

1. Титульный лист (скан-копия)

2. Технологическая карта дисциплины «Информатика»

2.1. Общие сведения о дисциплине.

Название дисциплины – «Информатика»

Факультет, на котором преподается данная дисциплина – математический

Направление подготовки – Информационные системы и технологии

Квалификация (степень) выпускника – бакалавр

Цикл дисциплин – естественно-научный

Часть цикла – базовая

Курс – 1

Семестры – 1

Всего зачетных единиц – 5

Всего часов – 180

Аудиторные занятия 90 часов (из них лекции – 36 часов, лабораторные занятия – 36 часов, практические занятия – 18 часов)

Самостоятельная работа – 54 часа

Экзамен – 1 семестр

Зачет – нет

2.2. Цели и задачи освоения дисциплины

Целями освоения дисциплины «Информатика» являются подготовка в области языков программирования, алгоритмов, библиотек и пакетов программ.

Задачи освоения дисциплины включают:

- Получение студентами теоретических знаний в области алгоритмизации и программирования на языках высокого уровня
- Получение практических навыков программирования

2.3. Место дисциплины в структуре ООП

Для изучения дисциплины требуются предварительные знания на уровне профильного курса «Основы информатики и ИКТ» средней школы. Знания и умения, приобретенные студентами в результате изучения дисциплины, будут использоваться при изучении курсов «Структуры данных на C++», «Комбинаторные алгоритмы», «Системное программирование», «Компьютерные сети», «Технология разработки ПО», при выполнении курсовых и дипломных работ, связанных с математическим моделированием и обработкой данных.

2.4. Требования к результатам освоения дисциплины:

2.4.1. Компетенции обучающегося, формируемые в результате освоения дисциплины:

В результате освоения дисциплины частично формируются следующие компетенции::

- способность разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные) (ПК-12);
- готовность осуществлять организацию контроля качества входной информации (ПК-22);
- способность проводить сбор, анализ научно-технической информации, отечественного и зарубежного опыта по тематике исследования (ПК-23);
- способность участвовать в постановке и проведении экспериментальных исследований (ПК-24);
- способность обосновывать правильность выбранной модели, сопоставляя результаты экспериментальных данных и полученных решений (ПК-25);
- готовность использовать математические методы обработки, анализа и синтеза результатов профессиональных исследований (ПК-26);

- готовность адаптировать приложения к изменяющимся условиям функционирования (ПК-34);
- способность составления инструкций по эксплуатации информационных систем (ПК-35).

2.4.2. Результаты освоения дисциплины.

Обучающийся должен:

- знать: синтаксис языка программирования Си и вызовы стандартной библиотеки в рамках стандарта ISO/IEC 9899, методы и технологии программирования, базовые алгоритмы обработки данных.
- уметь: разрабатывать и реализовывать алгоритмы на языке программирования высокого уровня Си, описывать основные структуры данных, работать в средах программирования;
- владеть: методами и технологиями разработки алгоритмов, описания структур данных, программирования на языке высокого уровня, работы в различных средах программирования.

3. Рабочая программа дисциплины (содержание и структура)

3.1. Содержание дисциплины

Таблица 1.

№ раздела	Наименование раздела	Содержание раздела
1.	Введение в программирование на языке высокого уровня Си	Процесс разработки и инструментальные средства. Структура программы на языке Си. Элементарные средства программирования. Базовые алгоритмы. Элементы профессионального стиля программирования.
2.	Базовое содержание: язык и стандартная библиотека.	Синтаксические конструкции языка Си и средства стандартной библиотеки.
3.	Элементы профессионального программирования	Вопросы низкоуровневого программирования. Конструирование абстрактных типов данных. Раздельная трансляция и компоновка программ. Средства современных редакций стандарта языка Си.

3.2. Структура дисциплины (темы, часы/зачетные единицы):

Таблица 2.

№ раздела	Наименование раздела	Количество часов / зачетных единиц				
		Аудиторная работа		Всего	В т.ч. дистанц.	Внеаудиторная работа
		Лекции	Лаб и практ. занятия			
1	Введение в программирование на языке высокого	6	16	22		12

	уровня Си					
2	Базовое содержание: язык и стандартная библиотека.	18	24	42		26
3	Элементы профессионального программирования	12	14	36		16
	Итого (час)	36	54	90		90
	Итого (зач. единиц)	1	1.5	2.5		2.5

3.3. Содержание лекций в соответствии с тематическим планом в электронном виде (тема и расширенное содержание).

Таблица 3.

