**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»
(ТУСУР)

УТВЕРЖДАЮ
Директор департамента образования
_____ П. Е. Троян
«___» _____ 20__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

# Информационные технологии

Уровень образования: **высшее образование - бакалавриат**
Направление подготовки / специальность: **11.03.04 Электроника и наноэлектроника**
Направленность (профиль) / специализация: **Промышленная электроника**
Форма обучения: **очная**
Факультет: **ФЭТ, Факультет электронной техники**
Кафедра: **ПрЭ, Кафедра промышленной электроники**
Курс: **1**
Семестр: **1, 2**
Учебный план набора 2017 года

Распределение рабочего времени

| № | Виды учебной деятельности | 1 семестр | 2 семестр | Всего | Единицы |
|---|---|---|---|---|---|
| 1 | Лекции | 20 | 20 | 40 | часов |
| 2 | Практические занятия | 18 | 18 | 36 | часов |
| 3 | Лабораторные работы | 16 | 16 | 32 | часов |
| 4 | Всего аудиторных занятий | 54 | 54 | 108 | часов |
| 5 | Самостоятельная работа | 54 | 54 | 108 | часов |
| 6 | Всего (без экзамена) | 108 | 108 | 216 | часов |
| 7 | Подготовка и сдача экзамена | 36 | 36 | 72 | часов |
| 8 | Общая трудоемкость | 144 | 144 | 288 | часов |
| | | 4.0 | 4.0 | 8.0 | З.Е. |

Экзамен: 1, 2 семестр

Томск 2018

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шелупанов А.А.
Должность: Ректор
Дата подписания: 23.08.2017
Уникальный программный ключ:
c53e145e-8b20-45aa-9347-a5e4dbb90e8d

Согласована на портале № 46394

Рассмотрена и одобрена на заседании кафедры
протокол № 50 от «26» 4 2018 г.

# ЛИСТ СОГЛАСОВАНИЯ

Рабочая программа дисциплины составлена с учетом требований федерального государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки (специальности) 11.03.04 Электроника и наноэлектроника, утвержденного 12.03.2015 года, рассмотрена и одобрена на заседании кафедры ПрЭ «___» _____ 20__ года, протокол №_____.

Разработчик:

| | | |
|---|---|---|
| зав. кафедрой, профессор каф. ПрЭ | _____ | С. Г. Михальченко |
| Заведующий обеспечивающей каф. ПрЭ | _____ | С. Г. Михальченко |

Рабочая программа дисциплины согласована с факультетом и выпускающей кафедрой:

| | | |
|---|---|---|
| Декан ФЭТ | _____ | А. И. Воронин |
| Заведующий выпускающей каф. ПрЭ | _____ | С. Г. Михальченко |

Эксперты:

| | | |
|---|---|---|
| Зам. зав. кафедрой ПРЭ по методической работе, профессор каф. ПРЭ | _____ | Н. С. Легостаев |
| доцент каф. ФЭ | _____ | И. А. Чистоедова |

Согласована на портале № 46394

## 1. Цели и задачи дисциплины

### 1.1. Цели дисциплины

Целью преподавания дисциплины «Информационные технологии» является изучение основ современных способов обработки информации с использованием средств вычислительной техники, знакомство с популярными программными продуктами, применяемыми как в инженерных расчетах, так в офисных технологиях, освоение навыков программирования на языке С++.

### 1.2. Задачи дисциплины

– Задачи изучения дисциплины «Информационные технологии» состоят в последовательном освоении материала по основам вычислительной техники и по ее применению в решении учебных и исследовательских задач. В ходе изучения дисциплины «Информационные технологии» студенты знакомятся с устройством, принципами работы компьютера, получают представление о системном и прикладном программном обеспечении, овладевают навыками практической работы с компонентами MS Office. К задачам освоения дисциплины относится изучение пакета математических расчетов MathCAD и основ программирования на языке С++.

## 2. Место дисциплины в структуре ОПОП

Дисциплина «Информационные технологии» (Б1.Б.8) относится к блоку 1 (базовая часть).

Предшествующими дисциплинами, формирующими начальные знания, являются: Информационные технологии.

Последующими дисциплинами являются: Информационные технологии, Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты, Математическое моделирование и программирование, Методы анализа и расчета электронных схем, Микропроцессорные устройства и системы, Программирование и отладка микроконтроллеров, Учебно-исследовательская работа.

## 3. Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

– ОПК-6 способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий;

– ОПК-7 способностью учитывать современные тенденции развития электроники, измерительной и вычислительной техники, информационных технологий в своей профессиональной деятельности;

– ОПК-9 способностью использовать навыки работы с компьютером, владеть методами информационных технологий, соблюдать основные требования информационной безопасности;

В результате изучения дисциплины обучающийся должен:

– **знать** принципы работы персональных компьютеров, идеологию современных операционных систем, основные методы разработки алгоритмов и программ, процесс компиляции и создания программного продукта, способы размещения в памяти компьютера структур данных, используемых для представления типовых информационных объектов и обращения к ним, типовые алгоритмы обработки данных, принципы использования прикладных математических пакетов.

– **уметь** давать стратегическую оценку решаемой технической задачи, основывающуюся на понимании и ясном представлении целей исследований и абстрагировании от шаблонов и алгоритмов поиска решений; решать задачи обработки данных с помощью современных инструментальных средств конечного пользователя (MatCAD); создавать, отлаживать и тестировать программы на платформеннонезависимом языке высокого уровня (С++); представлять результаты исследований в удобном формате (MsOffice).

– **владеть** инструментами создания современных программных продуктов - отладочными средами высокого уровня; навыками написания, тестирования и отладки программных средств; навыками постановки научно-технических задач, выбора методов их решения, получения и предоставления результатов, интерпретации и анализа полученных данных.

## 4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 8.0 зачетных единицы и представлена в табли-

Согласована на портале № 46394

це 4.1.

Таблица 4.1 – Трудоемкость дисциплины

| Виды учебной деятельности | Всего часов | Семестры | |
|---|---|---|---|
| | | 1 семестр | 2 семестр |
| Аудиторные занятия (всего) | 108 | 54 | 54 |
| Лекции | 40 | 20 | 20 |
| Практические занятия | 36 | 18 | 18 |
| Лабораторные работы | 32 | 16 | 16 |
| Самостоятельная работа (всего) | 108 | 54 | 54 |
| Оформление отчетов по лабораторным работам | 24 | 12 | 12 |
| Проработка лекционного материала | 40 | 20 | 20 |
| Подготовка к практическим занятиям, семинарам | 44 | 22 | 22 |
| Всего (без экзамена) | 216 | 108 | 108 |
| Подготовка и сдача экзамена | 72 | 36 | 36 |
| Общая трудоемкость, ч | 288 | 144 | 144 |
| Зачетные Единицы | 8.0 | 4.0 | 4.0 |

## 5. Содержание дисциплины

### 5.1. Разделы дисциплины и виды занятий

Разделы дисциплины и виды занятий приведены в таблице 5.1.

Таблица 5.1 – Разделы дисциплины и виды занятий

| Названия разделов дисциплины | Лек., ч | Прак. зан., ч | Лаб. раб., ч | Сам. раб., ч | Всего часов (без экзамена) | Формируемые компетенции |
|---|---|---|---|---|---|---|
| 1 семестр | | | | | | |
| 1 Средства обработки информационных ресурсов. Понятие алгоритма. Обращение к данным. Алгоритмические конструкции. Блок-схемы. Операторы. | 4 | 8 | 0 | 8 | 20 | ОПК-6, ОПК-7, ОПК-9 |
| 2 Обзор вычислительных математических пакетов. Принципы обращения к ячейкам памяти. Адрес переменной. Указатели. | 4 | 2 | 4 | 12 | 22 | ОПК-6, ОПК-7, ОПК-9 |
| 3 Подпрограммы: заголовок, описание, реализация алгоритма, вызов подпрограммы, передача параметров в тело функции. | 4 | 2 | 4 | 12 | 22 | ОПК-6, ОПК-7, ОПК-9 |
| 4 Массивы. Статические и динамические объекты, адресация. Одномерные и двумерные массивы. | 6 | 2 | 8 | 12 | 28 | ОПК-6, ОПК-7, ОПК-9 |
| 5 Возможности представления информации. Типы файлового хранения, | 2 | 4 | 0 | 10 | 16 | ОПК-6, ОПК-7, ОПК-9 |

Согласована на портале № 46394

| | | | | | | |
|---|---|---|---|---|---|---|
| расширения файлов. Стандартные подпрограммы обработки данных. MsOffice. | | | | | | |
| Итого за семестр | 20 | 18 | 16 | 54 | 108 | |
| 2 семестр | | | | | | |
| 6 Работа с символьными массивами. Потоковый ввод-вывод. Динамические строки. | 4 | 4 | 0 | 10 | 18 | ОПК-6, ОПК-7, ОПК-9 |
| 7 Файловая система. Программная реализация. Дескриптор файла. | 4 | 4 | 4 | 12 | 24 | ОПК-6, ОПК-7, ОПК-9 |
| 8 Структурированные типы данных, битовые поля, объединения. Приемы программирования микроконтроллеров, битовые операции. | 4 | 2 | 8 | 12 | 26 | ОПК-6, ОПК-7, ОПК-9 |
| 9 Понятие класса. Поля и методы класса. Спецификаторы доступа. Статические и динамические объекты класса. | 4 | 4 | 4 | 12 | 24 | ОПК-6, ОПК-7, ОПК-9 |
| 10 Доступ к данным. Информационная безопасность. | 4 | 4 | 0 | 8 | 16 | ОПК-6, ОПК-7, ОПК-9 |
| Итого за семестр | 20 | 18 | 16 | 54 | 108 | |
| Итого | 40 | 36 | 32 | 108 | 216 | |

### 5.2. Содержание разделов дисциплины (по лекциям)

Содержание разделов дисциплин (по лекциям) приведено в таблице 5.2.

