**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**

**Филиал федерального государственного бюджетного образовательного учреждения высшего образования «Мурманский арктический государственный университет»**

**в г. Апатиты**

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

**Б1.Б.12 Технологии программирования**

(шифр дисциплины и название в строгом соответствии

с федеральным государственным образовательным стандартом и учебным планом)

**образовательной программы
по направлению подготовки бакалавриата**

 **09.03.02 «Информационные системы и технологии»**

**профиль: Общий профиль**

(код и наименование направления подготовки

с указанием профиля (наименования магистерской программы)

**очная форма обучения**

 форма обучения

|  |  |
| --- | --- |
| **Составитель:** Тоичкин Н.А., к.т.н., доцент кафедрыинформатики, вычислительной техники и информационной безопасности | Утверждено на заседании кафедры информатики, вычислительной техники и информационной безопасности (протокол № \_\_ от \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ 2016г.)Зав. кафедрой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ подпись Ф.И.О.  |

**Структура рабочей программы дисциплины**

**1. НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ). Б1.Б.12 Технологии программирования**

**2. АННОТАЦИЯ К ДИСЦИПЛИНЕ**

Основной цельюизучения дисциплины "Технологии программирования" является освоение студентами методов, алгоритмов и технологий в области программирования на языках высокого уровня.

В результате изучения дисциплины студент должен:

*знать:*

* методы формализации, алгоритмизации и реализации задач на ЭВМ;
* способы записи алгоритмов и конструирования программ с использованием различных алгоритмических языков;
* основные структуры представления данных как статические (запись, объект, массив), так и динамические (список, дерево и т.п.), и основные приемы и алгоритмы работы с ними;
* модели решения некоторых функциональных задач;
* методы оценки эффективности алгоритмов;
* основные принципы структурной и объектно-ориентированной методологий программирования;
* основы доказательства правильности программ;
* основные принципы модульного программирования.

*уметь:*

* использовать методы и средства разработки алгоритмов и программ, современные технологии программирования;
* формализовать и формулировать задачу для решения конкретной проблемы;
* правильно и обоснованно выбирать алгоритм решения задачи;
* использовать рекурсивные и итерационные алгоритмы;
* программировать древовидные и графовые структуры данных.

*владеть:*

* методикой определения сложности алгоритмов;
* выбором технологии программирования и инструментальных программных средств высокого уровня;
* организацией данных в виде динамических структур (списков, кольцевых списков, бинарных деревьев) и методов работы с ними;
* декомпозицией разбиения задачи на отдельные модули;
* навыками отладки программных модулей средствами среды программирования.

**3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ** **РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ**

В результате освоении содержания дисциплины «Технологии программирования» обучающийся должен обладать следующими компетенциями:

* владением широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий (**ОПК-1**).

**4. УКАЗАНИЕ МЕСТА ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ**

Дисциплина входит в базовую часть профессионального цикла образовательной программы бакалавра по направлению «Информационные системы и технологии».

Для освоения данной дисциплины обучающиеся используют знания, умения, навыки, способы деятельности и установки, которые они получили в процессе изучения дисциплин: «Информатика», «Программирование на языках высокого уровня».

В свою очередь, «Технологии программирования» представляет собой методологическую базу для дисциплин: «Технологии обработки информации», «Интеллектуальные системы и технологии», «Методы и средства проектирования информационных систем и технологий», «Защита информации».

**5. ОБЪЕМ ДИСЦИПЛИНЫ (МОДУЛЯ) В ЗАЧЕТНЫХ ЕДИНИЦАХ С УКАЗАНИЕМ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ИЛИ АСТРОНОМИЧЕСКИХ ЧАСОВ, ВЫДЕЛЕННЫХ НА КОНТАКТНУЮ РАБОТУ ОБУЧАЮЩИХСЯ С ПРЕПОДАВАТЕЛЕМ (ПО ВИДАМ УЧЕБНЫХ ЗАНЯТИЙ) И НА САМОСТОЯТЕЛЬНУЮ РАБОТУ ОБУЧАЮЩИХСЯ**

Общая трудоемкость дисциплины составляет 5 зачетных единиц или 180 часов.

 (из расчета 1 ЗЕТ= 36 часов).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Курс | Семестр | Трудоемкость в ЗЭТ | Общая трудоемкость (час) | Контактная работа | Всего контактных часов | Из них в интерактивных формах | Кол-во часов на СРС | Формаконтроля |
| ЛК | ПР | ЛБ |
| **3** | **5** | **3** | **108** | **32** | **--** | **32** | **64** | **12** | **44****(из них 36 экзамен)** | **экзамен** |