№ раз-дела	Тема лекции	Форма контроля	Количество часов/ зачетных единиц
1	Введение в программирование на ЯВУ Си. Структура программы, тип <code>int</code> , функция <code>main</code> , присваивания, базовые арифметические операции, ввод-вывод. Сборка программы вручную и с помощью <code>make</code> . Обзор инструментов: компилятор, отладчик, текстовый редактор, система сборки.	Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2
1	Элементы структурного программирования и их реализация на языке Си. Базовые типы данных, присваивание, операции. Инструкции потока управления (организация ветвлений, циклов, функций). Использование макросов и функций стандартной библиотеки. Примеры. Массивы в языке Си. Индексирование. Идиомы типовых задач обработки массивов. Строка как нуль-терминальный массив символов. Идиомы типовых задач обработки строк.	Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2
1	Элементы профессионального стиля программирования: стандарты и рекомендации. Подходы к разработке программ. Шаблоны программ, кода. Проектирование сверху-вниз и снизу-вверх. Структуризация программы на уровне блоков, функций, файлов исходного кода. Примеры нечитаемого кода. Проблема	Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2

	сопровождения. Вопросы стиля программирования на примере системы соглашений, принятых в курсе «Информатика». Форматирование кода. Комментирование. Контроль возвращаемых значений функций. Приемы программирования.		
2	Объекты данных в программе на языке Си. Простые типы данных (целочисленные, вещественнозначные). Модификаторы unsigned, short/long. Вопросы представления данных. Типы-перечисления. Операция sizeof. Препроцессорные и непосредственные константы, суффиксы типов. Переменные: определение и инициализация. Размеры целочисленных типов (limits.h) и характеристики типов с плавающей точкой (float.h), стандартные предопределенные типы size_t, ssize_t и другие стандартов C99 и POSIX (stddef.h).	Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2
2	Операции и приоритеты. Арифметические и логические операции. Операции отношения и логические связи. Присваивания, тернарная операция. Система приоритетов. Приведение типов. Правила преобразования типов в присваиваниях. Вспомогательные макросы для целочисленных типов (stdint.h) и форматные преобразования (inttypes.h).	Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2
2	Адреса и указатели. Использование динамически распределяемой памяти. Операция взятия адреса и разыменование указателя. Адресная арифметика. Значение NULL. Предопределенный тип ptrdiff_t. Связь указателей и массивов. Распределение многомерных массивов в памяти. Массивы динамической памяти. Выделение, использование и ограничения, утечки памяти	Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2
2	Потоковый ввод-вывод (stdio.h). Понятие потока, стандартные потоки stdin, stdout, stderr. Форматный Ю, перенаправление потоков. Проблема ввода строки: scanf, fgets и решение Хэзфилда. Файловый ввод-вывод с	Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2

	использованием потоков, позиционирование в потоке. Блочный ввод-вывод. Аргументы командной строки (параметры функции main). Функции обработки символов и строк. Функции классификации и преобразования символов (ctype.h). Указатель char*. Представление и использование нультерминальных строк. Обзор типовых ошибок. Использование функций работы со строками стандартной библиотеки (string.h).		
2	Функции в языке Си. Определение. Передача параметров по значению и по ссылке. Указатели в параметрах функции. Использование квалификатора const для защиты аргументов. Квалификатор static в параметрах функции. Вызов функции и возвращаемое значение. Стек вызовов. Рекурсивный вызов функций. Создание и использование указателей на функции. Передача функции в качестве параметра. Функции быстрой сортировки и бинарного поиска стандартной библиотеки (stdlib.h). Использование typedef с указателями на функции.	Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2
2	Структурный тип. Структуры, прямой и косвенный доступ к элементам, передача в качестве параметра, структуры как параметры функций и как результат. Использование typedef с указателями на структуры. Вложенные структуры, массивы структур и массивы как поля структуры. Инициализация компонентов. Указатели в структурах. Динамически определяемые поля.	Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2
2	Связное представление данных. Указатели на структуры в полях структур на примере организации связанного списка.	Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2
2	Генерирование случайных чисел. Функции работы с временем и датой. Вспомогательные средства стандартной библиотеки (stdlib.h). Числовые преобразования. Генерирование случайных чисел в заданном промежутке: наивный подход, решение Александреску. Взаимодействие с операционной	Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2