Таблица 5.2 – Содержание разделов дисциплин (по лекциям)

| Названия разделов | Содержание разделов дисциплины (по лекциям) | Трудоемкость, ч | Формируемые компетенции |
|---|---|---|---|
| 1 семестр | | | |
| 1 Средства обработки информационных ресурсов. Понятие алгоритма. Обращение к данным. Алгоритмические конструкции. Блок-схемы. Операторы. | Понятие информации. Предмет информатики. Информационные процессы. Информатизация общества. Компьютеры. Технология проектирования и отладки программ. Инструментальные средства программирования. Основные принципы создания программного обеспечения. Основы программирования на языке верхнего уровня (на примере C++). Процесс создания программного кода. Отладка программ. Трассировка программного кода. | 2 | ОПК-6, ОПК-7 |
| | Алгоритмические конструкции. Операторы выбора. Перечисляемый тип данных. Операторы цикла. Использование переменных логического типа. Оператор безусловного перехода. Организация диалога с пользователем. | 2 | |
| | Итого | 4 | |
| 2 Обзор | Система компьютерной алгебры | 2 | ОПК-7, ОПК-9 |

Согласована на портале № 46394

| | | | |
|---|---|---|---|
| вычислительных математических пакетов. Принципы обращения к ячейкам памяти. Адрес переменной. Указатели. | Mathematica. Пакет символьной математики MAPLE. Пакет математических и инженерных вычислений MATLAB. Программный комплекс MathCAD. | | |
| | Статическое и динамическое распределение памяти. Функции динамического распределения памяти. Освобождение памяти. Операторы new и delete. Обращение к переменным "по имени" и "по адресу". Отличие указателя от динамической переменной. | 2 | |
| | Итого | 4 | |
| 3 Подпрограммы: заголовок, описание, реализация алгоритма, вызов подпрограммы, передача параметров в тело функции. | Подпрограммы: процедуры и функции. Структура подпрограммы. Формальные и фактические параметры. Передача параметров в тело функции. Вызов подпрограммы. Возвращение значения, оператор return. Область видимости переменной. | 2 | ОПК-7, ОПК-9 |
| | Описание и вызов подпрограммы. Прототип подпрограммы. Адрес функции, указатель на подпрограмму. Область видимости переменной. Глобальные и локальные переменные. Перегрузка функций. | 2 | |
| | Итого | 4 | |
| 4 Массивы. Статические и динамические объекты, адресация. Одномерные и двумерные массивы. | Массивы - хранилище однотипных данных. Одномерные (вектора) и многомерные (матрицы) массивы. Описание массива, обращение к элементам массива по индексу. Генератор случайных чисел. Работа с массивами в цикле. | 2 | ОПК-6, ОПК-7 |
| | Статические и динамические массивы. Функции calloc(), malloc() и free(). Адрес массива = адрес начальной ячейки. Указатель на массив. Передача массива в функцию. Динамические одномерные и двумерные массивы. Обращение к элементам массива "адрес+смещение". | 2 | |
| | Двумерные статические и динамические. Три способа размещения в памяти двумерных массивов. Освобождение памяти. Обращение к ячейкам двумерного массива. | 2 | |
| | Итого | 6 | |
| 5 Возможности представления информации. Типы файлового хранения, расширения файлов. Стандартные подпрограммы | Обращение к оборудованию ПК. BIOS. Setup BIOS. Драйверы. Работа с аудио-, видео- и графической информацией. Кодер/декодер. Приложения для работы с текстом (Word), таблицами (Excel), структурированными данными (Access) и презентациями (PowerPoint). | 2 | ОПК-6, ОПК-7, ОПК-9 |

Согласована на портале № 46394

| | | | |
|---|---|---|---|
| обработки данных. MsOffice. | Итого | 2 | |
| Итого за семестр | | 20 | |
| 2 семестр | | | |
| 6 Работа с символьными массивами. Потоковый ввод-вывод. Динамические строки. | Строки символов. Строка – массив символов. Вывод строки на экран. Последний элемент строки. Указатель на строку. | 2 | ОПК-6, ОПК-9 |
| | Указатель на строку. Копирование и сравнение строк. Статические и динамические строки. Функции преобразования строковых типов данных. | 2 | |
| | Итого | 4 | |
| 7 Файловая система. Программная реализация. Дескриптор файла. | Файловые подсистемы ОС, хранение данных на диске. Дескриптор файла в программе. Доступ к файлу, совместный доступ, транзакции. Файловые операции (связать, открыть, закрыть, читать, писать, определить конец). | 2 | ОПК-6, ОПК-7 |
| | Работа с файлами при помощи потоков ввода-вывода. Библиотека . Файловые операции. Поиск в файле. | 2 | |
| | Итого | 4 | |
| 8 Структурированные типы данных, битовые поля, объединения. Приемы программирования микроконтроллеров, битовые операции. | Структуры. Описание в программе, доступ к полям структуры. Размещение в памяти. Указатели на структуру. Статические и динамические переменные структурного типа. Битовые поля. Объединения. Статические и динамические переменные структурного типа. | 2 | ОПК-7, ОПК-9 |
| | Приемы программирования микроконтроллеров, битовые операции. Поразрядные логические операции. Поразрядные операции сдвига. Обращение к разрядам при помощи битовых полей. | 2 | |
| | Итого | 4 | |
| 9 Понятие класса. Поля и методы класса. Спецификаторы доступа. Статические и динамические объекты класса. | Введение в понятие "класс". Поля и методы класса. Спецификаторы доступа, использование set- и get- методов. Конструктор и деструктор. | 2 | ОПК-6, ОПК-9 |
| | Указатель на объект класса. Статические и динамические объекты класса. Конструктор и деструктор для статических и динамических объектов класса. Отделение интерфейса программы от реализации. Дружественные функции. Перегрузка операторов. Наследование. | 2 | |
| | Итого | 4 | |
| 10 Доступ к данным. | Программный комплекс MathCAD. Струк- | 2 | ОПК-6, ОПК-7 |

Согласована на портале № 46394

| | | | |
|---|---|---|---|
| Информационная безопасность. | тура программы. Численные и аналитические вычисления, точность вычислений. Вывод графиков. Алгоритмические возможности MathCAD. Операторы, операторные блоки. Матричные операции MathCAD. Встроенные функции. | | |
| | Компьютерные сети, вопросы информационной безопасности. Локальные и глобальные компьютерные сети. Модель OSI. Адресация в сети, форматы обмена данными (сетевые протоколы). Программное обеспечение для работы в сетях. Службы сети Интернет. Угрозы информационных атак. Компьютерные вирусы. | 2 | |
| | Итого | 4 | |
| Итого за семестр | | 20 | |
| Итого | | 40 | |

**5.3. Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (последующими) дисциплинами**

Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (последующими) дисциплинами представлены в таблице 5.3.

Таблица 5.3 – Разделы дисциплины и междисциплинарные связи

| Наименование дисциплин | № разделов данной дисциплины, для которых необходимо изучение обеспечивающих и обеспечиваемых дисциплин | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Предшествующие дисциплины | | | | | | | | | | |
| 1 Информационные технологии | + | + | + | + | + | + | + | + | + | + |
| Последующие дисциплины | | | | | | | | | | |
| 1 Информационные технологии | | | | | | + | + | + | + | + |
| 2 Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты | + | + | | | + | | | | | + |
| 3 Математическое моделирование и программирование | + | | + | | + | | + | | + | |
| 4 Методы анализа и расчета электронных схем | | + | | + | + | | + | | + | + |
| 5 Микропроцессорные устройства и системы | + | | + | + | | | | + | | + |
| 6 Программирование и отладка микроконтроллеров | + | + | + | + | + | + | + | + | + | + |
| 7 Учебно-исследовательская работа | + | + | | | + | + | + | | | + |

Согласована на портале № 46394

### 5.4. Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий представлено в таблице 5.4.

Таблица 5.4 – Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

| Компетенции | Виды занятий | | | | Формы контроля |
|---|---|---|---|---|---|
| | Лек. | Прак. зан. | Лаб. раб. | Сам. раб. | |
| ОПК-6 | + | + | + | + | Отчет по индивидуальному заданию, Конспект самоподготовки, Отчет по лабораторной работе, Опрос на занятиях, Тест, Отчет по практическому занятию |
| ОПК-7 | + | + | + | + | Отчет по индивидуальному заданию, Конспект самоподготовки, Отчет по лабораторной работе, Опрос на занятиях, Тест, Отчет по практическому занятию |
| ОПК-9 | + | + | + | + | Отчет по индивидуальному заданию, Конспект самоподготовки, Отчет по лабораторной работе, Опрос на занятиях, Тест, Отчет по практическому занятию |

## 6. Интерактивные методы и формы организации обучения

Не предусмотрено РУП.

## 7. Лабораторные работы

Наименование лабораторных работ приведено в таблице 7.1.

Таблица 7.1 – Наименование лабораторных работ

| Названия разделов | Наименование лабораторных работ | Трудоемкость, ч | Формируемые компетенции |
|---|---|---|---|
| 1 семестр | | | |
| 2 Обзор вычислительных математических пакетов. Принципы обращения к ячейкам памяти. Адрес переменной. Указатели. | Научиться работать со статическими и динамическими переменными. Освоить применение операций адресации и разадресации. Научиться выделять память под динамические переменные и освобождать ее при помощи операторов new и delete. Освоить обращение к динамическим объектам. Научиться выделять память под динамические переменные и освобождать ее при помощи функций calloc, malloc и free. Научиться визуализировать адреса статических и динамических объектов. | 4 | ОПК-6, ОПК-7, ОПК-9 |
| | Итого | 4 | |
| 3 Подпрограммы: | Научиться создавать подпрограммы и вы- | 4 | ОПК-6, ОПК- |