**6. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ, СТРУКТУРИРОВАННОЕ ПО ТЕМАМ (РАЗДЕЛАМ) С УКАЗАНИЕМ ОТВЕДЕННОГО НА НИХ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ИЛИ АСТРОНОМИЧЕСКИХ ЧАСОВ И ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №п/п | Наименованиераздела, темы | Контактная работа | Всего контактных часов | Из них в интерактивной форме | Кол-во часов наСРС |
| ЛК | ПР | ЛБ |
| 1 | **Введение в технологию программирования*** Этапы развития технологий программирования.
* Стихийное программирование.
* Структурное, модульное программирование.
* Объектный подход.
* Компонентный подход.
 | 2 | - | - | 2 | - | 0 |
| 2 | **Введение в анализ алгоритмов*** Анализ алгоритмов.
* Скорости роста, классификация скоростей роста.
* Примеры анализа сложности алгоритмов.
 | 2 | - | - | 2 | - | 1 |
| 3 | **Рекурсивные алгоритмы*** Рекурсия. Рекурсивные выражения. Рекурсивный алгоритм.
* Примеры рекурсивных алгоритмов.
* Построение дерева рекурсивных вызовов.
* Анализ рекурсивных алгоритмов.
* Рекурсия и итерация, анализ сложности.
 | 2 | - | - | 4 | 2 | 1 |
| 4 | **Динамическое программирование*** Постановка задачи. Идея динамического программирования.
* Сравнение рекурсии и динамического программирования.
* Задачи, решаемые методом динамического программирования: задача о полоске, задача о короле, задача о разложении числа.
* Задача о ранце.
* Анализ сложности алгоритмов динамического программирования.
 | 2 | - | - | 6 | 2 | 1 |
| 5 | **Алгоритмы сортировки*** Алгоритм сортировки вставками. Анализ сложности алгоритма.
* Бинарная сортировка. Анализ сложности алгоритма.
* Сортировка выбором. Анализ сложности алгоритма.
* Обменная сортировка. Анализ сложности алгоритма.
* Шейкерная сортировка. Анализ сложности алгоритма.
* Сортировка Шелла. Анализ сложности алгоритма.
* Сортировка элементов массива методом слияния. Анализ сложности алгоритма.
* Алгоритм быстрой сортировки. Анализ сложности алгоритма.
* Внешние сортировки.
* Классификация методов внешней сортировки.
* Сортировки слиянием.
* Многофазная сортировка.
* Каскадная сортировка.
* Оценка сложности алгоритмов.
 | 6 | - | 12 | 18 | 2 | 1 |
| 6 | **Динамические структуры данных*** Линейные структуры данных. Стек, очередь, дек и их реализация при помощи массивов. Основные операции. Примеры.
* Линейные списки, циклические списки. Основные операции со списками. Реализация списков с помощью массивов.
 | 4 | - | - | 4 | - | 1 |
| 7 | **Алгоритмы поиска*** Задачи поиска в структурах данных. Линейный поиск. Оценка сложности алгоритма.
* Обобщенный алгоритм целочисленного бинарного поиска. Оценка сложности алгоритма.
* Вещественный бинарный поиск. Оценка сложности алгоритма.
* Совпадение строк. Постановка задачи. Наивный алгоритм поиска. Оценка сложности.
* Префикс – функция. Алгоритм поиска префикс – функции. Оценка сложности.
* Алгоритм Кнута - Морриса – Пратта (КМП). Оценка сложности.
* Алгоритм Боуэра Мура (БМ).
* Хеширование. Постановка задачи. Хеш-функции. Методы разрешения коллизий.
 | 2 | - | - | 2 | 2 | 1 |
| 8 | **Алгоритмы на деревьях*** Деревья, бинарные деревья.
* Представление бинарных деревьев.
* Прохождение бинарных деревьев.
* Сортировка с прохождением бинарного дерева в симметричном порядке. Сортировка методом турнира с выбыванием.
* Применение бинарных деревьев для сжатия информации.
* Представление сильноветвящихся деревьев.
* Множественная задача точного поиска образцов в тексте: наивный алгоритм, обобщенный КМП.
 | 6 | - | 10 | 16 | 2 | 1 |
| 9 | **Алгоритмы на графах*** Графы. Лемма о рукопожатиии.
* Представление графа в памяти компьютера.
* Поиск в глубину. Оценка сложности алгоритма.
* Топологическая сортировка. Оценка сложности алгоритма.
* Определение наличия циклов в графе. Оценка сложности алгоритма.
* Раскраска графа.
* Остовные деревья.
* Алгоритм Прима.
* Алгоритм Краскала.
* Обход в ширину. Оценка сложности алгоритма.
* Алгоритм Дейкстры.
 | 6 | - | 10 | 16 | 2 | 1 |
|  | Итого: | **32** | - | **32** | **64** | **12** | **8** |
|  | **Экзамен** |  |  |  |  |  | 36 |

**7. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

1. Тоичкин Н.А., Богатиков В.Н., Виноградов Г.П. Технология создания программ. Алгоритмы на графах и деревьях: метод. указания по выполнению лаб. работ и курсового проектирования, (учебное пособие)/ Тверской Государственный технический университет, кафедра ИПМ, 2014. Электронное изд.
2. Тоичкин Н.А., Богатиков В.Н., Виноградов Г.П. Технология создания программ. Алгоритмы поиска: метод. указания по выполнению лаб. работ и курсового проектирования, (учебное пособие) / Тверской Государственный технический университет, кафедра ИПМ, 2014. Электронное изд.
3. Тоичкин Н.А., Богатиков В.Н., Виноградов Г.П. Технология создания программ. Алгоритмы сортировок: метод. указания по выполнению лаб. работ и курсового проектирования, (учебное пособие) / Тверской Государственный технический университет, кафедра ИПМ, 2014. Электронное изд.
4. Электронный образовательный ресурс «Технологии программирования» в системе MOODLE(модульная объектно-ориентированная динамическая учебная среда) МАГУ: <http://moodle.arcticsu.ru/course/view.php?id=54>.