	<p>средой: стек финальных функций, доступ к переменным среды. Функции работы с временем и датой. Временная отметка и структура времени tm. Функции преобразования и форматирования даты/времени.</p>		
2	<p>Средства математической библиотеки. Функции с переменным числом аргументов.</p> <p>Поддержка математики с плавающей точкой (math.h). Стандарт IEEE, типы, макросы классификации и сравнения. Функции математической библиотеки. Комплексные типы и арифметика (complex.h). Обобщенные безсуффиксные макросы (tgmath.h). Управление математической подсистемой (fenv.h): исключениями, режимами округления. Создание и использование функций с переменным числом параметров (stdarg.h). Семейство функций v.printf / v.scanf.</p>	<p>Выполнение заданий с использованием ЭВМ в рамках лабораторных работ</p>	2
3	<p>Препроцессор языка C. Механизмы раздельной трансляции.</p> <p>Препроцессорные средства: замены, включения, условная компиляция, стандартные макроопределения. Определение макросов вне программы. Разработка крупных программ. Структуризация программ с помощью модулей. Раздельная компиляция. Классы памяти. Блоки и области видимости. Объявления внешних объектов и заголовочные файлы. Компоновка объектных модулей. Алгоритм поиска зависимостей. Создание и использование статических и динамических библиотек. Компоновка во время загрузки и выполнения программы. Подсистема времени выполнения C.</p>	<p>Выполнение заданий с использованием ЭВМ в рамках лабораторных работ</p>	2
3	<p>Приемы низкоуровневого программирования. Диагностика ошибок.</p> <p>Практика использования битовых операций. Работа с битовыми полями структур. Использование объединений для создания множественных представлений данных. Квалификатор volatile. Диагностика ошибок средствами стандартной библиотеки (assert.h, errno.h), управление сигналами (signal.h). Нелокальные переходы (setjmp.h)</p>	<p>Выполнение заданий с использованием ЭВМ в рамках лабораторных работ</p>	2
3	<p>Конструирование абстрактных</p>	<p>Выполнение</p>	2

	типов данных. Примеры реализации абстрактных типов данных. Инкапсуляция и неполные типы. Вопросы проектирования АТД.	заданий с использованием ЭВМ в рамках лабораторных работ	
3	Локализация и интернационализация. Возможности интернационализации. Установка локали. Мультибайтовые и широкие символы. Стандарт Unicode/UCS и кодировки Unicode. Функции преобразования мультибайтовых и широких символов и строк. Универсальные имена символов. Вспомогательные функции управления мультибайтовыми символами и строками: установка направления потока, ввод-вывод, утилиты.	Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2
3	Обзор возможностей стандартов C99 и C11. inline-функции, ограниченные указатели, гибкие элементы массивов, массивы переменной длины, булевские типы и значения, многопоточное программирование	Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2
3	Лекция приглашенного профессора. Лекция представителя индустрии в рамках требований ФГОС о выделении не менее 5% лекционных занятий для проведения приглашенными специалистами. (В 2012-2013 уч. г. лекция профессора университета Тампере, председателя Программы открытых инноваций FRUCT С.И. Баландина).	Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2

3.4. Практические занятия

Таблица 4.

№ раздела	Тема занятия	Форма контроля	Количество часов/ зачетных единиц
1	Использование инструментальных средств разработки на языке Си. Чтение диагностических сообщений. Подготовка файла сборки Makefile	Самостоятельное решение задач в рамках занятия. Выполнение заданий с использованием ЭВМ в рамках	2

		лабораторных работ	
1	Базовые алгоритмы. Циклические алгоритмы: поиск min/max, алгоритмы с флажком.	Самостоятельное решение задач в рамках занятия. Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2
1	Базовые алгоритмы. Алгоритмы обработки одномерных числовых последовательностей.	Самостоятельное решение задач в рамках занятия. Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2
1	Базовые алгоритмы. Алгоритмы обработки двумерных массивов.	Самостоятельное решение задач в рамках занятия. Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2
1	Базовые алгоритмы. Алгоритмы обработки строк символов.	Самостоятельное решение задач в рамках занятия. Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2
2	Адресная арифметика. Решение задач с использованием указателей.	Самостоятельное решение задач в рамках занятия. Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2
2	Использование функций потокового ввода-вывода стандартной библиотеки языка Си.	Самостоятельное решение задач в рамках занятия. Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2

2	Структурный тип данных. Конструирование линейных списков.	Самостоятельное решение задач в рамках занятия. Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2
3	Проектирование и реализация абстрактных типов данных	Самостоятельное решение задач в рамках занятия. Выполнение заданий с использованием ЭВМ в рамках лабораторных работ	2

3.5. Лабораторные занятия (перечень лабораторных работ, график выполнения работ)

Перечень и график (установленные сроки защиты) лабораторных работ:

Наименование работы	Установленный срок защиты (недели семестра)
Лабораторная работа №1	3
Лабораторная работа №2	4
Лабораторная работа №3	5
Лабораторная работа №4	7
Лабораторная работа №5	9
Лабораторная работа №6	11
Лабораторная работа №7	13
Лабораторная работа №8	16

Пример перечня заданий лабораторных работ

Задача 1. Год юбилея

Предлагается шаблон программы "Hello, world" и шаблон файла сборки Makefile.

