проекта Каталог на B-деревьях.  Директивы препроцессора (доклад студента) |
| Тема 13 В-деревья, красно-черные деревья, реализация проекта | 2 | 2 | 1, 2, 3, 4 | Красно-черные деревья, способы реализации.  Тест на знание языка Си |
| Тема 14. Раскраски графа, гамильтоновы и Эйлеровы циклы, алгоритмы и реализация | 2 | 2 | 1, 2, 3, 4 | Способы поиска Эйлеровых циклов в графе, реализация на стеке.  Задачи полного перебора на примере поиска гамильтонова цикла в графе. |
| Тема 15. Двусвязные компоненты графа | 2 | 2 | 1, 2, 3, 4 | Реализация алгоритма последовательной раскраски графа. Использование метода поиска в глубину для поиска двусвязных компонент и точек сочленения |
| Контрольная работа. | 2 | 2 | 1, 2, 3, 4 |  |
| **Итого** | **32** | **32** |  |  |

Таблица 3.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Темы лабораторных занятий** | **Активные формы, час.** | **Часы** | **Ссылки на результаты обучения** | **Учебная деятельность** |
| Тема 1. Основные вопросы по Си, управление памятью | 2 | 2 | 1, 2, 3, 4 | Задачи без циклов, числовые типы данных. Реализация задач:  1. Треугольник  2. Скорая помощь  3. Корни  4. Цифра |
| Тема 2. Основные вопросы по Си, управление памятью | 2 | 2 | 1, 2, 3, 4 | Задачи с циклами, числовые типы данных. Реализация задач:  1. Сумма ряда  2. Обращение числа  3. Простые числа  4. Близкий к целому  5. Цифра |
| Тема 3 Задача поиска, методы поиска в массиве | 2 | 2 | 1, 2, 3, 4 | Одномерные массивы. Реализация задач:  1. Границы массива  2. Решето Эратосфена  3. Даты  4. Бинарный поиск  5. Подпоследовательности  6. Различные числа  7. Циклический сдвиг  8. Бинарный поиск 2 |
| Тема 4. Задача сортировки массивов, простые методы сортировки | 2 | 2 | 1, 2, 3, 4 | Функции, рекурсия, простые сортировки. Реализация задач:  1. Сортировка выбором  2. Сортировка вставками  3. Шейкер-сортировка  4. НОД  5. НОК |
| Тема 5. Основные вопросы по Си, управление памятью | 2 | 2 | 1, 2, 3, 4 | Литерный тип. Реализация задач:  1. Что бы ты ни сделал, я могу сделать лучше  2. Слова из трех букв  3. Делимость на 15  4. Сумма чисел |
| Тема 6. Основные вопросы по Си, управление памятью | 2 | 2 | 1, 2, 3, 4 | Строки. Реализация задач:  1. Игра в алфавит  2. Смешивание цветов  3. Топики  4. Разность множеств  5. Слова |
| Тема 7. Системы счисления (с.с.) | 2 | 2 | 1, 2, 3, 4 | Перевод из одной системы счисления в другую. Реализация задач:  1. Целое: значение -> запись  2. Целое: запись -> значение  3. Дробное: значение -> запись  4. Дробное: запись -> значение  5. Запись -> запись |
| Тема 8. Модели машинной арифметики с конечной разрядностью | 2 | 2 | 1, 2, 3, 4 | Биты и байты. Реализация задач:  1. Количество единиц  2. Обращение числа  3. Битовый массив |
| Тема 9. Многомерные массивы. | 2 | 2 | 1, 2, 3, 4 | Многомерные массивы. Реализация задач:  1. Японский кроссворд  2. Шифр Шерлока Холмса |
| Тема 10. Перестановки набора, методы поиска всех перестановок | 2 | 2 | 1, 2, 3, 4 | Перестановки. Реализация задач:  1. Следующая по алфавиту перестановка  2. Таблица инверсий  3. Восстановление перестановки |
| Тема 11. Задача сортировки массивов, улучшенные методы сортировки | 2 | 2 | 1, 2, 3, 4 | Улучшенные сортировки. Реализация:  1. Быстрая сортировка  2. Сортировка Шелла  3. Пирамидальная сортировка |