**8. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

**Общие сведения**

|  |  |  |
| --- | --- | --- |
| 1. | Кафедра  | **Информатики, вычислительной техники и информационной безопасности** |
| 2. | Направление подготовки | **09.03.02 «Информационные системы и технологии»**  |
| 3. | Дисциплина (модуль) | **Б1.Б.12 Технологии программирования**  |

**Перечень компетенций**

|  |
| --- |
| * владением широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий (**ОПК-1**)
 |

**Критерии и показатели оценивания компетенций на различных этапах их формирования**

|  |  |  |  |
| --- | --- | --- | --- |
| **Этап формирования компетенции (разделы, темы дисциплины)** | **Формируемая компетенция** | **Критерии и показатели оценивания компетенций** | **Формы контроля сформированности компетенций** |
| **Знать:** | **Уметь:** | **Владеть:** |
| 1. *Введение в технологию программирования*
 | ОПК-1  | этапы развития технологий программирования | разрабатывать блок-схемы алгоритмов | терминологией объектно-ориентированной разработки алгоритмов | Тест, доклад |
| 1. *Введение в анализ алгоритмов*
 | ОПК-1 | понятие анализ алгоритма; примеры расчета анализа сложности алгоритмов | выбирать наилучший алгоритм с точки зрения его производительности | методикой оценки анализа сложности алгоритмов | Тест, доклад, кейс-стади |
| 1. *Рекурсивные алгоритмы*
 | ОПК-1 | понятие рекурсии и рекурсивного алгоритма; Примеры рекурсивных алгоритмов | отличать рекурсивный алгоритм от итерационного; выполнять построение дерева рекурсивных вызовов | методикой расчета сложности рекурсивного алгоритма | Тест, кейс-стади, доклад |
| 1. *Динамическое программирование*
 | ОПК-1 | ключевую идею динамического программирования; отличия в реализации рекурсивных и динамических алгоритмов | выполнять разбиение исходной задачи на последовательность подзадач; выполнять анализ сложности алгоритмов динамического программирования  | разработки алгоритмов динамического программирования;построения рекуррентных формул | Тест, лабораторная работа |
| 1. *Алгоритмы сортировки*
 | ОПК-1 | основные алгоритмы внутренней и внешней сортировки | выбирать алгоритм сортировки в соответствии с поставленной задачей;выполнять анализ алгоритмов сортировки | навыками разработки алгоритмов сортировки в средах программирования | Тест, лабораторная работа |
| 1. *Динамические структуры данных*
 | ОПК-1 | основные принципы организации динамических структур данных | разрабатывать собственные алгоритмы работы с динамическими структурами данных в средах программирования | навыками работы с библиотечными динамическими структурами данных в средах программирования | Тест |
| 1. *Алгоритмы поиска*
 | ОПК-1 | основные алгоритмы поиска в различных структурах данных | выбирать алгоритм поиска в соответствии с поставленной задачей; выполнять анализ алгоритмов поиска | навыками разработки алгоритмов поиска в средах программирования | Тест, лабораторная работа  |
| 1. *Алгоритмы на деревьях*
 | ОПК-1 | основные алгоритмы на деревьях  | выбирать алгоритм работы на деревьях в соответствии с поставленной задачей; выполнять анализ алгоритмов | навыками разработки алгоритмов на деревьях в средах программирования | Тест, лабораторная работа |
| 1. *Алгоритмы на графах*
 | ОПК-1 | основные алгоритмы на графах | выбирать алгоритм работы на графах в соответствии с поставленной задачей; выполнять анализ алгоритмов | навыками разработки алгоритмов на графах в средах программирования | Тест, лабораторная работа |

**Критерии и шкалы оценивания**

1. **Тест**

Кол-во баллов = (Кол-во правильных ответов/Кол-во вопросов)\*3

1. **Критерии оценки выступление студентов с докладом, рефератом, на семинарах**

|  |  |
| --- | --- |
| **Баллы** | **Характеристики ответа студента** |
| 5 | - студент глубоко и всесторонне усвоил проблему;- уверенно, логично, последовательно и грамотно его излагает;- опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью;- умело обосновывает и аргументирует выдвигаемые им идеи;- делает выводы и обобщения;- свободно владеет понятиями  |
| 3 | - студент твердо усвоил тему, грамотно и по существу излагает ее, опираясь на знания основной литературы;- не допускает существенных неточностей;- увязывает усвоенные знания с практической деятельностью;- аргументирует научные положения;- делает выводы и обобщения;- владеет системой основных понятий  |
| 2 |  - тема раскрыта недостаточно четко и полно, то есть студент освоил проблему, по существу излагает ее, опираясь на знания только основной литературы;- допускает несущественные ошибки и неточности;- испытывает затруднения в практическом применении знаний;- слабо аргументирует научные положения;- затрудняется в формулировании выводов и обобщений;- частично владеет системой понятий  |
| **0** | - студент не усвоил значительной части проблемы;- допускает существенные ошибки и неточности при рассмотрении ее;- испытывает трудности в практическом применении знаний;- не может аргументировать научные положения;- не формулирует выводов и обобщений;- не владеет понятийным аппаратом  |

1. **Выполнение лабораторной работы**

10 баллов выставляется, студент выполнил полностью все задания указанные в лабораторной работе и может аргументировано пояснить ход своего решения.

5 баллов выставляется, если студент выполнил не менее 85 % заданий указанных в лабораторной работе, и может аргументировано пояснить ход своего решения и указать.

2 балла выставляется, если студент решил не менее 50% заданий указанных в лабораторной работе, и может аргументировано пояснить ход своего решения.

0 баллов выставляется, если студент не может аргументированно пояснить ход своего решения.