Согласована на портале № 46394

| | | | |
|---|---|---|---|
| заголовок, описание, реализация алгоритма, вызов подпрограммы, передача параметров в тело функции. | зывать их из тела основной функции. Освоить применение оператора return, возвращающего значение функции. Научиться передавать параметры в функцию и корректно возвращать значения – результаты вычислений из функции. Освоить передачу параметров в функцию по значению, по ссылке и по указателю. Научиться создавать прототипы функций, освоить перегрузку функций. Изучить функции библиотеки math.h. Научиться пользоваться средствами пошаговой трассировки кода и просмотра текущих значений переменных. | | 7, ОПК-9 |
| | Итого | 4 | |
| 4 Массивы. Статические и динамические объекты, адресация. Одномерные и двумерные массивы. | Научиться создавать массивы, выделять память под элементы массива и инициировать их значениями. Научиться обращаться к элементам массивов как при помощи оператора имя[индекс], так и посредством методики *(адрес+смещение). Освоить понятия: адрес массива, адрес элемента массива, смещение, индекс. Освоить пошаговую трассировку программы с массивами, научиться отражать в окне watch элементы массива и их адреса. Научиться передавать массив в функцию. | 4 | ОПК-6, ОПК-7, ОПК-9 |
| | Освоить способ динамического захвата и освобождения памяти под двумерные массивы данных (матрицы) – при помощи оператора new и посредством функций malloc( ) и calloc( ). Повторить понятия, операции и закрепить умения и навыки матричной алгебры. Уяснить практическую разницу и сходство в языке С++ понятий «массив» и «указатель». Повысить навыки отладки программного кода на С++, трассировки программы и просмотра значений переменных в окне Watch. | 4 | |
| | Итого | 8 | |
| Итого за семестр | | 16 | |
| 2 семестр | | | |
| 7 Файловая система. Программная реализация. Дескриптор файла. | Научиться работать с файловыми переменными посредствоом подпрограмм библиотек fstream.h и stdio.h. Закрепить навыки работы со строками. Научиться открывать файлы для записи и чтения, создавать и уничтожать файлы, определять конец файла. | 4 | ОПК-6, ОПК-7, ОПК-9 |
| | Итого | 4 | |
| 8 Структурированные | Освоить умение работать со структурами | 4 | ОПК-6, ОПК- |

Согласована на портале № 46394

| | | | |
|---|---|---|---|
| типы данных, битовые поля, объединения. Приемы программирования микроконтроллеров, битовые операции. | данных. Научиться создавать новый комбинированный тип данных – структуру. Научиться обращаться к статическим и динамическим переменным этого типа и к массиву структур. Закрепить навыки работы с файлами и символьными строками. | | 7 |
| | Освоить умение работать с разрядами в байте. Научиться изменять произвольный бит в байте, не меняя остальных. Выводить на экран двоичный код переменной произвольного типа. Повысить понимание преобразований двоичного формата. Применять операции поразрядного сдвига и поразрядные логические операции, а также при помощи структур объединение (union) и битовые поля. | 4 | |
| | Итого | 8 | |
| 9 Понятие класса. Поля и методы класса. Спецификаторы доступа. Статические и динамические объекты класса. | Научиться создавать класс и описывать его свойства и методы. Изучить спецификаторы доступа private и public.Применение конструктора, деструктора, set- и get- методов. Работа с объектами класса: корректно создавать, инициализировать, получать значения свойств, визуализировать и удалять. Интерфейс и реализация класса. | 4 | ОПК-6, ОПК-7, ОПК-9 |
| | Итого | 4 | |
| Итого за семестр | | 16 | |
| Итого | | 32 | |

## 8. Практические занятия (семинары)

Наименование практических занятий (семинаров) приведено в таблице 8.1.

Таблица 8.1 – Наименование практических занятий (семинаров)

| Названия разделов | Наименование практических занятий (семинаров) | Трудоемкость, ч | Формируемые компетенции |
|---|---|---|---|
| 1 семестр | | | |
| 1 Средства обработки информационных ресурсов. Понятие алгоритма. Обращение к данным. Алгоритмические конструкции. Блок-схемы. Операторы. | Создание проекта на Visual C++, написать простейшую программу на языке C++, отладить и запустить на выполнение программу. Научиться задавать переменные и константы следующих типов: целого, вещественного, символьного и строкового. Освоить два способа ввода с клавиатуры и вывода на экран значений этих переменных – форматный и потоковый способы. Изучить спецификаторы, форматы ввода-вывода и управляющие символы. | 4 | ОПК-7, ОПК-9 |
| | Изучить операторы языка C++, научиться применять их при написании программ. Научиться отслеживать значения числовых и логических переменных и выражений в процессе отладки программы. | 4 | |

Согласована на портале № 46394

| | | | |
|---|---|---|---|
| | Освоить умение переводить мнемонические выражения в последовательность операторов. | | |
| | Итого | 8 | |
| 2 Обзор вычислительных математических пакетов. Принципы обращения к ячейкам памяти. Адрес переменной. Указатели. | Демонстрация возможностей MathCAD. Получение навыков отладки программы, выполнения вычислений и вывода результатов. Наработка умения задавать переменные, константы и графики на рабочем поле MathCAD при помощи шаблонов. | 2 | ОПК-6, ОПК-9 |
| | Итого | 2 | |
| 3 Подпрограммы: заголовок, описание, реализация алгоритма, вызов подпрограммы, передача параметров в тело функции. | Научиться оформлять блок-схему программы в соответствии с ЕСКД. Дерево вызова процедур и список формальных параметров подпрограммы. Соотнесение фактических параметров с формальными, обращение к подпрограммам по ссылке, по значению и по указателю. | 2 | ОПК-6, ОПК-7 |
| | Итого | 2 | |
| 4 Массивы. Статические и динамические объекты, адресация. Одномерные и двумерные массивы. | Работа с матрицами и векторами в MathCAD. Вычисление определителя матрицы A [4×4] путем разложения матрицы по соответствующему столбцу (или строке), сравнить результаты с вычислением определителя при помощи встроенного оператора. | 2 | ОПК-6, ОПК-9 |
| | Итого | 2 | |
| 5 Возможности представления информации. Типы файлового хранения, расширения файлов. Стандартные подпрограммы обработки данных. MsOffice. | Подтверждение умений и навыков работы со следующим ПО: приложениями для работы с текстом (Word), таблицами (Excel), структурированными данными (Access) и презентациями (PowerPoint). | 4 | ОПК-7, ОПК-9 |
| | Итого | 4 | |
| Итого за семестр | | 18 | |
| 2 семестр | | | |
| 6 Работа с символьными массивами. Потоковый ввод-вывод. Динамические строки. | Научиться работать со строками. Освоить способ динамического захвата и освобождения памяти под символьные массивы. Уяснить разницу между понятиями «размер массива символов» и «длина строки» при помощи функций sizeof и strlen. Освоить функции библиотеки «string.h». Научится переводить числовые данные в строковые и обратно. | 4 | ОПК-6, ОПК-7 |
| | Итого | 4 | |

Согласована на портале № 46394

| | | | |
|---|---|---|---|
| 7 Файловая система. Программная реализация. Дескриптор файла. | Операционная система. Файловые таблицы. Обращение к оборудованию ПК. BIOS. Setup BIOS. Драйверы. Работа с аудио-, видео- и графической информацией. Кодер/декодер. | 4 | ОПК-6, ОПК-9 |
| | Итого | 4 | |
| 8 Структурированные типы данных, битовые поля, объединения. Приемы программирования микроконтроллеров, битовые операции. | Работа с отдельными разрядами в байте. Использование объединения (union) и структуры "битовых полей". Размещение в статической и динамической памяти ячеек данных типов. | 2 | ОПК-7, ОПК-9 |
| | Итого | 2 | |
| 9 Понятие класса. Поля и методы класса. Спецификаторы доступа. Статические и динамические объекты класса. | Создание класса и описание его свойств и методов. Спецификаторы доступа private и public. Применение конструктора, деструктора, set- и get- методов. Работа с объектами класса: корректно создавать, инициализировать, получать значения свойств, визуализировать и удалять. | 4 | ОПК-6, ОПК-7 |
| | Итого | 4 | |
| 10 Доступ к данным. Информационная безопасность. | Информационная безопасность. Классы и их реализация. Отделение интерфейса класса от реализации класса в соответствующих *.h и *.cpp файлах, корректное подключение их к проекту, использование "препроцессорной обертки". | 4 | ОПК-6, ОПК-9 |
| | Итого | 4 | |
| Итого за семестр | | 18 | |
| Итого | | 36 | |

## 9. Самостоятельная работа

Виды самостоятельной работы, трудоемкость и формируемые компетенции представлены в таблице 9.1.

Таблица 9.1 – Виды самостоятельной работы, трудоемкость и формируемые компетенции

| Названия разделов | Виды самостоятельной работы | Трудоемкость, ч | Формируемые компетенции | Формы контроля |
|---|---|---|---|---|
| 1 семестр | | | | |
| 1 Средства обработки информационных ресурсов. Понятие алгоритма. Обращение к данным. Алгоритмические конструкции. Блок-схемы. Операторы. | Подготовка к практическим занятиям, семинарам | 4 | ОПК-6, ОПК-7, ОПК-9 | Конспект самоподготовки, Отчет по индивидуальному заданию, Тест |
| | Проработка лекционного материала | 4 | | |
| | Итого | 8 | | |
| 2 Обзор | Подготовка к практиче- | 4 | ОПК-6, ОПК- | Конспект самопод- |

Согласована на портале № 46394

| | | | | |
|---|---|---|---|---|
| вычислительных математических пакетов. Принципы обращения к ячейкам памяти. Адрес переменной. Указатели. | ским занятиям, семинарам | | 7, ОПК-9 | готовки, Отчет по индивидуальному заданию, Тест |
| | Проработка лекционного материала | 4 | | |
| | Оформление отчетов по лабораторным работам | 4 | | |
| | Итого | 12 | | |
| 3 Подпрограммы: заголовок, описание, реализация алгоритма, вызов подпрограммы, передача параметров в тело функции. | Подготовка к практическим занятиям, семинарам | 4 | ОПК-6, ОПК-7, ОПК-9 | Конспект самоподготовки, Отчет по индивидуальному заданию, Тест |
| | Проработка лекционного материала | 4 | | |
| | Оформление отчетов по лабораторным работам | 4 | | |
| | Итого | 12 | | |
| 4 Массивы. Статические и динамические объекты, адресация. Одномерные и двумерные массивы. | Подготовка к практическим занятиям, семинарам | 4 | ОПК-6, ОПК-9, ОПК-7 | Конспект самоподготовки, Отчет по индивидуальному заданию, Тест |
| | Проработка лекционного материала | 4 | | |
| | Оформление отчетов по лабораторным работам | 4 | | |
| | Итого | 12 | | |
| 5 Возможности представления информации. Типы файлового хранения, расширения файлов. Стандартные подпрограммы обработки данных. MsOffice. | Подготовка к практическим занятиям, семинарам | 6 | ОПК-6, ОПК-7, ОПК-9 | Конспект самоподготовки, Отчет по индивидуальному заданию, Тест |
| | Проработка лекционного материала | 4 | | |
| | Итого | 10 | | |
| Итого за семестр | | 54 | | |
| | Подготовка и сдача экзамена | 36 | | Экзамен |
| 2 семестр | | | | |
| 6 Работа с символьными массивами. Потоковый ввод-вывод. Динамические строки. | Подготовка к практическим занятиям, семинарам | 6 | ОПК-6, ОПК-9, ОПК-7 | Конспект самоподготовки, Отчет по индивидуальному заданию, Тест |
| | Проработка лекционного материала | 4 | | |
| | Итого | 10 | | |
| 7 Файловая система. | Подготовка к практическим занятиям, семина- | 4 | ОПК-6, ОПК-7, ОПК-9 | Конспект самоподготовки, Отчет по |