Необходимо выполнить следующие действия вместе с инструктором:

1. Подготовить каталог для программы
2. Скопировать код программы в каталог
3. Выполнить сборку и запуск вручную
4. Выполнить сборку посредством Makefile (отдельно и из emacs)
5. Внести намеренную ошибку, разобрать диагностическое сообщение, обратить внимание на номер строки с ошибкой
6. Выполнить программу в отладчике:
 - выполнить по шагам, отслеживая значение `year`
 - поставить контрольную точку на последний вызов функции `fprintf`
 - обратить внимание на неинициализированную переменную `year`

Самостоятельно модифицировать программу следующим образом: пользователь вводит текущий год и год рождения, необходимо рассчитать и вывести на экран год очередного юбилея.

Задача 2. Маленький Гаусс

Существует легенда об одном эпизоде из жизни великого математика Карла Фридриха Гаусса, произошедшем с ним в школьные годы. Однажды учитель задал классу нелегкую задачу: сложить числа 1, 2, 3 — и т.д. до 20. Он надеялся освободить себе немного времени, пока ученики будут заняты нахождением суммы такого длинного ряда чисел, и был поэтому неприятно удивлен, когда маленький Гаусс шагнул вперед — в то время, как остальные ученики еще только собирались приступить к работе, — положил грифельную доску на конторку учителя и сказал: "Готово"...

Напишите программу, имитирующую этот эпизод. Программа должна запрашивать со стандартного ввода натуральное число N (гарантируется $N < 1000$), вычислять и выводить на стандартный вывод сумму чисел от 1 до N .

Задача 3. Цифровой корень

Цифровой корень натурального числа N вычисляется с помощью следующей итеративной процедуры: вычисляется сумма цифр числа, затем сумма цифр суммы и т.д. до тех пор, пока не останется одна цифра.

Напишите программу для вычисления и печати на экране цифрового корня натурального N ($N < 10^{1000000}$), заданного стандартным вводом.

Задача 4. Числа Шрёдера

Число Шрёдера представляет число траекторий, позволяющих переместить точку из югозападного в северо-восточный углы сетки размером $M \times M$, используя на каждом шаге один из трех возможных ходов: на одну клетку вверх, одну клетку вправо и одну клетку по диагонали вверх-вправо, при этом проходящих ниже диагонали, соединяющей юго-западный и северо-восточный углы.

Напишите программу для вычисления и печати на экране N первых чисел Шрёдера. Число N задается аргументом командной строки, если аргумент отсутствует — запрашивается со стандартного ввода.

Задача 5. Гипотеза Гольдбаха

В 1742 г. немецкий математик Христиан Гольдбах в письме к Эйлеру сформулировал гипотезу о том, что любое нечётное число не меньше семи можно представить в виде суммы трёх простых чисел. В ответ Эйлер предложил более сильный вариант: любое чётное число не меньше четырёх можно представить в виде суммы двух простых чисел. Эта гипотеза на сегодняшний день не подтверждена (то есть является так называемой «проблемой»), но и не опровергнута. С использованием современных вычислительных машин удалось проверить ее истинность для огромного набора значений — реализовать собственную программу проверки предлагается и Вам.

В реализации программы должны быть соблюдены следующие требования, связанные с обработкой входных и выходных данных:

- если задан первый аргумент командной строки, то программа пытается считать входные данные из файла, заданного первым аргументом, при этом:

- если открыть файл не удастся, программа выводит соответствующее сообщение и завершается с кодом возврата, равным 1;
- если файл доступен для чтения, но считать заданные условием задачи данные не удастся, программа выводит соответствующее сообщение и завершается с кодом возврата, равным 2;
- если первый аргумент командной строки не задан, то программа пытается считать входные данные из стандартного ввода, при этом:
 - если считать заданные условием задачи данные не удастся, программа выводит соответствующее сообщение и завершается с кодом возврата, равным 3;
- гарантируется, что если все входные данные присутствуют, они заданы корректно (т.е. если в условии задачи указано, что число положительное — оно точно положительное), дополнительная «защита от дурака» **не требуется**;
- если задан второй аргумент командной строки, то программа пытается записать результат в файл, заданный вторым аргументом, при этом:
 - если открыть файл не удастся, программа выводит соответствующее сообщение и завершается с кодом возврата, равным 4;
 - если файл доступен для записи, но записать заданные условием задачи данные не удастся, программа выводит соответствующее сообщение и завершается с кодом возврата, равным 5;
- если второй аргумент командной строки не задан, то программа пытается записать результат в стандартный вывод, при этом:
 - если записать данные не удастся, программа выводит соответствующее сообщение и завершается с кодом возврата, равным 6;
- коды возвратов должны быть определены константами с помощью enum.