В случае если сроки сдачи работ превышены, количество баллов сокращается на 50%.

1. **Решение кейс - стади**

|  |  |
| --- | --- |
| **Баллы** | **Критерии оценивания** |
| 3 | * изложение материала логично, грамотно, без ошибок;
* свободное владение профессиональной терминологией;
* умение высказывать и обосновать свои суждения;
* студент дает четкий, полный, правильный ответ на теоретические вопросы;
* студент организует связь теории с практикой.
 |
| 2 | * студент грамотно излагает материал; ориентируется в материале, владеет профессиональной терминологией, осознанно применяет теоретические знания для решения кейса, но содержание и форма ответа имеют отдельные неточности;
* ответ правильный, полный, с незначительными неточностями или недостаточно полный.
 |
| 0,5 | * студент излагает материал неполно, непоследовательно, допускает неточности в определении понятий, в применении знаний для решения кейса, не может доказательно обосновать свои суждения;
* обнаруживается недостаточно глубокое понимание изученного материала.
 |
| 0 | * отсутствуют необходимые теоретические знания; допущены ошибки в определении понятий, искажен их смысл, не решен кейс;
* в ответе студента проявляется незнание основного материала учебной программы, допускаются грубые ошибки в изложении, не может применять знания для решения кейса.
 |

***Типовые контрольные задания и методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы***

***1) Типовое тестовое задание***

1. Каков порядок сложности алгоритма бинарного (двоичного) поиска элемента в массиве?

1. *O(n)*
2. *O(n2)*
3. *O(log(n))*
4. *O(2n)*

2. Каков порядок сложности алгоритма линейного поиска элемента в массиве?

1. *O(n)*
2. *O(n2)*
3. *O(log(n))*
4. *O(2n)*

3. Каков порядок сложности «наивного» алгоритма поиска подстроки (длиной m) в строке (длиной n):

1. *O(n\*m)*
2. *O(n-m)*
3. *O(n+m)*
4. *O(n/m)*

4. Для любой строки S, через S[1 ..i], обозначается?

1. *префикс строки S, заканчивающийся в позиции j;*
2. *суффикс строки S, начинающийся в позиции i;*
3. *подстрока строки S, начинающаяся в позиции i и заканчивающаяся в позиции j;*
4. *длина строки S.*

5. Суффикс строки S называется собственным если?

1. *он не пустой и совпадает с префиксом;*
2. *он не пустой и не совпадает с S;*
3. *он не пустой и совпадает с S;*
4. *он не пустой.*

6. Рассматриваем алгоритм вычисления префикс – функции для символов строки S. Пусть мы построили решение для первых i символов. Чему равно Пi+1(S), если i+1 символ строки S совпадает с символом, стоящим за префиксом Пi(S)?

1. *Пi+1(S) = Пi(S);*
2. *Пi+1(S) = Пi(S) + 1;*
3. *Пi+1(S) = Пi(S) -1.*

7. В каком из перечисленных методов внутренней сортировки выгодно применяется метод установки «барьера»?

1. *Сортировка выбором.*
2. *Сортировка «пузырьком».*
3. *Сортировка вставками.*
4. *Сортировка слиянием.*

8. В каком из перечисленных методов внутренней сортировки используется тот факт, что при обработке некоторого элемента a[i] элементы a[1], a[2], … ,a[i -1], уже могут быть упорядоченными?

1. *Сортировка Шелла.*
2. *Сортировка «пузырьком».*
3. *Сортировка вставками.*
4. *Бинарная сортировка.*

9. Многопроходная внутренняя сортировка, при которой исходный массив разбивается на части, каждая из которых сортируется отдельно, причем на каждом проходе число частей уменьшается, называется?

1. *Сортировкой Шелла.*
2. *Сортировкой «пузырьком».*
3. *Сортировкой вставками.*
4. *Бинарной сортировка.*

10. Представлен фрагмент псевдокода для некоторого алгоритма внутренней сортировки. Какой оператор закрыт рамкой?



1. *a[j + 1] : = x;*
2. *a[j – 1] : = x;*
3. *a[ j ] : = x;*
4. *нет оператора.*

**Ключ:** 1-3, 2-1, 3-1, 4-1, 5-2, 6-2, 7-3, 8-4, 9-1, 10 - 1

***2) Примерные темы докладов***

1. Процессы жизненного цикла программного продукта по стандарту ISO/IEC 12207.
2. Проектирование интерфейса пользователя. Модели и технологии.
3. Организация процесса тестирования программного обеспечения.
4. Функциональное тестирование программного обеспечения или тестирование «черного ящика».
5. Структурное тестирование программного обеспечения или тестирование «белого ящика».
6. Модели надежности программного обеспечения.
7. Обзор современных сред разработки программного обеспечения.
8. Общие принципы разработки распределенных систем.
9. Введение в унифицированный процесс разработки программных приложений (RUP).
10. Применение case-средств проектирования в процессе разработки программных приложений.
11. Общие принципы компонентно-ориентированной разработки.
12. Аспектно и атрибутно – ориентированная разработка программных приложений.
13. Платформа .NET.
14. Общие принципы и технологии разработки приложений для систем реального времени.
15. Общие принципы разработки приложений для встраиваемых систем.
16. Общие принципы разработки корпоративных приложений.
17. Использование паттернов проектирования в процессе разработки программных приложений.
18. Процессы жизненного цикла программного продукта.