Согласована на портале № 46394

| | | | | |
|---|---|---|---|---|
| Программная реализация. Дескриптор файла. | рам | | | индивидуальному заданию, Тест |
| | Проработка лекционного материала | 4 | | |
| | Оформление отчетов по лабораторным работам | 4 | | |
| | Итого | 12 | | |
| 8 Структурированные типы данных, битовые поля, объединения. Приемы программирования микроконтроллеров, битовые операции. | Подготовка к практическим занятиям, семинарам | 4 | ОПК-6, ОПК-7, ОПК-9 | Конспект самоподготовки, Отчет по индивидуальному заданию, Тест |
| | Проработка лекционного материала | 4 | | |
| | Оформление отчетов по лабораторным работам | 4 | | |
| | Итого | 12 | | |
| 9 Понятие класса. Поля и методы класса. Спецификаторы доступа. Статические и динамические объекты класса. | Подготовка к практическим занятиям, семинарам | 4 | ОПК-6, ОПК-7, ОПК-9 | Конспект самоподготовки, Отчет по индивидуальному заданию, Тест |
| | Проработка лекционного материала | 4 | | |
| | Оформление отчетов по лабораторным работам | 4 | | |
| | Итого | 12 | | |
| 10 Доступ к данным. Информационная безопасность. | Подготовка к практическим занятиям, семинарам | 4 | ОПК-6, ОПК-9, ОПК-7 | Конспект самоподготовки, Отчет по индивидуальному заданию, Тест |
| | Проработка лекционного материала | 4 | | |
| | Итого | 8 | | |
| Итого за семестр | | 54 | | |
| | Подготовка и сдача экзамена | 36 | | Экзамен |
| Итого | | 180 | | |

## 10. Курсовой проект / курсовая работа

Не предусмотрено РУП.

## 11. Рейтинговая система для оценки успеваемости обучающихся

### 11.1. Балльные оценки для элементов контроля

Таблица 11.1 – Балльные оценки для элементов контроля

| Элементы учебной деятельности | Максимальный балл на 1-ую КТ с начала семестра | Максимальный балл за период между 1КТ и 2КТ | Максимальный балл за период между 2КТ и на конец семестра | Всего за семестр |
|---|---|---|---|---|
| 1 семестр | | | | |
| Конспект самоподготовки | 5 | 5 | 5 | 15 |

Согласована на портале № 46394

| | | | | |
|---|---|---|---|---|
| Опрос на занятиях | 5 | 5 | 5 | 15 |
| Отчет по индивидуальному заданию | | 5 | 5 | 10 |
| Отчет по лабораторной работе | | 5 | 5 | 10 |
| Отчет по практическому занятию | | 5 | 5 | 10 |
| Тест | 5 | | 5 | 10 |
| Итого максимум за период | 15 | 25 | 30 | 70 |
| Экзамен | | | | 30 |
| Нарастающим итогом | 15 | 40 | 70 | 100 |
| 2 семестр | | | | |
| Конспект самоподготовки | 5 | 5 | 5 | 15 |
| Опрос на занятиях | 5 | 5 | 5 | 15 |
| Отчет по индивидуальному заданию | | 5 | 5 | 10 |
| Отчет по лабораторной работе | | 5 | 5 | 10 |
| Отчет по практическому занятию | | 5 | 5 | 10 |
| Тест | 5 | | 5 | 10 |
| Итого максимум за период | 15 | 25 | 30 | 70 |
| Экзамен | | | | 30 |
| Нарастающим итогом | 15 | 40 | 70 | 100 |

### 11.2. Пересчет баллов в оценки за контрольные точки

Пересчет баллов в оценки за контрольные точки представлен в таблице 11.2.

Таблица 11.2 – Пересчет баллов в оценки за контрольные точки

| Баллы на дату контрольной точки | Оценка |
|---|---|
| ≥ 90% от максимальной суммы баллов на дату КТ | 5 |
| От 70% до 89% от максимальной суммы баллов на дату КТ | 4 |
| От 60% до 69% от максимальной суммы баллов на дату КТ | 3 |
| < 60% от максимальной суммы баллов на дату КТ | 2 |

### 11.3. Пересчет суммы баллов в традиционную и международную оценку

Пересчет суммы баллов в традиционную и международную оценку представлен в таблице 11.3.

Таблица 11.3 – Пересчет суммы баллов в традиционную и международную оценку

| Оценка (ГОС) | Итоговая сумма баллов, учитывает успешно сданный экзамен | Оценка (ECTS) |
|---|---|---|
| 5 (отлично) (зачтено) | 90 - 100 | A (отлично) |

Согласована на портале № 46394

| | | |
|---|---|---|
| 4 (хорошо) (зачтено) | 85 - 89 | B (очень хорошо) |
| | 75 - 84 | C (хорошо) |
| | 70 - 74 | D (удовлетворительно) |
| 3 (удовлетворительно) (зачтено) | 65 - 69 | |
| | 60 - 64 | E (посредственно) |
| 2 (неудовлетворительно) (не зачтено) | Ниже 60 баллов | F (неудовлетворительно) |

## 12. Учебно-методическое и информационное обеспечение дисциплины

### 12.1. Основная литература

1. Михальченко Сергей Геннадьевич Информационное обеспечение задач расчета и проектирования электрических схем [Электронный ресурс]: моногр. / С. Г. Михальченко. – Томск: Изд-во Томск. гос. ун-та систем упр. и радиоэлектроники, 2016. – 200 с. : ил., табл. – Библиогр.: с. 192. — Режим доступа: http://ie.tusur.ru/docs/msg/io.rar (дата обращения: 16.09.2018).

### 12.2. Дополнительная литература

1. Информатика. Базовый курс: Учебник для вузов / С. В. Симонович [и др.] ; ред. С. В. Симонович. – 2-е изд. – СПб. : Питер, 2007. – 639 с. (наличие в библиотеке ТУСУР - 57 экз.)
2. Компьютерные технологии в науке, образовании и производстве электронной техники / Кручинин В.В., Тановицкий Ю.Н., Хомич С.Л. – Томск, 2012. – 154 с. [Электронный ресурс]: — Режим доступа: http://edu.tusur.ru/training/publications/967 (дата обращения: 16.09.2018).
3. Саликаев, Юрий Рафаэльевич. Компьютерное моделирование и проектирование [Электронный ресурс]: учебное пособие / Ю. Р. Саликаев; Томский государственный университет систем управления и радиоэлектроники. – Электрон. текстовые дан. – Томск: [б. и.], 2012. – on-line, 94 с. — Режим доступа: http://edu.tusur.ru/training/publications/2548 (дата обращения: 16.09.2018).
4. Информационные технологии. Часть 2. Профессиональные математические пакеты [Электронный ресурс]: Руководство по организации самостоятельной работы студентов / С. Г. Михальченко; Томск. гос. ун-т сист. упр. и радиоэлектроники, Кафедра промышленной электроники – Томск: ТУСУР, 2016. – 128 с. : ил., табл., прил. – Библиогр.: с. 97. — Режим доступа: http://ie.tusur.ru/docs/msg/it_2.rar (дата обращения: 16.09.2018).

### 12.3. Учебно-методические пособия

#### 12.3.1. Обязательные учебно-методические пособия

1. Информационные технологии. Часть 1. Пограммирование на C++ [Электронный ресурс]: Руководство по организации самостоятельной работы студентов / С. Г. Михальченко; Томск. гос. ун-т сист. упр. и радиоэлектроники, Кафедра промышленной электроники – Томск: ТУСУР, 2016. – 162 с. : ил., табл., прил. – Библиогр.: с. 162. — Режим доступа: http://ie.tusur.ru/docs/msg/it_1.rar (дата обращения: 16.09.2018).
2. Информационные технологии. Часть 1. Программирование на C++ [Электронный ресурс]: Учебное пособие по практическим работам / С. Г. Михальченко; Томский государственный университет систем управления и радиоэлектроники, Кафедра промышленной электроники – Томск: ТУСУР, 2017. – 52 с. : ил., табл. – Библиогр.: с. 42. — Режим доступа: http://ie.tusur.ru/docs/msg/it_1.rar (дата обращения: 16.09.2018).
3. Основы информационных технологий [Электронный ресурс]: Учебно-методическое пособие по лабораторным работам / Исакова А. И. - 2017. 83 с. — Режим доступа: https://edu.tusur.ru/publications/7102 (дата обращения: 16.09.2018).

#### 12.3.2. Учебно-методические пособия для лиц с ограниченными возможностями здоровья и инвалидов

Учебно-методические материалы для самостоятельной и аудиторной работы обучающихся из числа лиц с ограниченными возможностями здоровья и инвалидов предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации.

**Для лиц с нарушениями зрения:**

Согласована на портале № 46394

– в форме электронного документа;
– в печатной форме увеличенным шрифтом.