Из входного файла считывается пара натуральных четных чисел N и M ($4 \leq N$, $M < 1000$, $N < M$). В выходной файл для каждого K из диапазона $[N, M]$ в порядке возрастания K вывести K , общее количество разложений, и одну пару простых чисел X и Y через пробел, таких что $X + Y = N$. В случае, если разложений числа несколько, следует вывести пару с наименьшим X . Для каждого значения диапазона разложение должно быть напечатано на отдельной строке. Разложения, отличающиеся только порядком следования чисел, различными не считаются. Пример:

```
Вход: 4 10
Выход: 4 1 2 2
        6 1 3 3
        8 1 3 5
        10 2 3 7
```

Задача 6. Обработка растровых изображений

Перед Вами поставлена задача разработать несколько функций для перспективного редактора растровой графики. Для демонстрации реализованного функционала Вам также необходимо написать небольшую демонстрационную программу.

Текущая версия редактора не поддерживает формат BMP, Вам потребуется исправить это досадное недоразумение.

Аргументы командной строки программы задают последовательность файлов, предположительно в формате BMP. Выполнив необходимые проверки, реализуйте преобразование данных каждого изображения, применив фильтр Grayscale ($R' = G' = B' = (R * 0.3 + G * 0.59 + B * 0.11)$) или Sepia ($R' = (R * 0.393 + G * 0.769 + B * 0.189)$; $G' = (R * 0.349 + G * 0.686 + B * 0.168)$; $B' = (R * 0.272 + G * 0.534 + B * 0.131)$).

Задача 7. Коды Хэмминга

При передаче данных по зашумленному каналу связи возможны искажения, в результате которых принятое получателем информационное сообщение отличается от переданного отправителем. Для обнаружения и даже коррекции ошибочных фрагментов сообщения без повторной передачи используются различные методы, так или иначе связанные с внедрением в сообщение специальных проверочных битов (или наборов битов). Коды Хэмминга представляют собой один из наиболее вариантов самокорректирующихся кодов.

В рассматриваемом в нашей задаче примере сообщение представляет собой поток байтов. При этом каждые четыре бита сообщения дополняются четырьмя проверочными битами, вычисленными по следующему правилу (символом \oplus обозначена сумма по модулю 2, операция xor):

$$\begin{aligned}x_4 &= x_0 \oplus x_1 \oplus x_2 \\x_5 &= x_1 \oplus x_2 \oplus x_3 \\x_6 &= x_0 \oplus x_2 \oplus x_3 \\x_7 &= x_4 \oplus x_5 \oplus x_6\end{aligned}$$

Таким образом, каждый байт сообщения кодируется двумя байтами, при этом первый по порядку байт содержит младшие четыре бита исходного байта сообщения и проверочные биты, а второй — старшие четыре бита исходного байта и проверочные биты, например:

Исходное сообщение длиной в один байт: 0xAЕ (двоичный код - 10101110), младшая половина 0xE (1110), старшая половина 0xA (1010).

Кодируем младшую половину:

$$\begin{aligned}x_4 &= 0 \oplus 1 \oplus 1 = 0 \\x_5 &= 1 \oplus 1 \oplus 1 = 1 \\x_6 &= 0 \oplus 1 \oplus 1 = 0 \\x_7 &= 0 \oplus 1 \oplus 0 = 1\end{aligned}$$

Кодируем старшую половину:

$$\begin{aligned}x_4 &= 0 \oplus 1 \oplus 0 = 1 \\x_5 &= 1 \oplus 0 \oplus 1 = 0 \\x_6 &= 0 \oplus 0 \oplus 1 = 1 \\x_7 &= 1 \oplus 0 \oplus 1 = 0\end{aligned}$$

Итоговые два байта: 0xAЕ (10101110) 0x5A (01011010)

Вам необходимо написать программу для декодирования сообщения, сконструированного по заданным выше правилам. Программа должна автоматически исправлять ошибочные байты. Декодированное сообщение необходимо вывести в стандартный поток вывода.

Задача 8. Раздельная трансляция программы работы с кодами Хэмминга

Программа «Коды Хэмминга» содержит несколько вспомогательных функций. Вам необходимо вынести эти функции в отдельный файл кода, создать заголовочный файл с прототипами функций и написать файл сборки (Makefile). Напишите сценарий оболочки (shell), позволяющий выполнить запуск программы для серии входных тестовых файлов, сравнить результат с эталонными файлами и сформировать файл отчета с именем в следующем формате: testreport-20111225-111927. Здесь 20111225 и 111927 — дата и время создания отчета.