***3) Пример задания на лабораторную работу***

*Тема:* Разработать программу вычислительного эксперимента для исследования характеристик эффективности сортировки одним из методов (согласно варианту). Проверить совпадение модельных представлений и результатов вычислительного эксперимента.Предусмотреть вывод исходной и результирующей информации в файл.

*Требования:*

Разработать программу в среде Visual Studio или Borland Developer Studio, приложение типа Windows Application, на языке программирования или C# или С++.

В программе реализовать:

* реализовать в программе один из методов сортировки массивов согласно варианту задания в виде отдельной функции;
* использовать динамические массивы;
* вычислительный эксперимент выполнить для нескольких массивов, число которых устанавливает пользователь;
* число элементов в массиве устанавливает пользователь, массив создается с помощью генератора случайных чисел;
* для каждого эксперимента привести теоретическое и практическое (полученное в результате эксперимента) число сравнений и перестановок;
* результаты эксперимента записать в файл;
* обработать исключительные ситуации;
* предусмотреть диалог с пользователем;
* отобразить графически на диаграмме полученные результаты эксперимента для каждого массива.

***4). Пример кейс - стади***

**Пример 1.** В некотором городе есть N городов. Требуется соединить их телефонной сетью, минимизируя при этом расходы на провода. В каждом городе расположена всего одна телефонная стация; от нее может отходить любое количество проводов, ведущих в другие города. Телефонное соединение между городами не обязательно должно быть прямым: если город А связан с городом В, а город В – с городом С, можно смело полагать, что А связан с С. Расстояние между любыми двумя городами (соответственно и требуемая длина провода) считается известным.

*Вопросы*

1. В каком виде лучше всего описывается данная задача?
2. Как можно представить данную задачу в графическом виде?
3. Какой алгоритм на ваш взгляд подходит для решения данной задачи?
4. Приведите алгоритм решения задачи.

**Пример 2.** Дается карта автодорог некоторой страны. Требуется найти кратчайший путь из однного города в другой. Вес пути из города А в В, может быть не равен пути из В в А (в гору ехать больше чем с горы). Необходимо найти маршрут из вершины помеченной как стартовая, в вершину, помеченную как финишная; при этом сумма весов путей, входящих в маршрут, должна быть минимальной.

*Вопросы*

1. В каком виде лучше всего описывается данная задача?
2. Как можно представить данную задачу в графическом виде?
3. Какой алгоритм на ваш взгляд подходит для решения данной задачи?
4. Приведите алгоритм решения задачи.

**Вопросы к экзамену**

1. Стихийное программирование.
2. Структурное программирования. Модульное программирование.
3. Объектно-ориентированное программирование.
4. Компонентно-ориентированное программирование.
5. Анализ алгоритмов. Скорости роста, классификация скоростей роста. Пример анализа сложности алгоритмов.
6. Рекурсивные алгоритмы. Определение рекурсии. Схемы прямой (простой) и косвенной (сложной) рекурсии. Примеры рекурсивных алгоритмов. Пример: «Ханойская башня».
7. Реализация механизма рекурсивного вызова процедуры.
8. Постановка задачи. Идея динамического программирования. Сравнение рекурсии и динамического программирования.
9. Задачи, решаемые методом динамического программирования: задача о полоске.
10. Задачи, решаемые методом динамического программирования: задача о короле.
11. Задачи, решаемые методом динамического программирования: задача о разложении числа.
12. Задачи, решаемые методом динамического программирования: задача о ранце.
13. Алгоритм сортировки вставками. Анализ сложности алгоритма.
14. Бинарная сортировка. Анализ сложности алгоритма.
15. Сортировка выбором. Анализ сложности алгоритма.
16. Обменная сортировка. Анализ сложности алгоритма.
17. Шейкерная сортировка. Анализ сложности алгоритма.
18. Сортировка Шелла. Анализ сложности алгоритма.
19. Сортировка элементов массива методом слияния. Анализ сложности алгоритма.
20. Алгоритм быстрой сортировки. Анализ сложности алгоритма.
21. Пирамидальная сортировка. Анализ сложности алгоритма.
22. Внешние сортировки. Классификация методов внешней сортировки. Простое слияние.
23. Внешние сортировки. Естественное слияние.
24. Внешние сортировки. Получение начальных серий для многопутевого слияния посредством выбора с замещением.
25. Многофазная сортировка.
26. Каскадная сортировка.
27. Совпадение строк. Постановка задачи. Наивный алгоритм поиска. Оценка сложности. Ускорение наивного алгоритма. Оценка сложности.
28. Префикс – функция. Алгоритм поиска префикс – функции. Оценка сложности.
29. Алгоритм Кнута - Морриса – Пратта (КМП). Оценка сложности.
30. Хеширование. Постановка задачи. Хеш-функции. Методы разрешения коллизий.
31. Алгоритмы вставки поиска и удаления для хеш-таблицы, имеющей три состояния элементов. Переполнение таблицы и рехеширование. Оценка качества хеш-функций.
32. Линейные структуры данных. Стек, очередь, дек.
33. Линейные списки, циклические списки. Основные операции со списками.
34. Деревья, бинарные деревья. Представление бинарных деревьев. Прохождение бинарных деревьев.
35. Сортировка с прохождением бинарного дерева в симметричном порядке. Сортировка методом турнира с выбыванием.
36. Применение бинарных деревьев для сжатия информации.
37. Представление сильноветвящихся деревьев. Множественная задача точного поиска образцов в тексте: наивный алгоритм, обобщенный КМП.
38. Представление графов в памяти компьютера.
39. Поиск в глубину. Оценка сложности алгоритма.
40. Топологическая сортировка. Оценка сложности алгоритма.
41. Определение наличия циклов в графе. Оценка сложности алгоритма.
42. Обход в ширину. Оценка сложности алгоритма.
43. Ракраска графа.
44. Алгоритм Прима.
45. Алгоритм Краскала.
46. Алгоритм Дейкстры.