**Для лиц с нарушениями слуха:**
– в форме электронного документа;
– в печатной форме.

**Для лиц с нарушениями опорно-двигательного аппарата:**
– в форме электронного документа;
– в печатной форме.

### 12.4. Профессиональные базы данных и информационные справочные системы

1. https://lib.tusur.ru/ru/resursy/bazy-dannyh
2. http://protect.gost.ru/
3. https://elibrary.ru/defaultx.asp

## 13. Материально-техническое обеспечение дисциплины и требуемое программное обеспечение

### 13.1. Общие требования к материально-техническому и программному обеспечению дисциплины

#### 13.1.1. Материально-техническое и программное обеспечение для лекционных занятий

Для проведения занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации используется учебная аудитория с количеством посадочных мест не менее 22-24, оборудованная доской и стандартной учебной мебелью. Имеются демонстрационное оборудование и учебно-наглядные пособия, обеспечивающие тематические иллюстрации по лекционным разделам дисциплины.

#### 13.1.2. Материально-техническое и программное обеспечение для практических занятий

Лаборатория компьютерных сетей и промышленной автоматизации / Лаборатория (ГПО) / Компьютерный класс

учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий лабораторного типа, учебная аудитория для проведения занятий семинарского типа, помещение для курсового проектирования (выполнения курсовых работ)

634034, Томская область, г. Томск, Вершинина улица, д. 74, 338 ауд.

Описание имеющегося оборудования:
- Персональные компьютеры (13 шт.);
- Стенды «Промышленная электроника» Деконт-182 (7 шт.);
- Комплект имитаторов сигналов(7 шт.);
- Коммутатор 3COM SuperStackSwitch 4226T;
- Коммутатор 3COM SuperStack-3 Switch 3226;
- Коммутационный шкаф с патч-панелями;
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:
– Google Chrome
– LibreOffice
– Microsoft Visio 2010
– PTC Mathcad13, 14
– Visual Studio
– Windows XP

Вычислительная лаборатория / Компьютерный класс

учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий лабораторного типа, учебная аудитория для проведения занятий семинарского типа, помещение для курсового проектирования (выполнения курсовых работ)

Согласована на портале № 46394

634034, Томская область, г. Томск, Вершинина улица, д. 74, 301б ауд.
Описание имеющегося оборудования:
- Персональные компьютеры (16 шт.);
- Интерактивная доска – «Smart-board» DViT (1 шт.);
- Мультимедийный проектор NEC (1 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:
- Far Manager
- Google Chrome
- LibreOffice
- PTC Mathcad13, 14
- Visual Studio
- Windows XP Pro

Вычислительная лаборатория
учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий лабораторного типа, учебная аудитория для проведения занятий семинарского типа, помещение для курсового проектирования (выполнения курсовых работ)
634034, Томская область, г. Томск, Вершинина улица, д. 74, 201б ауд.
Описание имеющегося оборудования:
- Персональные компьютеры (16 шт.);
- Интерактивная доска – «Smart-board» DViT (1 шт.);
- Мультимедийный проектор NEC (1 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:
- Google Chrome
- LibreOffice
- Microsoft Visual Studio
- PTC Mathcad13, 14
- Windows XP

**13.1.3. Материально-техническое и программное обеспечение для лабораторных работ**

Лаборатория компьютерных сетей и промышленной автоматизации / Лаборатория (ГПО) / Компьютерный класс
учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий лабораторного типа, учебная аудитория для проведения занятий семинарского типа, помещение для курсового проектирования (выполнения курсовых работ)
634034, Томская область, г. Томск, Вершинина улица, д. 74, 338 ауд.
Описание имеющегося оборудования:
- Персональные компьютеры (13 шт.);
- Стенды «Промышленная электроника» Деконт-182 (7 шт.);
- Комплект имитаторов сигналов(7 шт.);
- Коммутатор 3COM SuperStackSwitch 4226T;
- Коммутатор 3COM SuperStack-3 Switch 3226;
- Коммутационный шкаф с патч-панелями;
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:
- Google Chrome
- LibreOffice
- Microsoft Visio 2010

Согласована на портале № 46394

– PTC Mathcad13, 14
– Visual Studio
– Windows XP

Вычислительная лаборатория
учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий лабораторного типа, учебная аудитория для проведения занятий семинарского типа, помещение для курсового проектирования (выполнения курсовых работ)
634034, Томская область, г. Томск, Вершинина улица, д. 74, 201б ауд.
Описание имеющегося оборудования:
- Персональные компьютеры (16 шт.);
- Интерактивная доска – «Smart-board» DViT (1 шт.);
- Мультимедийный проектор NEC (1 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.
Программное обеспечение:
– Far Manager
– Google Chrome
– LibreOffice
– Microsoft Visual Studio
– PTC Mathcad13, 14
– Windows XP

Вычислительная лаборатория / Компьютерный класс
учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий лабораторного типа, учебная аудитория для проведения занятий семинарского типа, помещение для курсового проектирования (выполнения курсовых работ)
634034, Томская область, г. Томск, Вершинина улица, д. 74, 301б ауд.
Описание имеющегося оборудования:
- Персональные компьютеры (16 шт.);
- Интерактивная доска – «Smart-board» DViT (1 шт.);
- Мультимедийный проектор NEC (1 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.
Программное обеспечение:
– 7-Zip
– Google Chrome
– LibreOffice
– PTC Mathcad13, 14
– Visual Studio
– Windows XP Pro

**13.1.4. Материально-техническое и программное обеспечение для самостоятельной работы**

Для самостоятельной работы используются учебные аудитории (компьютерные классы), расположенные по адресам:
- 634050, Томская область, г. Томск, Ленина проспект, д. 40, 233 ауд.;
- 634045, Томская область, г. Томск, ул. Красноармейская, д. 146, 201 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 47, 126 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 74, 207 ауд.

Состав оборудования:
- учебная мебель;
- компьютеры класса не ниже ПЭВМ INTEL Celeron D336 2.8ГГц. - 5 шт.;

Согласована на портале № 46394

- компьютеры подключены к сети «Интернет» и обеспечивают доступ в электронную информационно-образовательную среду университета.

Перечень программного обеспечения:
- Microsoft Windows;
- OpenOffice;
- Kaspersky Endpoint Security 10 для Windows;
- 7-Zip;
- Google Chrome.

### 13.2. Материально-техническое обеспечение дисциплины для лиц с ограниченными возможностями здоровья и инвалидов

Освоение дисциплины лицами с ограниченными возможностями здоровья и инвалидами осуществляется с использованием средств обучения общего и специального назначения.

При занятиях с обучающимися **с нарушениями слуха** предусмотрено использование звукоусиливающей аппаратуры, мультимедийных средств и других технических средств приема/передачи учебной информации в доступных формах, мобильной системы преподавания для обучающихся с инвалидностью, портативной индукционной системы. Учебная аудитория, в которой занимаются обучающиеся с нарушением слуха, оборудована компьютерной техникой, аудиотехникой, видеотехникой, электронной доской, мультимедийной системой.

При занятиях с обучающимися **с нарушениями зрениями** предусмотрено использование в лекционных и учебных аудиториях возможности просмотра удаленных объектов (например, текста на доске или слайда на экране) при помощи видеоувеличителей для комфортного просмотра.

При занятиях с обучающимися **с нарушениями опорно-двигательного аппарата** используются альтернативные устройства ввода информации и другие технические средства приема/передачи учебной информации в доступных формах, мобильной системы обучения для людей с инвалидностью.

## 14. Оценочные материалы и методические рекомендации по организации изучения дисциплины

### 14.1. Содержание оценочных материалов и методические рекомендации

Для оценки степени сформированности и уровня освоения закрепленных за дисциплиной компетенций используются оценочные материалы в составе:

#### 14.1.1. Тестовые задания

1. Что будет выведено на экран после выполнения следующего программного кода:
float y= 123.4567890; printf("x=%10.3f",y);
Варианты ответов:
• x=123,457
• y=123,4567890
• x=123,4567890
• y=123,456
2. Что будет выведено на экран после выполнения следующего программного кода:
float x= 123.4; cout << "y=" << (int)x/2;
Варианты ответов:
• y=61
• y=61.7
• x=61
• y=123.2
3. Что будет выведено на экран после выполнения следующего программного кода:
char a=65; a= a++; cout << a << endl;
если ASCII-код символа "A" равен 65?
Варианты ответов:
• B
• a=65endl

Согласована на портале № 46394

• a=66
• a=A

4. Что будет выведено на экран после выполнения следующего программного кода:
int A[10];
for(int i=0; i<10; i++) A[i]=i;
cout << A[5] << endl;
Варианты ответов:
• 5
• 4
• i
• 10

5. Что будет выведено на экран после выполнения следующего программного кода:
int A[10];
for(int i=0; i<10; i++) A[i]=10-i;
int *P= A; cout << *(P+2) << endl;
Варианты ответов:
• 8
• 9
• 1
• 2

6. Что будет выведено на экран после выполнения следующего программного кода:
int a= 12; int *b=&a; cout << *b+1 << endl;
если переменная а лежит по адресу 0012FF60?
Варианты ответов:
• 13
• 12
• 0012FF60
• 0012FF64

7. Что будет выведено на экран после выполнения следующего программного кода:
int a= 12; int *b=&a; cout << b+1 << endl;
если переменная а лежит по адресу 0012FEA0?
Варианты ответов:
• 0012FEA4
• 12
• 0012FEA0
• 13

8. Что будет выведено на экран после выполнения следующего программного кода:
float *x=(float*)calloc(10,sizeof(float));
for(int i=0; i<10; i++) *(x+i)=10-i;
printf("x=%f\n", *(x+4));
Варианты ответов:
• x=6,000000
• x=6
• 4
• x=4,000000

9. Что будет выведено на экран после выполнения следующего программного кода:
double x= 12.3; double *y=&x;
y=y+1; cout << *(y-1)-1 << endl;
если переменная а лежит по адресу 0012FF6E?
Варианты ответов:
• 11,3
• 12,3
• 0012FF6E
• 0012FF6A

