Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Национальный исследовательский ядерный университет «МИФИ»

ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

КАФЕДРА КИБЕРНЕТИКИ

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ**

**ПО ДИСЦИПЛИНЕ**

ИНФОРМАТИКА (ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ, АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ, ПРОЕКТИРОВАНИЕ ЧЕЛОВЕКО-МАШИННОГО ИНТЕРФЕЙСА)

|  |  |
| --- | --- |
| Направление подготовки (специальность) | 09.03.04 Программная инженерия |
| Профиль подготовки | *[при его наличии]* |
| Наименование образовательной программы (специализация) | Математическое и программное обеспечение вычислительных машин и компьютерных сетей |
| Квалификация (степень) выпускника | бакалавр |
| Форма обучения | очная |

**АННОТАЦИЯ**

В курсе изучаются фундаментальные элементы современного программирования: базовые структуры данных, такие, как списки и массивы, и производные от них; возможности, организация и алгоритмы работы полиморфных структур данных; приемы модульного программирования; принципы абстрагирования и последовательной детализации на примере разработки абстрактных типов данных (АТД). Вырабатываются практические навыки работы с промышленными средами разработки. Изучаются элементы модульного тестирования. Приобретаются навыки создания простейших пользовательских интерфейсов. В рамках курса студенты выполняют индивидуальные задания, которые формируются на основе их собственного выбора, что обеспечивает элементы индивидуальности учебной траектории студента в рамках курса.

**СТРУКТУРА ДИСЦИПЛИНЫ**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Семестр** | **Интерактив** | **Трудоемкость, кред.** | **Общий объем курса, час.** | **Лекции, час.** | **Практич. занятия, час.** | **Лаборат. работы, час.** | **СРС, час.** | **КСР, час.** | **Форма(ы) контроля, экз./зач./КР/КП** |
| 2 |  | 5 | 180 | 15 | 15 | 30 | 66 | 0 | Э |
| ИТОГО | 0 | 5 | 180 | 15 | 15 | 30 | 66 | 0 |  |

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

**Указания к выполнению лабораторных работ**

По курсу предусмотрено выполнение трех лабораторных работ в формате самостоятельного выполнения индивидуального задания по каждой работе. Состав заданий и методические указания приводятся в соответствующих приложениях – по каждой лабораторной работе. На выполнение каждой лабораторной отводится по 5 недель. В процессе выполнения на любом этапе студент может получать консультации по всем вопросам, касающимся выполнения задания. Выполнение лабораторной работы оценивается по итогам защиты, которая проводится в форме устного собеседования с обязательной демонстрацией программы в действии и разбором исходного кода.

**Методика оценки результатов выполнения лабораторных работ**

Общая методика оценки выполнения индивидуального задания приведена ниже. В каждой лабораторной работе она может незначительно корректироваться.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Качество программного кода: | * стиль (в т.ч.: имена, отступы и проч.) (0-2) * структурированность (напр. декомпозиция сложных функций на более простые) (0-2) * качество основных и второстепенных алгоритмов (напр. обработка граничных случаев и некорректных исходных данных и т.п.) (0-3) | 0-5  баллов |
|  | Качество пользовательского интерфейса: | * предоставляемые им возможности (0-2) * наличие ручного/автоматического ввода исходных данных (0-2) * настройка параметров для автоматического режима   отображение исходных данных и промежуточных и конечных результатов и др. (0-2) | 0-5  баллов |
|  | Качество тестов | * степень покрытия * читаемость * качество проверки (граничные и некорректные значения, и др.) | 0-5  баллов |
|  | Полнота выполнения задания и качество ТЗ | Оценивается качество подготовки ТЗ, полнота выполнений минимальных требований | 0-5  баллов |
|  | Владение теорией | знание алгоритмов, области их применимости, умение сравнивать с аналогами, оценить сложность, корректность реализации | 0-3  баллов |
|  | Оригинальность реализации | оцениваются отличительные особенности конкретной реализации – например, общность структур данных, наличие продвинутых графических средств, средств ввода-вывода, интеграции с внешними системами и др. | 0-5  баллов |
|  | Итого | | 0-30  баллов |

**Список вопросов к экзамену**

1. Классификация структур данных. Последовательности и декартовы произведения.
2. Структуры данных: разновидности последовательностей. Последовательности с произвольным доступом. Примеры.
3. Структуры данных: разновидности последовательностей. Последовательности с последовательным доступом. Примеры.
4. Массивы в языки C. Использовать указателей для реализации массивов.
5. Полиморфизм в C. Использование void\* для реализации массивов в языке C.
6. Указатели на функции. Массивы функций. Примеры на языках C и C++.
7. Абстрактные типы данных. Реализация абстрактных типов данных в языке C. Сокрытие деталей объявления и реализации с помощью заголовочных файлов.
8. Полиморфные типы данных в C++. Шаблоны.
9. Односвязные списки. Операция поиска элемента по номеру. Алгоритм, оценка алгоритмической сложности. Пример реализации на языках C и C++.
10. Односвязные списки. Операция вставки. Алгоритм, оценка алгоритмической сложности. Пример реализации на языках C и C++.
11. Односвязные списки. Операция удаления. Алгоритм, оценка алгоритмической сложности. Пример реализации на языках C и C++.
12. Двусвязные списки. Операция поиска элемента по номеру. Алгоритм, оценка алгоритмической сложности. Пример реализации на языках C и C++.
13. Двусвязные списки. Операция вставки. Алгоритм, оценка алгоритмической сложности. Пример реализации на языках C и C++.
14. Двусвязные списки. Операция удаления. Алгоритм, оценка алгоритмической сложности. Пример реализации на языках C и C++.
15. Кольцевые односвязные и двусвязные списки. Операция поиска элемента по номеру. Алгоритм, оценка алгоритмической сложности. Пример реализации на языках C и C++.
16. Кольцевые односвязные и двусвязные списки. Операция вставки. Алгоритм, оценка алгоритмической сложности. Пример реализации на языках C и C++.
17. Связные Кольцевые односвязные и двусвязные списки. Операция удаления. Алгоритм, оценка алгоритмической сложности. Пример реализации на языках C и C++.
18. Очередь как разновидность связного списка. Реализации операций enqueue. Пример реализации на языках C и C++.
19. Очередь как разновидность связного списка. Реализации операций dequeue. Пример реализации на языках C и C++.
20. Стек как разновидность связного списка. Реализация операций push. Пример реализации на языках C и C++.
21. Стек как разновидность связного списка. Реализация операций pop. Пример реализации на языках C и C++.
22. Дек как разновидность связного списка. Реализация операций вставки. Пример реализации на языках C и C++.
23. Дек как разновидность связного списка. Реализация операций удаления. Пример реализации на языках C и C++.
24. Реализация коллекции с произвольным доступом с помощью массивов. Реализация операции вставки. Пример реализации на языках C и C++.
25. Реализация коллекции с произвольным доступом с помощью массивов. Реализация операции удаления. Пример реализации на языках C и C++.
26. Бинарное дерево. Поиск элемента. Алгоритм, оценка сложности.
27. Бинарное дерево. Вставка элемента. Алгоритм, оценка сложности.
28. Бинарное дерево. Удаление элемента. Алгоритм, оценка сложности
29. Бинарное дерево. Обходы: ЛКП, ПКЛ, КЛП, КПЛ
30. Бинарное дерево: восстановление по обходу.
31. Работа с последовательностями: операция where. Сравнение реализации на массивах и на списках.
32. Работа с последовательностями: операция map. Сравнение реализации на массивах и на списках.
33. Работа с последовательностями: операция reduce. Сравнение реализации на массивах и на списках.