**9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)**

*а) основная литература:*

1. [Костюкова Н. И.](https://biblioclub.ru/index.php?page=author_red&id=81220) Комбинаторные алгоритмы для программистов. М.: [Национальный Открытый Университет «ИНТУИТ»](https://biblioclub.ru/index.php?page=publisher_red&pub_id=16364), 2016, 217 с.
2. [Седжвик Р.](https://biblioclub.ru/index.php?page=author_red&id=152112) Алгоритмы на С++. М.: [Национальный Открытый Университет «ИНТУИТ»](https://biblioclub.ru/index.php?page=publisher_red&pub_id=16364)
3. Головин И.Г. Языки и методы программирования.-М.:Академия, 2012

*б) дополнительная литература:*

1. [Громов Ю. Ю.](https://biblioclub.ru/index.php?page=author_red&id=106639) , [Иванова О. Г.](https://biblioclub.ru/index.php?page=author_red&id=106640) , [Беляев М. П.](https://biblioclub.ru/index.php?page=author_red&id=106887) ,[Минин Ю. В.](https://biblioclub.ru/index.php?page=author_red&id=106858) Технология программирования. Тамбов: [Издательство ФГБОУ ВПО «ТГТУ»](https://biblioclub.ru/index.php?page=publisher_red&pub_id=17574), 2013. 173 с.
2. [Терехов А. Н.](https://biblioclub.ru/index.php?page=author_red&id=81245) Технология программирования. М.: [Интернет-Университет Информационных Технологий](https://biblioclub.ru/index.php?page=publisher_red&pub_id=16091), 2007, 2-е изд.149 с.
3. [Алексеев В. Е.](https://biblioclub.ru/index.php?page=author_red&id=151866) , [Таланов В. А.](https://biblioclub.ru/index.php?page=author_red&id=151867) Структуры данных. Модели вычислений. М.: [Национальный Открытый Университет «ИНТУИТ»](https://biblioclub.ru/index.php?page=publisher_red&pub_id=16364), 2016, 248 c.

**10. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ "ИНТЕРНЕТ" (ДАЛЕЕ - СЕТЬ "ИНТЕРНЕТ"), НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)**

* 1. <http://www.intuit.ru> – национальный открытый университет.
	2. <http://citforum.ru> – справочная информация по различным разделам информационных технологий.

**11. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Приступая к изучению дисциплины, студенту необходимо внимательно ознакомиться с тематическим планом занятий, списком рекомендованной литературы. Следует уяснить последовательность выполнения индивидуальных учебных заданий. Самостоятельная работа студента предполагает работу с научной и учебной литературой. Уровень и глубина усвоения дисциплины зависят от активной и систематической работы на лекциях, изучения рекомендованной литературы, выполнения контрольных письменных заданий.

Студентам рекомендуется регулярно посещать лекции, тщательно конспектировать и прорабатывать их с одним из рекомендованных литературных источников.

При изучении курса студентам рекомендуется проводить:

* конспектирование первоисточников и другой учебной литературы;
* проработку учебного материала (по конспектам, учебной и научной литературе);
* подготовку к коллоквиумам с использованием электронных методических указаний по курсу;
* самотестирование по контрольным вопросам (тестам).

Каждый студент перед началом занятий записывается преподавателем на электронный курс по данному предмету, к которому можно получить доступ через сеть Интернет. Курс поддерживается системой дистанционного обучения *moodle* (модульная объектно-ориентированная динамическая учебная среда), к которой может получить доступ любой зарегистрированный пользователь через сеть Интернет. Адрес курса в системе *moodle* МАГУ: <http://moodle.arcticsu.ru/course/view.php?id=54> [[1]](#footnote-1).

В рамках данного курса в системе *moodle*, организовано:

* взаимодействие обучающихся между собой и с преподавателем: для чего используются форумы и чаты.
* передача знаний в электронном виде: с помощью файлов, архивов, веб-страниц, лекций.
* проверка знаний и обучение с помощью тестов и заданий:  результаты работы студенты могут отправлять в текстовом виде или в виде файлов.
* совместная учебная и исследовательская работа студентов по определенной теме: с помощью встроенных механизмов: семинаров, форумов и пр.
* журнал оценок: в котором учитывается успеваемость студентов по балльной системе.

Таким образом, вся самостоятельная работа студента организуется через систему дистанционного обучения *moodle* МАГУ. Так же данная система используется преподавателем и в процессе проведения аудиторных занятий, для: проведения тестов, предоставления текстов лекций и лабораторных работ, учета успеваемости учащихся.

Основными видами аудиторной работы студентов являются лекции и лабораторные работы.

В ходе лекций преподаватель излагает и разъясняет основные, наиболее сложные понятия темы, а также связанные с ней теоретические и практические проблемы, дает рекомендации на лабораторные работы и указания на самостоятельную работу. Электронные конспекты презентаций и полных текстов лекций доступны для студента в системе *moodle* МАГУ на сайте курса: «Технологии программирования».