Согласована на портале № 46394

10. Что будет выведено на экран после выполнения следующего программного кода:
int a=12; a--;
if (a%2!=0) cout << "a%2" << endl;
else cout << "a%2-1" << endl;
Варианты ответов:
• a%2
• 1
• a%2-1
• 5
11. Какой смысл имеет приведенный ниже программный код?
int a=0; int b=0;
while (a<10) { b+=a; a++; }
cout << b;
Варианты ответов:
• вычисление суммы целых чисел от 0 до 9
• вычисление чисел, меньших 10
• вычисление суммы целых чисел от 0 до 10
• вычисление среднего значения 10 целых чисел
12. Сколько раз выполнится тело цикла в приведенном ниже программном коде?
int a=6;
do { cout << a; a++; } while (a<10);
Варианты ответов:
• 4
• 9
• 6789
• 5
13. Что будет выведено на экран после выполнения следующего программного кода:
int a= 12; while (a<25) { a++; } cout << a;
Варианты ответов:
• 25
• 24
• 12
• 13
14. Какой смысл имеет приведенный ниже программный код?
int a=12; int b=13;
if (a<b) cout << a; else cout << b;
Варианты ответов:
• вычисление наименьшего из чисел
• вычисление среднего значения двух чисел
• вычисление наибольшего из чисел
• вычисление максимума
15. Что будет выведено на экран после выполнения следующего программного кода:
int *x= new int[5]; int i=0;
for(; i<5; i++) { x[i]=i; cout << x[i]; }
Варианты ответов:
• 01234
• 0123456789
• 012345
• 12345
16. К какому типу операторов относится запись:
do i++; while( i<100 );
Варианты ответов:
• оператор цикла
• оператор выбора

Согласована на портале № 46394

• оператор ввода-вывода
• оператор присваивания

17. Что будет выведено на экран после выполнения следующего программного кода:
int y=14; f(y); cout << y; y=16;
Если функция f задана следующим образом:
void f(int x) { x++; }
Варианты ответов:
• 14
• 15
• 16
• 17

18. Что будет выведено на экран после выполнения следующего программного кода:
int y=14; f(y); cout << y; y=16;
Если функция f задана следующим образом:
void f(int &x) { x++; }
Варианты ответов:
• 15
• 17
• 16
• 14

19. Чему будет равен элемент y[2][1] матрицы y?
int y[4][4]; int *p=&y[0][0];
for(int i=0; i<16; i++) { *(p+i)= i; }
cout << y[2][1] ;
Варианты ответов:
• 9
• 6
• 1
• 4

20. При помощи какого оператора освобождается динамическая память, если выделение памяти производилось оператором new?
int *y = new(int)
Варианты ответов:
• delete y;
• y= (int*)malloc(sizeof(int));
• free(y);
• y= (int*)calloc(1, sizeof(int));

21. Как подключить фрагменты программного кода, библиотеки и заголовочные файлы класса к своей программе?
Варианты ответов:
• используя директиву #include
• используя директиву #define
• используя файл описания *.h (header)
• используя файловую переменную FILE*

### 14.1.2. Экзаменационные вопросы

Типовые вопросы:

1. Что представляет собой символьная строка?
2. Как используются стандартные функции форматного и потокового ввода-вывода в работе со строками?
3. Как обозначается конец строки, для чего он нужен?
4. Как производится присваивание (копирование) строковых переменных?
5. Как определить количество символов в массиве? В строке?
6. Можно ли обращаться к элементам строки при помощи оператора «квадратные скобки», а по технологии «адрес + смещение»?

Согласована на портале № 46394

7. Что такое адрес строки? Как сравнить две строковых переменных?

8. Что такое текстовый файл? Опишите последовательность записи строки в файл потоковым способом.

9. Что представляют собой файловые переменные типа FILE из библиотеки <stdio.h> как с ними работать? Какое им присваивается значение?

10. Что представляют собой объекты файлового ввода-вывода <ifstream> и <ofstream> из библиотеки <iostream>? Как с ними работать?

11. Опишите последовательность записи числа в файл форматным способом, используя переменные типа FILE из библиотеки <stdio.h>.

12. В чем разница использования метода getline() и оператора >> при потоковом вводе данных из файла – подпрограмм класса <ifstream>.

13. Возможно ли в полях структуры размещать данные различных типов?

14. Как выделяется память под динамическую переменную структурного типа, как освобождается?

15. Как размещаются в переменной структурного типа ячейки полей структуры?

16. Как осуществляется обращение к полям статической переменной структурного типа?

17. Что такое объединение (union), как размещаются поля в переменных такого типа данных?

18. Что такое указатель на структуру, как с ним работать, как обращаться к полям структурной переменной «по адресу?»

19. Как осуществляется обращение к полям динамической переменной структурного типа?

20. Как запрограммировать побитное обращение к переменным целых типов.

21. Как размещаются в памяти поля переменной типа union?

22. Как работает двоичная побитная операция | «или», что получится в результате вычисления 23|112?

23. Как работает двоичная побитная операция ^ «исключающее или», что получится в результате 103^112?

24. Что такое битовые поля и как с ними работать?

25. Что такое класс? Для чего он используется, в чем его особенности, преимущества?

26. Спецификатор доступа private – для чего он применяется?

27. Что называется полями и методами класса? Что такое объект класса?

28. В чем смысл технологии отделения интерфейса от реализации методов класса?

29. Когда вызывается деструктор класса для статических и динамических объектов?

30. Как организовано в классах ограничение и разрешение доступа к полям и методам?

31. Конструктор и деструктор – для чего они нужны? Когда используются?

32. Спецификатор public – как он используется в конструировании класса?

33. Оператор принадлежности к классу («двойное двоеточие», «::») – для чего и как он применяется? Можно ли обойтись без него?

34. Что такое set- и get- методы? В каком случае они необходимы?

35. Что такое заголовочный файл и чем он отличается от файла реализации методов? Как их использовать в проекте?

36. Что такое спецификатор доступа friend? Как он используется?

37. Что представляют собой операторы? Можно ли создавать операторы для собственных классов?

38. В чем смысл наследования? Для чего применяется спецификатор protected?

39. Как наследуются конструкторы и деструкторы? Что такое абстрактный класс?

40. Что такое дружественные классы и функции?

Типовые вопросы:

1. Что представляет собой программа на языке С++?

2. Как задается переход на новую строку при выводе на экран?

3. Как использовать функции потокового ввода-вывода?

4. Как в форматном спецификаторе указать, в скольких ячейках выводить число?

5. Чем отличаются операторы выбора от операторов цикла?

6. Что такое «пошаговая трассировка программного кода»? Для чего она используется?

Согласована на портале № 46394

7. Сколько раз выполняется оператор, содержащийся в конструкции if ?

8. Как определить, сколько раз будет выполнен оператор, содержащийся в конструкции for?

9. Для чего используется оператор break? Что будет, если его применить в теле цикла?

10. Какие значения могут принимать логические переменные? Можно ли их задавать числами? Какой объем памяти они занимают?

11. Чем отличается цикл с предусловием от цикла с постусловием? Приведите пример.

12. Что такое тернарная логическая операция? Какова семантика её применения?

13. Как используется оператор break? Что будет, если его пропустить в одном из case-блоков оператора switch?

14. Что такое указатель и для чего он используется?

15. Как узнать значение переменной, если известен только ее адрес?

16. Какие действия выполняет оператор new?

17. В чем разница между динамической переменной и указателем на нее?

18. Какое значение возвращают функции malloc и calloc при выделении памяти?

19. Как получить адрес переменной?

20. Что такое адрес переменной?

21. Какой размер занимает переменная-указатель?

22. Что такое динамическая переменная, чем она отличается от статической?

23. В каком случае нужно пользоваться оператором delete, а в каком – функцией free( ) для освобождения памяти?

24. Как происходит вызов подпрограммы и возвращение из нее в языке C++?

25. Почему нельзя изменить переменные вызывающей функции, передаваемые в вызываемую подпрограмму по значению?

26. Как реализована передача значений переменных вызывающей функции в тело вызываемой подпрограммы по указателю? Изменяется ли значение указателя при этом?

27. Что такое точка возврата из функции, где она находится?

28. Как передать фактические параметры в подпрограмму по ссылке? Изменятся ли их значения в вызывающей программе?

29. Для чего нужна команда return? Какой тип данных должна иметь подпрограмма, не возвращающая никакого значения?

30. Что такое динамический массив, как выделять под него память?

31. Как располагаются в памяти элементы массива?

32. Что такое статический массив, какие есть два основных способа обращения к элементам массива?

33. Можно ли хранить в ячейках одномерного массива переменные разных типов?

34. Какие операторы и функции используются для выделения памяти под динамический массив?

35. Как определить адреса статического и динамического массивов, где они хранятся?

36. Как размещается в памяти статический двумерный массив?

37. Как выделяется память под матрицу, хранящуюся в виде одномерного массива?

38. Как обращаться к элементам с координатами i и j матрицы, хранящейся в виде одномерного массива?

39. Как передается в подпрограмму статический двумерный массив?

40. Как размещается в памяти двумерный динамический массив?

### 14.1.3. Темы опросов на занятиях

Семестр 1.

Для чего нужна строка форматной спецификации, как она выглядит?

Что такое библиотеки и как их подключить к своей программе?

Что представляет собой программа на языке C++?

Чем отличаются операторы выбора от операторов цикла?

Как использовать функции форматного ввода-вывода?

Что хранится в ячейке символьного типа (char)? Как это значение выводится на экран?

Как организовать бесконечный цикл?

Как увидеть текущее значение переменной во время выполнения программы?

Согласована на портале № 46394

Что такое счетчик цикла? Как его использовать в операторе for?

Сколько раз выполняется оператор, содержащийся в конструкции if-else ?

В чем особенность оператора continue? Как его применяют?

Каковы правила использования оператора множественного выбора switch? Для чего нужен блок default?

Имеет ли статическая переменная адрес?

Что такое указатель и для чего он используется?

Как получить адрес переменной?

Какое имя можно задать динамической переменной?

Какие пять действий происходят при выделении памяти функциями malloc, calloc?

Что такое динамическая переменная, чем она отличается от статической?

В каком случае нужно пользоваться оператором delete, а в каком – функцией free( ) для освобождения памяти?

Как происходит вызов подпрограммы и возвращение из нее в языке C++?

Что такое перегрузка функций? Составьте пример перегруженной функции.

Что такое точка возврата из функции, где она находится?

Для чего нужна команда return? Какой тип данных должна иметь подпрограмма, не возвращающая никакого значения?

Почему нельзя изменить переменные вызывающей функции, передаваемые в вызываемую подпрограмму по значению?

Что такое динамический массив, как выделять под него память?

Как скопировать один массив в другой?

Какие операторы и функции применяются для освобождения динамической памяти?