3.6. Самостоятельная работа

Самостоятельная работа студентов подразумевает следующие виды работ:

- разбор лекционного материала и подготовка к коллоквиуму и экзамену из примерного расчета 1 час самостоятельной работы на 2 часа лекционных занятий;
- выполнение лабораторных работ из примерного расчета 1 час самостоятельной работы на 1 аудиторный час лабораторных занятий.

3.7 Информационное обеспечение дисциплины в системе электронного (дистанционного) обучения

В настоящее время для поддержки учебного процесса используется страница дисциплины по адресу: <http://cs.petrSU.ru/~aborod/inf/>

В рамках дальнейшей работы планируется создание дистанционного курса.

4. Образовательные технологии

Сочетание традиционных образовательных технологий в форме лекции с интерактивными семинарскими занятиями и компьютерными автоматизированными информационными технологиями при выполнении лабораторных работ и проведении контрольных мероприятий (коллоквиума, экзамена).

Далее предполагается создание системы тестов с использованием дистанционных образовательных технологий.

Удельный вес занятий, проводимых в интерактивных формах, составляет не менее 50% аудиторных занятий. Занятия лекционного типа для соответствующих групп студентов не могут составлять более 50% аудиторных занятий.

5. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения

Контроль качества подготовки осуществляется путем проверки теоретических знаний и практических навыков посредством промежуточных контрольных работ, коллоквиума на 12-й неделе, экзамена в конце 1 семестра, проверки и приема текущих семестровых заданий и лабораторных работ.

5.1 Контрольные тесты по темам / разделам / в целом по дисциплине

Пример теста – файл «Итоговый-тест.docx»

5.2 Примерные варианты расчетно-графических работ

Не предусмотрены ФГОС

5.3 Задания для промежуточных и итоговых контрольных работ и аттестаций

Пример контрольной работы – файл «Контрольная-02.pdf»

5.4 Вопросы к экзамену.

Примерный перечень экзаменационных вопросов

1. Концепция типа данных. Типы int, char и модификаторы знака (signed и unsigned) и размера (short и long). Формат внутреннего представления. Непосредственные целочисленные константы и суффиксы типов. Размеры целочисленных типов (limits.h).
2. Вещественнозначные типы float, double, long double. Внутреннее представление и особенности вычислений с плавающей точкой. Непосредственные константы с плавающей точкой и суффиксы типов. Характеристики объектов с плавающей точкой (float.h). Средства математической библиотеки (math.h).
3. Определение и инициализация переменных (в том числе составных типов). Определение именованных констант: модификатор const, перечисления (enum). Препроцессорные подстановки (#define). Использование typedef для именования типов. Типы в стандартной библиотеке, определенные с помощью typedef. Явное приведение типов.
4. Арифметические и логические операции, инкременты, тернарная операция. Приоритеты и особенности использования. Присваивание и приведение типов в присваивании.
5. Одномерные и двумерные массивы, нуль-терминальные строки. Определение и доступ к элементам. Типовые алгоритмы обработки. Ошибки выхода за границу