| Тема 12. Основные вопросы по Си, управление памятью | 2 | 2 | 1, 2, 3, 4 | Текстовые файлы. Реализация задач:  1. Арифметическая прогрессия  2. Количество символов  3. Количество слов  4. Количество строк |
| Тема 13. Классические модели динамической памяти: список, стек, очередь; операции, способы реализации, задачи | 2 | 2 | 1, 2, 3, 4 | Односвязные списки. Реализация задач:  1. Построение списка добавлением в хвост  2. Построение списка добавлением в голову  3. Построение упорядоченного списка |
| Тема 14. Классические модели динамической памяти: список, стек, очередь; операции, способы реализации, задачи | 2 | 2 | 1, 2, 3, 4 | Преобразование списков. Реализация задач:  1. Удаление из списка  2. Задача Иосифа  3. Слияние списков |
| Тема 15. Классические модели динамической памяти: список, стек, очередь; операции, способы реализации, задачи | 2 | 2 | 1, 2, 3, 4 | Линейные списки: стеки, очереди. Задачи:  1. Перевод из инфиксной формы в постфиксную  2. Вычисление на стеке  3. Циклический буфер |
| Тема 16. Классические модели динамической памяти: список, стек, очередь; операции, способы реализации, задачи | 2 | 2 | 1, 2, 3, 4 | Реализация проекта Калькулятор |
| **Итого:** | **32** | **32** |  |  |
| **Семестр: 2** | | | | |
| Тема 1. Динамическое программирование | 2 | 2 | 1, 2, 3, 4 | Реализация функции с переменным числом параметров my\_printf |
| Тема 2-3. Динамическое программирование | 4 | 4 | 1, 2, 3, 4 | Динамическое программирование.  Задачи: 1. Рюкзак 2. Делимость 3. Гангстеры 4. Телефонный номер |
| Тема 4. Абстрактные структуры данных: графы, деревья, способы реализации | 2 | 2 | 1, 2, 3, 4 | Деревья.  Задачи: 1. Дерево двоичного поиска 2. Обходы дерева 3. Дерево-формула |
| Тема 5. Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, реализация | 2 | 2 | 1, 2, 3, 4 | Обход графа в глубину. Задачи:  1. Компоненты связности  2. Лабиринт  3. Знакомства |
| Тема 6. Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, реализация | 2 | 2 | 1, 2, 3, 4 | Обход графа в ширину. Задачи:  1. Лабиринт  2. Встреча шаманов |
| Тема 7. Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, реализация | 2 | 2 | 1, 2, 3, 4 | Топологическая сортировка. Задачи:  1. Наложение рамок  2. Топологическая сортировка |
| Тема 8-9. Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, реализация | 4 | 4 | 1, 2, 3, 4 | Кратчайшие пути в графе. Задачи:  1. Шлю я за пакетом пакет…  2. Поиск пути  3. Сплетницы  4. Авиаперелеты |
| Тема 10. Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, реализация | 2 | 2 | 1, 2, 3, 4 | Транзитивное замыкание графа. Задачи:  1. Спички  2. Фиолетовое такси |
| Тема 11-12. Алгоритмы перебора на абстрактных динамических структурах: обходы, минимальные пути, остовные деревья, реализация | 4 | 4 | 1, 2, 3, 4 | Каркасы в графе, СНМ. Задачи:  1. Highways  2. Дорожки |
| Тема 13-15. Элементы теории вероятностей, информации и кодирования, реализация проекта  В-деревья, красно-черные деревья, реализация проекта | 6 | 6 | 1, 2, 3, 4 | Работа над одним из проектов:  − Архиватор Хаффмана,  − Каталог файлов (B-деревья) |
| Сдача проекта | 2 | 2 | 1, 2, 3, 4 |  |
| **Итого:** | **32** | **32** |  |  |