Как располагаются в памяти элементы массива?

Можно ли хранить в ячейках одномерного массива переменные разных типов?

Какие операторы и функции используются для выделения памяти под динамический массив?

Являются ли эти записи идентичными для одномерного массива A: &(A[i]) или A+i?

Задана матрица N×M, при такой размерности N – это число строк или столбцов?

Как используется вложенный цикл при работе с матрицами?

Чем фактически являются строки любой – статической или динамической матрицы?

Семестр 2.

Что представляет собой символьная строка?

Размер символьного массива и длина строки – это одно и то же?

Как обозначается конец строки, для чего он нужен?

Какие операторы и функции служат для освобождения памяти, занятой динамическим символьным массивом?

Может ли строка символов быть статической или динамической?

Можно ли обращаться к элементам строки при помощи оператора «квадратные скобки», а по технологии «адрес + смещение»?

Какие функции служат для перевода числа в символьную строку, какие – для перевода символьной строки в число? Что возвращается, если перевод сделать не удалось?

Что такое текстовый файл? Опишите последовательность записи строки в файл потоковым способом.

С точки зрения механизма размещения двоичных данных в памяти компьютера имеются ли особенности хранения файлов различного назначения?

Опишите логику последовательности чтения отдельного слова из файла для файловых переменных (FILE) из библиотеки <stdio.h>.

Что представляют собой объекты файлового ввода-вывода <ifstream> и <ofstream> из библиотеки <iostream>? Как с ними работать?

В чем разница использования метода getline() и оператора >> при потоковом вводе данных из файла – подпрограмм класса <ifstream>.

Что такое структурный тип данных и как им пользоваться?

Возможно ли в полях структуры размещать данные различных типов? Можно ли указывать,

Согласована на портале № 46394

сколько бит выделять под каждое поле структуры?

Как осуществляется обращение к полям статической и динамической переменной структурного типа?

Как передается в подпрограмму статическая переменная структурного типа?

Что такое битовые поля и как с ними работать?

Как размещаются в памяти поля переменной типа union?

Как работает двоичная побитная операция & «и», что получится в результате вычисления 23&126?

Как работает побитная операция «сдвиг», что получится в результате 23>>2?

Объединение содержит три поля unsigned short X, double Z и char Y, какой размер будет занимать переменная этого типа?

Что такое класс? Для чего он используется, в чем его особенности, преимущества?

В чем различие понятий класс и объект класса?

Может ли быть в классе несколько конструкторов, деструкторов, set- и get-методов?

Как организовано в классах ограничение и разрешение доступа к полям и методам?

Конструктор и деструктор – для чего они нужны? Когда используются?

Как реализовано обращение к полям и методам статических и динамических объектов класса?

Что такое унарные и бинарные операторы? Что такое приоритет операций?

Какие спецификаторы доступа вы знаете? В чем их особенность?

Можно прописать реализацию методов класса в самом теле класса, а можно – в другом месте и даже в другом файле. В чем различие?

Что такое «препроцессорная обертка»? Как это применяется?

В чем смысл наследования? Для чего применяется спецификатор protected?

Как наследуются конструкторы и деструкторы? Что такое абстрактный класс?

Для чего применяются виртуальные методы?

Объединение содержит три поля unsigned short X, double Z и char Y, какой размер будет занимать переменная этого типа?

### 14.1.4. Темы индивидуальных заданий

1. Создать проект на Visual C++, написать простейшую программу на языке C++, отладить и запустить на выполнение программу. Научиться задавать переменные и константы следующих типов: целого, вещественного, символьного и строкового – в соответствии с индивидуальным заданием. Освоить два способа ввода с клавиатуры и вывода на экран значений этих переменных – форматный и потоковый способы. Изучить спецификаторы, форматы ввода-вывода и управляющие символы.

2. Изучить операторы языка C++, научиться применять их при написании программ. Научиться отслеживать значения числовых и логических переменных и выражений в процессе отладки программы. Освоить умение переводить мнемонические выражения в последовательность операторов.

3. Научиться работать со статическими и динамическими переменными. Освоить применение операций адресации и разадресации. Научиться выделять память под динамические переменные и освобождать ее при помощи операторов new и delete. Освоить обращение к динамическим объектам. Научиться выделять память под динамические переменные и освобождать ее при помощи функций calloc, malloc и free. Научиться визуализировать адреса статических и динамических объектов. Освоить функции randomize() и rand() генерирующие случайные числа.

4. Научиться создавать подпрограммы и вызывать их из тела основной функции. Освоить применение оператора return, возвращающего значение функции. Научиться передавать параметры в функцию и корректно возвращать значения – результаты вычислений из функции. Освоить передачу параметров в функцию по значению, по ссылке и по указателю. Научиться создавать прототипы функций, освоить перегрузку функций. Изучить функции библиотеки math.h. Научиться пользоваться средствами пошаговой трассировки кода и просмотра текущих значений переменных.

5. Научиться создавать массивы, выделять память под элементы массива и инициировать их значениями. Научиться обращаться к элементам массивов как при помощи оператора имя[индекс], так и посредством методики *(адрес+смещение). Освоить понятия: адрес массива, адрес элемента

Согласована на портале № 46394

массива, смещение, индекс. Освоить пошаговую трассировку программы с массивами, научиться отражать в окне watch элементы массива и их адреса. Научиться передавать массив в функцию.

6. Освоить способ динамического захвата и освобождения памяти под двумерные массивы данных (матрицы) – при помощи оператора new и посредством функций malloc( ) и calloc( ). Повторить понятия, операции и закрепить умения и навыки матричной алгебры. Уяснить практическую разницу и сходство в языке С++ понятий «массив» и «указатель». Повысить навыки отладки программного кода на С++, трассировки программы и просмотра значений переменных в окне Watch.

7. Научиться работать со строками. Освоить способ динамического захвата и освобождения памяти под символьные массивы. Уяснить разницу между понятиями «размер массива символов» и «длина строки» при помощи функций sizeof и strlen. Освоить функции библиотеки «string.h». Научится переводить числовые данные в строковые и обратно.

8. Научиться работать с файловыми переменными посредствам подпрограмм библиотек fstream.h и stdio.h. Закрепить навыки работы со строками. Научиться открывать файлы для записи и чтения, создавать и уничтожать файлы, определять конец файла.

9. Освоить умение работать со структурами данных. Научиться создавать новый комбинированный тип данных – структуру. Научиться обращаться к статическим и динамическим переменным этого типа и к массиву структур. Закрепить навыки работы с файлами и символьными строками. Освоить понятия "битовая карта" (bitmap) и "объединение" (union).

10. Освоить умение работать с разрядами в байте. Научиться изменять произвольный бит в байте, не меняя остальных. Выводить на экран двоичный код переменной произвольного типа. Повысить понимание преобразований двоичного формата.

11. Научиться создавать класс и описывать его свойства и методы. Изучить спецификаторы доступа private и public. Освоить применение конструктора, деструктора, set- и get- методов. Освоить работу с объектами класса: корректно создавать, инициализировать, получать значения свойств, визуализировать и удалять. Научиться выделять память под динамические объекты класса и освобождать ее. Освоить обращение к свойствам и методам динамических объектов.

12. Ознакомиться с понятиями информационной безопасности, с программными средствами информационной безопасности. Научиться программировать интерфейс класса и реализацию класса в соответствующих *.h и *.cpp файлах, корректно подключать их к проекту, используя препроцессорную обертку. Закрепить навыки конструирования классов, выделения памяти под динамические объекты класса, освобождения памяти, обращения к их свойствам и методам.

13. Научиться создавать классы-потомки на основе базовых классов и описывать их свойства и методы. Научиться применять конструкторы базового класса в соответствующих конструкторах наследуемых классов. Освоить умение работать с абстрактными классами, дружественными и виртуальными функциями и научиться перегружать операторы.

14. Получить компетенции в части создания динамических структур данных (списки, кольца, деревья); получить компетенции по методам доступа к данным (стек, очередь); научиться создавать унифицированный программный код (шаблоны функций, классов); освоить умения и навыки в технологии поиска и сортировки. Закрепить навыки конструирования классов, выделения памяти под динамические объекты класса, освобождения памяти, обращения к их свойствам и методам.

### 14.1.5. Вопросы на самоподготовку

Понятие информации. Предмет информатики. Информационные процессы. Информатизация общества. Компьютеры. Технология проектирования и отладки программ. Инструментальные средства программирования. Основные принципы создания программного обеспечения. Основы программирования на языке верхнего уровня (на примере С++). Процесс создания программного кода. Отладка программ. Трассировка программного кода.

Алгоритмические конструкции. Операторы выбора. Перечисляемый тип данных. Операторы цикла. Использование переменных логического типа. Оператор безусловного перехода. Организация диалога с пользователем.

Система компьютерной алгебры Mathematica. Пакет символьной математики MAPLE. Пакет математических и инженерных вычислений MATLAB. Программный комплекс MathCAD.

Статическое и динамическое распределение памяти. Функции динамического распределе-

Согласована на портале № 46394

ния памяти. Освобождение памяти. Операторы new и delete. Обращение к переменным "по имени" и "по адресу". Отличие указателя от динамической переменной.

Подпрограммы: процедуры и функции. Структура подпрограммы. Формальные и фактические параметры. Передача параметров в тело функции. Вызов подпрограммы. Возвращение значения, оператор return. Область видимости переменной.

Описание и вызов подпрограммы. Прототип подпрограммы. Адрес функции, указатель на подпрограмму. Область видимости переменной. Глобальные и локальные переменные. Перегрузка функций.

Массивы - хранилище однотипных данных. Одномерные (вектора) и многомерные (матрицы) массивы. Описание массива, обращение к элементам массива по индексу. Генератор случайных чисел. Работа с массивами в цикле.

Статические и динамические массивы. Функции calloc(), malloc() и free(). Адрес массива = адрес начальной ячейки. Указатель на массив. Передача массива в функцию. Динамические одномерные и двумерные массивы. Обращение к элементам массива "адрес+смещение".

Двумерные статические и динамические. Три способа размещения в памяти двумерных массивов. Освобождение памяти. Обращение к ячейкам двумерного массива.