- выделенной памяти.
6. Указатели: определение, присваивание указателю адреса объекта, разыменование указателя. Значение NULL. Адресная арифметика, особенности приоритетов операций. Нетипизированные указатели void*. Приведение типов указателей. Указатели на указатели.
 7. Массивы и указатели. Имя массива как указатель, использование указателей для обработки элементов одномерного или многомерного (распределение элементов многомерного массива в памяти!) массива. Использование указателей для передачи массивов в качестве параметров функций.
 8. Массивы динамической памяти. Функции работы с динамической памятью (malloc, calloc, realloc, free), определенные в заголовочном файле stdlib.h. Проблема утечек памяти.
 9. Управление потоком: ветвление. Оператор if, полная и неполная формы. Вложенные if. Оператор switch . Допустимые типы констант. Провал в следующий case и использование break. Ветвь default.
 10. Управление потоком: циклы. Операторы циклов while и do-while. Заикливание. Использование break и continue. Оператор цикла for. Заикливание. Использование break и continue.
 11. Поточный I/O: потоки и файлы, структура FILE, стандартные потоки stdin, stdout, stderr. Текстовые и двоичные файлы: функции чтения и записи символов и строк, функции блочного чтения и записи.
 12. Поточный I/O: потоки и файлы, открытие и закрытие потоков ввода-вывода, позиционирование в потоке.
 13. Поточный I/O: потоки и файлы, функции форматного ввода-вывода.
 14. Обработка символов и строк: анализа и преобразование символов вручную и с помощью средств, определенных в заголовочном файле ctype.h.
 15. Обработка символов и строк: функции str* и mem*, определенные в string.h.
 16. Средства доступа к аргументам командной строки.
 17. Функции в Си: общий вид определения функции, объявление функции, вызов.
 18. Функции в Си: передача параметров в функцию, возвращаемое значение.
 19. Функции в Си: указатели на функции, пример использования.
 20. Функции в Си: функции в качестве параметров других функций, qsort и bsearch.
 21. Функции в Си: стек вызовов функции.
 22. Функции в Си: рекурсия.
 23. Определение объекта структурного типа и указателя на структурный объект. Прямой и косвенный доступ к полям структуры. Инициализация структуры.
 24. Вложенные структуры, массивы структур и массивы как поля структуры.
 25. Указатели на структуры в качестве полей структуры. Связное представление данных. Формирование линейного списка и операции над ним.
 26. Функции генерирования случайных чисел. Инициализация датчика. Получение случайных чисел, равномерно распределенных на заданном промежутке.
 27. Функции работы с временем и датой. Временная отметка и структура времени tm. Преобразование и форматирование даты и времени.
 28. Создание и использование функций с переменным числом параметров (stdarg.h).
 29. Препроцессорные средства: замены, включения, условная компиляция, параметризованные макроопределения.
 30. Раздельная трансляция. Использование заголовочных файлов
 31. Компоновка. Алгоритм разрешения зависимостей при компоновке с использованием библиотек функций. Алгоритм разрешения множественно определенных имен.
 32. Создание и использование статических библиотек
 33. Создание и использование динамических библиотек. Компоновка во время

- выполнения программы
34. Битовые операции. Приоритеты, особенности использования. Реализация установки, сброса и тестирования заданного бита
 35. Использование объединений и битовых полей в структурах
 36. Конструирование абстрактных типов данных. Инкапсуляция и неполные типы
 37. Диагностика ошибок средствами стандартной библиотеки
 38. Управление сигналами
 39. Стандарт Unicode/UCS и кодировки Unicode. Типы данных для представления символов Unicode
 40. Функции работы с мультбайтовыми символами и строками

Пример экзаменационного билета в 1 семестре

1. Стек вызовов функций. Размещение параметров в соответствии с соглашениями языка Си. Пример переопределения адреса возврата функции при переполнении стека вызовов.
2. Создание и использование функций с переменным числом параметров.

Пример билета коллоквиума

1. Операторы организации циклов в языке Си.
2. Адреса и указатели. Адресная арифметика. Значение NULL.

5.5 Балльно-рейтинговая система

Таблица 6.

Распределение времени и баллов при контроле успеваемости

№	Испытания/Формы СРС	Время на подготовку/выполнение, час	Баллы	Примечание
1	Лабораторная работа 1	4	2	
1	Лабораторная работа 2	4	2	
1	Лабораторная работа 3	8	4	
1	Лабораторная работа 4	8	4	
1	Лабораторная работа 5	16	8	
1	Лабораторная работа 6	16	8	
1	Лабораторная работа 7	16	8	
1	Лабораторная работа 8	8	4	
1	Коллоквиум	10	2	
1	Экзамен	36		
	Итого за семестр	126 часов	40	

Реализация балльно-рейтинговой системы

Предусмотрена балльно-рейтинговая система, при этом в рамках выполнения лабораторных работ в течение семестра студент может набрать до 40 баллов, в рамках коллоквиума на 12-й неделе обучения – до 20 баллов, на финальном экзамене – до 40 баллов.

Таблица 8.

Общая сумма баллов	Автоматическая оценка	Итоговая оценка
96 - 100	Отлично, А	5
91 - 95	Очень хорошо, В	
76 - 90	Хорошо, С	4
61 – 75	Удовлетв., D	3

51 - 60	Посредственно, Е	3
41 – 50	Условно неудовлет- ворительно, FX	2
< 40	Безусловно неуд., F	2

6. Методические указания для студентов

6.1. Методические указания для выполнения практических заданий

Практические занятия проводятся в классах с доской под управлением инструктора, для подготовки к практическому занятию необходимо повторить лекционный материал.

6.2. Методические указания для выполнения лабораторных работ

Лабораторные занятия проводятся в дисплейных классах. Всего лабораторных работ — 8 (описание и сроки приведены ниже). Каждая работа засчитывается при удовлетворении **всем требованиям протокола оценки** и может быть оценена в зависимости от срока защиты. За каждую неделю задержки базовая оценка за работу уменьшается вдвое. Для получения зачета должны быть сданы все работы, количество набранных по результатам защиты работ баллов участвует в общей экзаменационной оценке (40%).