**4. Самостоятельная работа студентов**

Таблица 4.1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Виды самостоятельной работы** | | **Ссылки на результаты обучения** | | **Часы на выполнение** | | **Часы на консультации** | |
| **Семестр: 1** | | | | | | | | |
| 1 | Изучение темы дисциплины по учебной литературе, учебным пособиям, поиск в интернете | | 1, 2, 3, 4 | | 20 | |  | |
| Изучение предлагаемых теоретических разделов в соответствии с настоящей Программой. Учебно-методические материалы по дисциплине «Императивное программирование» выложены на странице курса в сети Интернет | | | | | | | |
| 2 | Подготовка к практическим занятиям, к лабораторным занятиям, к текущему контролю знаний и промежуточной аттестации | | 1, 2, 3, 4 | | 50 | |  | |
| Разбор решенных задач, самостоятельное решение задач, реализация проекта | | | | | | | |
| 3 | Подготовка к дифференцированному зачету | |  | | 12 | |  | |
| Повторение теоретического материала по вопросам, совпадающим с темами лекций | | | | | | | |
| **Итого** |  | | | **82** | |  | |
| **Семестр: 2** | | | | | | | | |
| 1 | Изучение темы дисциплины по учебной литературе, учебным пособиям, поиск в интернете | | 1, 2, 3, 4 | | 20 | |  | |
| Изучение предлагаемых теоретических разделов в соответствии с настоящей Программой. Учебно-методические материалы по дисциплине «Императивное программирование» выложены на странице курса в сети Интернет | | | | | | | |
| 2 | Подготовка к практическим занятиям, к лабораторным занятиям, к текущему контролю знаний и промежуточной аттестации | | 1, 2, 3, 4 | | 60 | |  | |
| Разбор решенных задач, самостоятельное решение задач, реализация проекта | | | | | | | |
| 3 | Подготовка к экзамену | | 1, 2, 3, 4 | | 36 | | 2 | |
| Повторение теоретического материала по вопросам, совпадающим с темами лекций | | | | | | | |
| **Итого** | | |  | | **116** | | **2** |

**5. Образовательные технологии**

В ходе реализации учебного процесса по дисциплине проводятся лекционные, лабораторные и практические занятия. Темы, рассматриваемые на лекциях и изучаемые самостоятельно, закрепляются на практических и лабораторных занятиях.

В ходе реализации учебного процесса по дисциплине применяются лекционные, лабораторные и практические занятия, а также применяются следующие интерактивные формы обучения (таблица 5.1).

Таблица 5.1

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | | Лекция в форме дискуссии | ОПК-8 |
| **Формируемые умения:** знать способы реализации структур данных; способы использования языка Си и стандартных библиотек языка Си, знать классические алгоритмы и уметь их обоснованно применять и их применение для конкретных ситуаций. | | | |
| **Краткое описание применения:** Обсуждение, в контексте изученной теории, практического применения различных аспектов языка программирования Си, техники программирования, различных структур данных, разбор типовых задач программирования, основных моделей и методов их решения | | | |
| **2** | Портфолио | | ОПК-8 |
| **Формируемые умения:**  владеть способами реализации структур данных; навыками использования языка Си и стандартных библиотек языка Си  знать классические алгоритмы и уметь их обоснованно применять  владеть различными стилями программирования | | | |
| **Краткое описание применения:** бакалавры ведут портфолио (оценки за задания, контрольные работы, тестирование, проект), которое является основой для проведения аттестации по дисциплине | | | |

Для организации и контроля самостоятельной работы бакалавров, а также проведения консультаций применяются информационно-коммуникационные технологии (таблица 5.2).

Таблица 5.2

|  |  |
| --- | --- |
| Информирование | Адрес почты – сообщается бакалаврам на первом занятии. |
| Консультирование | Адрес почты – сообщается бакалаврам на первом занятии. |
| Контроль | Адрес почты – сообщается бакалаврам на первом занятии. |
| Размещение учебных материалов | <https://classroom.google.com/u/1/c/MTc2OTM4ODgzNTla> |

**6. Правила аттестации студентов по учебной дисциплине**

В соответствии с учебным планом устанавливаются следующие формы контроля

1 семестр: Портфолио и диф.зачет.

2 семестр Портфолио и экзамен.

Оценка промежуточной аттестации для диф.зачета 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.

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 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.

При оценке ответа обучающегося преподаватель руководствуется шкалами оценивания из Приложения к настоящей программе дисциплины.

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

В таблице 6.1 представлено соответствие форм аттестации заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Коды компетенций ФГОС** | **Результаты обучения** | **Формы аттестации** | | | |
| **семестр 1** | | **семестр 2** | |
| портфолио | дифзачет | портфолио | экзамен |
| **ОПК-8** | ОПК-8.1 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения | **+** | **+** | **+** | **+** |
| **ОПК-8** | ОПК-8.2 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы , интегрировать программные модули | **+** | **+** | **+** | **+** |
| **ОПК-8** | ОПК-8.3 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы | **+** | **+** | **+** | **+** |

Требования к структуре и содержанию портфолио, оценочные средства, а также критерии оценки сформированности компетенций и освоения дисциплины в целом, представлены в Фонде оценочных средств, являющемся приложением 1 к настоящей рабочей программе дисциплины.