Обращение к оборудованию ПК. BIOS. Setup BIOS. Драйверы. Работа с аудио-, видео- и графической информацией. Кодер/декодер. Приложения для работы с текстом (Word), таблицами (Excel), структурированными данными (Access) и презентациями (PowerPoint).

Строки символов. Строка – массив символов. Вывод строки на экран. Последний элемент строки. Указатель на строку.

Указатель на строку. Копирование и сравнение строк. Статические и динамические строки. Функции преобразования строковых типов данных.

Файловые подсистемы ОС, хранение данных на диске. Дескриптор файла в программе. Доступ к файлу, совместный доступ, транзакции. Файловые операции (связать, открыть, закрыть, читать, писать, определить конец).

Работа с файлами при помощи потоков ввода-вывода. Библиотека . Файловые операции. Поиск в файле.

Структуры. Описание в программе, доступ к полям структуры. Размещение в памяти. Указатели на структуру. Статические и динамические переменные структурного типа. Битовые поля. Объединения. Статические и динамические переменные структурного типа.

Приемы программирования микроконтроллеров, битовые операции. Поразрядные логические операции. Поразрядные операции сдвига. Обращение к разрядам при помощи битовых полей.

Введение в понятие "класс". Поля и методы класса. Спецификаторы доступа, использование set- и get- методов. Конструктор и деструктор.

Указатель на объект класса. Статические и динамические объекты класса. Конструктор и деструктор для статических и динамических объектов класса. Отделение интерфейса программы от реализации. Дружественные функции. Перегрузка операторов. Наследование.

Программный комплекс MathCAD. Структура программы. Численные и аналитические вычисления, точность вычислений. Вывод графиков. Алгоритмические возможности MathCAD. Операторы, операторные блоки. Матричные операции MathCAD. Встроенные функции.

Компьютерные сети, вопросы информационной безопасности. Локальные и глобальные компьютерные сети. Модель OSI. Адресация в сети, форматы обмена данными (сетевые протоколы). Программное обеспечение для работы в сетях. Службы сети Интернет. Угрозы информационных атак. Компьютерные вирусы.

### 14.1.6. Вопросы для подготовки к практическим занятиям, семинарам

Создать проект на Visual C++, написать простейшую программу на языке C++, отладить и запустить на выполнение программу.

Отладить программный код программы, используя точки останова и окно трассировки.

Задать переменные и константы следующих типов: целого, вещественного, символьного и строкового.

Освоить два способа ввода с клавиатуры и вывода на экран значений переменных – форматный и потоковый способы.

Изучить спецификаторы, форматы ввода-вывода и управляющие символы.

Согласована на портале № 46394

Изучить операторы языка С++, научиться применять их при написании программ.

Отслеживать значения числовых и логических переменных и выражений в процессе отладки программы.

Написать программу, использующую указатели, операции адресации и разадресации.

Переводить мнемонические выражения в последовательность операторов.

Выводить на экран адрес переменной и значение переменной для хранения адреса.

Произвести демонстрацию возможностей MathCAD, навыков отладки программы, выполнения вычислений и вывода результатов.

Продемонстрировать умение задавать переменные, константы и графики на рабочем поле MathCAD при помощи шаблонов.

Научиться оформлять блок-схему программы в соответствии с ЕСКД.

Составить дерево вызова процедур со списками формальных параметров подпрограмм.

Обратиться к подпрограмме по ссылке, по значению и по указателю.

Набрать текст (Word), создать и заполнить таблицу (Excel).

Изучить методы создания простейших баз данных (Access).

Создать презентацию (PowerPoint).

Запрограммировать работу с массивом, выводить на экран элементы массива.

Создавать и использовать динамические и статические массивы.

Создавать, заполнять и выводить на экран двумерные массивы.

Обращаться к элементам массива по индексу и по технологии «адрес+смещение».

Освоить способ динамического захвата и освобождения памяти под символьные массивы.

Уяснить разницу между понятиями «размер массива символов» и «длина строки» при помощи функций sizeof и strlen.

Освоить функции библиотеки «string.h».

Научится переводить числовые данные в строковые и обратно.

Обращаться к оборудованию ПК через BIOS. Изучить Setup BIOS.

Продемонстрировать работу с аудио-, видео- и графической информацией.

Работать с отдельными разрядами в байте, используя объединения и битовые поля.

Создать класс и описать его свойства и методы, использовать спецификаторы доступа private и public.

Написать класс, продемонстрировать применение конструктора, деструктора, set- и get- методов.

Корректно создавать, инициализировать, получать значения свойств, визуализировать и удалять статические и динамические объекты класса.

Научиться корректно отделять интерфейс класса от его реализации.

Продемонстрировать корректное подключение файлов к проекту с использованием "препроцессорной обертки".

### 14.1.7. Темы лабораторных работ

Научиться работать со статическими и динамическими переменными. Освоить применение операций адресации и разадресации. Научиться выделять память под динамические переменные и освобождать ее при помощи операторов new и delete. Освоить обращение к динамическим объектам. Научиться выделять память под динамические переменные и освобождать ее при помощи функций calloc, malloc и free. Научиться визуализировать адреса статических и динамических объектов.

Научиться создавать подпрограммы и вызывать их из тела основной функции. Освоить применение оператора return, возвращающего значение функции. Научиться передавать параметры в функцию и корректно возвращать значения – результаты вычислений из функции. Освоить передачу параметров в функцию по значению, по ссылке и по указателю. Научиться создавать прототипы функций, освоить перегрузку функций. Изучить функции библиотеки math.h. Научиться пользоваться средствами пошаговой трассировки кода и просмотра текущих значений переменных.

Научиться создавать массивы, выделять память под элементы массива и инициировать их значениями. Научиться обращаться к элементам массивов как при помощи оператора имя[индекс], так и посредством методики *(адрес+смещение). Освоить понятия: адрес массива, адрес элемента массива, смещение, индекс. Освоить пошаговую трассировку программы с массивами, научиться

Согласована на портале № 46394

отражать в окне watch элементы массива и их адреса. Научиться передавать массив в функцию.

Освоить способ динамического захвата и освобождения памяти под двумерные массивы данных (матрицы) – при помощи оператора new и посредством функций malloc( ) и calloc( ). Повторить понятия, операции и закрепить умения и навыки матричной алгебры. Уяснить практическую разницу и сходство в языке С++ понятий «массив» и «указатель». Повысить навыки отладки программного кода на С++, трассировки программы и просмотра значений переменных в окне Watch.

Научиться работать с файловыми переменными посредстваом подпрограмм библиотек fstream.h и stdio.h. Закрепить навыки работы со строками. Научиться открывать файлы для записи и чтения, создавать и уничтожать файлы, определять конец файла.

Освоить умение работать со структурами данных. Научиться создавать новый комбинированный тип данных – структуру. Научиться обращаться к статическим и динамическим переменным этого типа и к массиву структур. Закрепить навыки работы с файлами и символьными строками.

Освоить умение работать с разрядами в байте. Научиться изменять произвольный бит в байте, не меняя остальных. Выводить на экран двоичный код переменной произвольного типа. Повысить понимание преобразований двоичного формата. Применять операции поразрядного сдвига и поразрядные логические операции, а также при помощи структур объединение (union) и битовые поля.

Научиться создавать класс и описывать его свойства и методы. Изучить спецификаторы доступа private и public.Применение конструктора, деструктора, set- и get- методов. Работа с объектами класса: корректно создавать, инициализировать, получать значения свойств, визуализировать и удалять. Интерфейс и реализация класса.

### 14.1.8. Методические рекомендации

Дисциплина является обеспечивающей, формируемые компетенции требуются для освоения предметов, непосредственно относящихся к электронике и наноэлектронике, силовой электронике (power electronics) и промышленной электронике (industrial electronics). В связи с этим при проведении всех видов занятий и форм контроля необходимо заострять внимание обучающихся на практическом применении получаемых компетенций в предметной области будущей профессиональной деятельности.

## 14.2. Требования к оценочным материалам для лиц с ограниченными возможностями здоровья и инвалидов

Для лиц с ограниченными возможностями здоровья и инвалидов предусмотрены дополнительные оценочные материалы, перечень которых указан в таблице 14.

Таблица 14 – Дополнительные материалы оценивания для лиц с ограниченными возможностями здоровья и инвалидов

| Категории обучающихся | Виды дополнительных оценочных материалов | Формы контроля и оценки результатов обучения |
|---|---|---|
| С нарушениями слуха | Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы | Преимущественно письменная проверка |
| С нарушениями зрения | Собеседование по вопросам к зачету, опрос по терминам | Преимущественно устная проверка (индивидуально) |
| С нарушениями опорно-двигательного аппарата | Решение дистанционных тестов, контрольные работы, письменные самостоятельные работы, вопросы к зачету | Преимущественно дистанционными методами |
| С ограничениями по общемедицинским показаниям | Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы, устные ответы | Преимущественно проверка методами исходя из состояния обучающегося на момент проверки |

Согласована на портале № 46394

### 14.3. Методические рекомендации по оценочным материалам для лиц с ограниченными возможностями здоровья и инвалидов

Для лиц с ограниченными возможностями здоровья и инвалидов предусматривается доступная форма предоставления заданий оценочных средств, а именно:

- в печатной форме;
- в печатной форме с увеличенным шрифтом;
- в форме электронного документа;
- методом чтения ассистентом задания вслух;
- предоставление задания с использованием сурдоперевода.

Лицам с ограниченными возможностями здоровья и инвалидам увеличивается время на подготовку ответов на контрольные вопросы. Для таких обучающихся предусматривается доступная форма предоставления ответов на задания, а именно:

- письменно на бумаге;
- набор ответов на компьютере;
- набор ответов с использованием услуг ассистента;
- представление ответов устно.

Процедура оценивания результатов обучения лиц с ограниченными возможностями здоровья и инвалидов по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

**Для лиц с нарушениями зрения:**

- в форме электронного документа;
- в печатной форме увеличенным шрифтом.

**Для лиц с нарушениями слуха:**

- в форме электронного документа;
- в печатной форме.

**Для лиц с нарушениями опорно-двигательного аппарата:**

- в форме электронного документа;
- в печатной форме.

При необходимости для лиц с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения может проводиться в несколько этапов.

Согласована на портале № 46394