Защита работы включает в себя:

- предоставление файла с исходным кодом программы и сопутствующих файлов (Makefile и т. п.)
- презентацию исходного кода (демонстрация, пояснение, ответы на вопросы преподавателя)
- презентацию программы (демонстрация сборки и исполнения программы, ответы на вопросы преподавателя)
- предоставление других документов, указанных в условии задачи

6.3. Методические указания для выполнения самостоятельной работы

Самостоятельную работу следует спланировать таким образом, чтобы еженедельно тратить 1 час на разбор лекционного материала и работу с учебниками; 2 часа на выполнение заданий лабораторных работ. Оставшиеся часы самостоятельной работы следует направить на подготовку к экзамену.

7. Учебно-методическое и информационное обеспечение дисциплины

7.1 Основная литература:

Богоявленская О. Ю. Введение в программирование. – 2-е изд., стер. – Петрозаводск, 2006.

7.2 Дополнительная литература:

Керниган Б. Язык программирования Си / Б. Керниган, Д. Ритчи — 2-е изд. — М.: Вильямс, 2007. — С. 304. — ISBN 0-13-110362-8

Кочан С. Программирование на языке Си. / С. Кочан. — М.: Вильямс, 2007. — С. 496. — ISBN 5-8459-1088-9

Прага С. Язык программирования C: Лекции и упражнения / С. Прага. — 1-е изд. — М.: Вильямс, 2006. — С. 960. — ISBN 5-8459-0986-4

7.3 Периодические издания

Нет

7.4 Методические разработки ВУЗа

Богоявленская О. Ю. Введение в программирование. – 2-е изд., стер. – Петрозаводск, 2006.

7.5 Интернет-ресурсы

Калинина Н.А. Основы программирования на языке C [Электронный ресурс] / Intuit.ru : Интернет университет информационных технологий. URL: <http://intuit.ru/department/pl/c/>

Кетков Ю.Л. Введение в языки программирования C и C++ [Электронный ресурс] / Intuit.ru : Интернет университет информационных технологий. URL: <http://intuit.ru/department/pl/c/>

7.6 Другие материалы

Демидович Е.М. Основы алгоритмизации и программирования. Язык C: Учебник для ВУЗов / Е.М. Демидович. — СПб.: ВHV-Санкт-Петербург, 2006. — С. 448. — ISBN 5-94157-459-2

Дорогов В.Г. Основы программирования на языке C. Учебное пособие / В.Г. Дорогов, Е.Г. Дорогова; Под общ. ред. проф. Л.Г. Гагариной — М.: ИД ФОРУМ: ИНФРА-М, 2011. - 224 с.: 60x90 1/16. - (Высшее образование). (переплет) ISBN 978-5-8199-0471-8

Гриффитс А. GCC. Полное руководство / А. Гриффитс — ТИД «ДС», 2004. — С. 624. — ISBN 966-7992-33-0, 0-07-222405-3

Подбельский В.В. Программирование на языке Си / В.В. Подбельский, С.С.Фомин. — М.: Финансы и статистика, 2005 — С. 600. — ISBN 5-279-02180-6, 5-279-02180-2

Bermudez M. Study Guide for C Programming: A Modern Approach / M. Bermudez. — W. W. Norton & Company, 1998. — С. 256. — ISBN 0-393-96946-0, 978-0393969467

Bryant R. Computer Systems: A Programmer's Perspective / R. Bryant, D. O'Hallaron — AddisonWesley, 2010. — С. 1080. — ISBN 0136108040, 978-013-610804-7

Kalicharan N. C Programming. An Advanced Course / N. Kalicharan — CreateSpace, 2008. — С. 224. — ISBN 1-438-27557-9, 978-1-438-27557-4

Kalicharan N. Data Structures in C / N. Kalicharan — CreateSpace, 2008. — С. 308. — ISBN 1-438-25327-3, 978-1-438-25327-5

King K.N. C Programming: A Modern Approach / K.N. King. — W. W. Norton & Company, 2008. — С. 832. — ISBN 0-393-97950-4, 978-0-393-97950-3

8. Информационно-техническое и материально-техническое обеспечение дисциплины

В курсе используются следующие свободно-распространяемые программные продукты:

- ОС на базе ядра Linux (openSUSE 12.2);
- семейство компиляторов gcc;
- отладчик gdb;
- утилита make;
- командный интерпретатор bash;
- утилиты пакета binutils.

Петрозаводский университет обеспечен необходимым комплектом лицензионного программного обеспечения.

9. Дополнительные материалы (презентации к лекциям и т.д.)

Для курса разработаны презентации к лекциям и конспекты части лекций. См. приложенные файлы.