На практических занятиях студенты совместно с преподавателем обсуждают выданные им задания, задают интересующие их вопросы и выполняют на компьютерах самостоятельно или в группах свои проекты, используя программное обеспечение представленной в рабочей программе. Каждую выполненное задание студент обязан оформить в виде отчета и защитить его. Электронные конспекты лабораторных заданий доступны для студента в системе *moodle* МАГУ на сайте курса: «Технологии программирования».

Качество учебной работы студентов преподаватель оценивает с использованием технологической карты дисциплины, размещенной на сайте МАГУ.

**Планы лабораторных работ:**

**Лабораторная работа 1. Алгоритмы внутренней сортировки.**

**План:**

1. Основные определения.
2. Постановка задачи сортировки.
3. Внутренние и внешние сортировки.
4. Сложность алгоритмов сортировки.
5. Сортировка вставками. Анализ алгоритма сортировки вставками.
6. Бинарная сортировка. Анализ алгоритма бинарной сортировки.
7. Сортировка Шелла. Анализ алгоритма.
8. Пузырьковая сортировка. Анализ алгоритма.
9. Шейкерная сортировка. Анализ алгоритма.

*Литература*: [1, c. 147-157; 2, с. 373-419].

*Вопросы для самоконтроля*

* 1. Выполните постановку задачи сортировки.
	2. В чем отличие методов внешней и внутренней сортировки?
	3. Как выполняется алгоритм сортировки вставками?
	4. Как выполняется алгоритм бинарной сортировки?
	5. Как выполняется алгоритм сортировки Шелла?
	6. Как выполняется алгоритм пузырьковой сортировки?
	7. Как выполняется алгоритм шейкерной сортировки?
	8. Какой из рассматриваемых алгоритмов сортировки имеет наименьшую сложность?

*Задание для самостоятельной работы*

1. Изучить предложенные алгоритмы внутренней сортировки.
2. Разработать программу вычислительного эксперимента для исследования характеристик эффективности сортировки одним из методов (согласно варианту).
3. Проверить совпадение модельных представлений и результатов вычислительного эксперимента. Предусмотреть вывод исходной и результирующей информации в файл.
4. Протестировать работу программы.
5. Подготовиться к ответам на вопросы.

**Лабораторная работа 2. Алгоритмы внешней сортировки.**

**План:**

1. Необходимость методов внешней сортировки.
2. Процедура слияния и сортировка слиянием.
3. Анализ алгоритмов внешней сортировки.
4. Характеристики алгоритмов внешней сортировки.
5. Естественное слияние.
6. Естественное сбалансированное слияние.
7. Многофазная сортировка.
8. Каскадная сортировка.

*Литература*: [1, c. 158-169; 2, с. 496-533].

*Вопросы для самоконтроля*

1. В чем отличие методов внешней и внутренней сортировки?
2. Перечислите характеристики алгоритмов внешней сортировки.
3. Что означает процедура слияния?
4. Что означает естественное сбалансированное слияние?
5. Как выполняется алгоритм внешней сортировки методом простого слияния?
6. В чем отличие метода естественного слияния от простого слияния?
7. В чем состоит идея метода многофазной и каскадной сортировок?
8. В чем заключается алгоритм распределения пустых серий в методе многофазной сортировки?

*Задание для самостоятельной работы*

* 1. Разработать программу, реализующую один из методов внешней сортировки (согласно варианту задания). Предусмотреть вывод результирующей информации в файл.
	2. Протестировать работу программы.
	3. Подготовиться к ответам на вопросы.

**Лабораторная работа 3. Алгоритмы на деревьях.**

**План:**

1. Определение дерева. Основные определения.
2. Бинарные и сильноветвящиеся деревья.
3. Представление деревьев.
4. Прохождения деревьев.
5. Пирамидальная сортировка.
6. Применение деревьев для сжатия данных.
7. Деревья поиска. Определение Trie дерева.
8. Вставка, поиск и удаление элемента в/из Trie дерево(а).
9. Реализация Trie дерева.

*Литература*: [1, c. 32-42; 2, с. 942-956].

*Вопросы для самоконтроля*

1. Дайте определение дерева, как программной и алгоритмической конструкции.
2. Каким образом реализуются в программах бинарные и сильноветвящиеся деревья?
3. В чем отличие бинарных от сильноветвящихся деревьев?
4. Какие методы прохождения бинарных деревьев вы знаете, и как они выполняются?
5. Дайте определение пирамиды. В чем состоит идея пирамидальной сортировки?
6. Что такое префиксное дерево?
7. Как происходит вставка, удаление и поиск элемента в префиксном (Trie) дереве?
8. Приведите пример алгоритма сжатия данных, с помощью бинарного дерева.

*Задание для самостоятельной работы*

1. Разработать программу, реализующую один из алгоритмов обработки бинарного или сильноветвящегося дерева (согласно варианту задания). Предусмотреть вывод результирующей информации в файл.
2. Протестировать работу программы.
3. Подготовиться к ответам на вопросы.

**Лабораторная работа 4. Алгоритмы на графах.**

**План:**

1. Определение графа. Основные определения.
2. Представление графов.
3. Базовые алгоритмы на графах.
4. Поиск цикла в графе.
5. Поиск в глубину.
6. Поиск в ширину.
7. Топологическая сортировка.
8. Алгоритм Дейкстры.
9. Алгоритмы построения лабиринтов.