**7. Литература**

1. Керниган, Б.В., Д.М. Ричи Язык программирования C / М. : Интернет-Университет Информационных Технологий, 2006. - 272 с. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=234039>.

*Интернет-ресурсы*

Таблица 7.1

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование Интернет-ресурса | Краткое описание |
|  | <http://www.spsl.nsc.ru> | Портал ГПНТБ СОРАН |
|  | <https://olympic.nsu.ru/nsuts-new/login.cgi> | Система тестирования NSUts |

**8. Учебно-методическое и программное обеспечение дисциплины**

**8.1. Учебно-методическое обеспечение**

Учебно-методические материалы по дисциплине «Императивное программирование» выложены на странице курса в сети Интернет:

<https://classroom.google.com/u/1/c/MTc2OTM4ODgzNTla>

1. Цикоза, Виталий Аркадьевич. Методы программирования : учебное пособие : [для студентов 1 курса ФИТ и 1-2 курсов Высшего колледжа информатики НГУ : в 2 ч.] / В.А. Цикоза, Т.Г. Чурина ; М-во образования Рос. Федерации, Новосиб. гос. ун-т, Фак. информ. технологий, Каф. систем информатики. Новосибирск : Новосибирский государственный университет, 2003-2006. ; 20 см. Ч.1: Представление и кодирование информации. 2003. 47 с. : ил. (27 экз)
2. Цикоза, Виталий Аркадьевич. Методы программирования : учебное пособие : [для студентов 1 курса ФИТ и 1-2 курсов Высшего колледжа информатики НГУ : в 2 ч.] / В.А. Цикоза, Т.Г. Чурина ; М-во образования Рос. Федерации, Новосиб. гос. ун-т, Фак. информ. технологий, Каф. систем информатики. Новосибирск : Новосибирский государственный университет, 2003-2006. ; 20 см. Ч.2: Перестановки, поиск и сортировка. 2006. 57, [1] с. : ил. ISBN 5-94356-370-9. (45 экз)
3. Чурина, Татьяна Геннадьевна. Методы программирования : алгоритмы и структуры данных : учебное пособие : [для студентов физико-математических специальностей вузов] / Т.Г. Чурина, Т.В. Нестеренко ; М-во образования и науки РФ, Новосиб. гос. ун-т, Фак. информ. технологий, Каф. систем информатики. Новосибирск : Редакционно-издательский центр НГУ, 2014-. ; 20 см. Ч.3: Динамические структуры данных, алгоритмы на графах. 2014. 214 с. : ил. URL: <http://e-lib.nsu.ru/dsweb/Get/Resource-719/page001.pdf>. ISBN 978-5-4437-0278-0.

**8.2. Программное обеспечение**

Для обеспечения реализации дисциплины используется стандартный комплект программного обеспечения (ПО), включающий регулярно обновляемое лицензионное ПО Windows и MS Office.

Специализированное программное для освоения дисциплины обеспечение не требуется.

**9. Профессиональные базы данных и информационные справочные системы**

1. Полнотекстовые журналы Springer Journals за 1997-2015 г., электронные книги (2005-2016 гг.), коллекция научных биомедицинских и биологических протоколов SpringerProtocols, коллекция научных материалов в области физических наук и инжиниринга SpringerMaterials, реферативная БД по чистой и прикладной математике zbMATH.
2. Электронная библиотека диссертаций Российской государственной библиотеки (ЭБД РГБ)
3. Электронные ресурсы Web of Science Core Collection (Thomson Reuters Scientific LLC.), Journal Citation Reports + ESI
4. БД Scopus (Elsevier)

**10. Материально-техническое обеспечение**

Таблица 10.1

|  |  |  |
| --- | --- | --- |
| **№** | **Наименование** | **Назначение** |
| 1 | Презентационное оборудование (мультимедиа-проектор, экран, компьютер для управления) | Для проведения лекционных занятий |
| 2 | Компьютерный класс (с выходом в Internet) | Для проведения лабораторных занятий и организации самостоятельной работы обучающихся |

Материально-техническое обеспечение образовательного процесса по дисциплине для обучающихся из числа лиц с ограниченными возможностями здоровья осуществляется согласно «Порядку организации и осуществления образовательной деятельности по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья в Новосибирском государственном университете».

**Лист актуализации рабочей программы дисциплины  
«Императивное программирование»**

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

1. С учетом выделенных часов на промежуточную аттестацию [↑](#footnote-ref-1)