*Литература*: [1, c. 117-129, с. 170-182; 2, с. 1055-1590].

*Вопросы для самоконтроля*

1. Дайте определение графа, как программной и алгоритмической конструкции.
2. Каким образом реализуются в программах графы?
3. В чем отличие графа от деревьев?
4. Какие графы называют циклическими?
5. В чем состоит идея поиска в глубину на графе? Какая структура данных применяется в алгоритме поиска в глубину?
6. В чем состоит идея поиска в ширину на графе? Какая структура данных применяется в алгоритме поиска в ширину?
7. В чем заключается алгоритм Дейкстры?
8. Что такое топологическая сортировка?

*Задание для самостоятельной работы*

1. Разработать программу, реализующую один из алгоритмов обработки графа (согласно варианту задания). Предусмотреть вывод результирующей информации в файл.
2. Протестировать работу программы.

**12. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ (МОДУЛЮ), ВКЛЮЧАЯ ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ (ПРИ НЕОБХОДИМОСТИ)**

*Программное обеспечение:*

1. Microsoft Windows XP – операционная система.
2. Microsoft Word или Open Office Writer – текстовый редактор.
3. Microsoft Visual Studio – средство разработки программных приложений.
4. Любой web – браузер.

**Тренажеры:** В целях обучения студентов, усвоения и контроля полученных знаний используется электронный образовательный ресурс moodle: (<http://moodle.arcticsu.ru/>) в котором студенты могут проходить тесты. Адрес курса: <http://moodle.arcticsu.ru/course/view.php?id=54>

**13. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

|  |  |  |
| --- | --- | --- |
| № п\п | Наименование оборудованных учебных кабинетов, объектов для проведения занятий с перечнем основного оборудования | Фактический адрес учебных кабинетов и объектов, номер ауд. |
| 1 | Кабинет для проведения практических занятий по курсу с комплектом мультимедийного оборудования, включающий ноутбук, мультимедиа-проектор и экран  |  г. Апатиты, ул. Энергетическая, 19, корпус 5, ауд. 4 |
| 2 |  Компьютерный класс для проведения лабораторных занятий и тестирования.  | г. Апатиты, ул. Энергетическая, 19, корпус 5, ауд. 4 |

**14. ТЕХНОЛОГИЧЕСКАЯ КАРТА ДИСЦИПЛИНЫ**

**ОСНОВНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА**

**09.03.02 «Информационные системы и технологии» профиль Общий профиль**

(код, направление, профиль)

**ТЕХНОЛОГИЧЕСКАЯ КАРТА**

|  |  |
| --- | --- |
| Шифр дисциплины по РУП | **Б1.Б.12** |
| Дисциплина | **Технологии программирования**  |
| Курс  | **3** | семестр | **5** |
| Кафедра | **Информатики, вычислительной техники и информационной безопасности** |
| Ф.И.О. преподавателя, звание, должность | **Тоичкин Николай Александрович, к.т.н., доцент** |
|  |
| Общ. трудоемкостьчас/ЗЕТ | **108/3** | Кол-во семестров | **1** | Интерактивные формыобщ./тек. сем. | **12/12** |
| ЛКобщ./тек. сем. | **32/32** | ПР/СМобщ./тек. сем. | **--/--** | ЛБобщ./тек. сем. | **32/32** | Форма контроля | **Экзамен** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание задания** | **Количество мероприятий** | **Максимальное количество баллов** | **Срок предоставления** |
| ***Вводный блок*** |
| Не предусмотрен |  |  |  |
| ***Основной блок*** |
| Решение бланочных тестов  | 4 | 12 | 5 неделя7 неделя11 неделя16 неделя |
| Лабораторные работы | 4 | 40 | 4 неделя8 неделя12 неделя16 неделя |
| Подготовка докладов по теме | 1 | 5 | По согласованию с преподавателем |
| Решение кейс-стади | 1 | 3 | 7 неделя14 неделя |
|  |  |  |  |
| **Итого:** | **60** |  |
| Экзамен | Вопрос 1 | 20 | В сроки сессии |
|  | Вопрос 2 | 20 | В сроки сессии |
| **Всего:** | **40** |  |
| **Итого:** | **100** |  |
| ***Дополнительный блок*** |
| Выполнение дополнительной лабораторной работы | 10 | по согласованию с преподавателем |
| Подготовка глоссария | 5 |
| **Всего баллов по дополнительному блоку:** | **15** |

Шкала оценивая в рамках балльно-рейтинговой системы МАГУ: «2» - 60 баллов и менее, «3» - 61-80 баллов, «4» - 81-90 баллов, «5» - 91-100 баллов.

**15. ИНЫЕ СВЕДЕНИЯ И МАТЕРИАЛЫ НА УСМОТРЕНИЕ ВЕДУЩЕЙ КАФЕДРЫ**

Не предусмотрено.

**16. ОБЕСПЕЧЕНИЕ ОБРАЗОВАНИЯ ДЛЯ ЛИЦ С ОВЗ**

Для обеспечения образования инвалидов и лиц с ограниченными возможностями здоровья реализация дисциплины *Б1.Б.12 «Технологии программирования»* может осуществляться в адаптированном виде, с учетом специфики освоения и дидактических требований, исходя из индивидуальных возможностей и по личному заявлению обучающегося.

1. Ссылка активна на момент написания рабочей программы (24.09.16). Для получения доступа к курсу необходима регистрация в системе и запись на курс. [↑](#footnote-ref-1)