

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тверской государственный технический университет»
(ТвГТУ)

УТВЕРЖДАЮ
Проректор
по учебной работе
_____ Э.Ю. Майкова
« ____ » _____ 2021 г.

РАБОЧАЯ ПРОГРАММА
дисциплины обязательной части
Блока 1 «Дисциплины (модули)»
«Структуры и алгоритмы обработки данных»

Направление подготовки бакалавров 27.03.04 Управление в технических системах

Направленность (профиль) – Автоматизация и управление технологическими процессами и производствами

Типы задач профессиональной деятельности – проектно-конструкторский, сервисно-эксплуатационный

Форма обучения – очная, заочная

Факультет информационных технологий
Кафедра «Автоматизация технологических процессов»

Тверь 2021

Рабочая программа дисциплины соответствует ОХОП подготовки бакалавров в части требований к результатам обучения по дисциплине и учебному плану.

Разработчик программы: доцент кафедры АТП _____ В.Г. Васильев

Программа рассмотрена и одобрена на заседании кафедры АТП
« ____ » _____ 2021 г., протокол № ____.

Заведующий кафедрой АТП _____ Б.И. Марголис

Согласовано
Начальник учебно-методического
отдела УМУ _____ Д.А. Барчуков

Начальник отдела
комплектования
зональной научной библиотеки _____ О.Ф. Жмыхова

1. Цели и задачи дисциплины

Цель изучения дисциплины:

- сформировать у студентов базовые понятия, лежащие в основе процесса разработки вычислительных алгоритмов, применения в них типовых структур данных для решения задач разработки прикладного программного обеспечения;
- применять технологии модульного, структурного и объектно-ориентированного программирования для разработки сложных программных комплексов;
- сформировать представления и знания об основных алгоритмах обработки экспериментальных и статистических данных, используемых в них типовых структурах данных и вычислительных методах решения задач на их основе;
- научить программной реализации типовых алгоритмов, структур данных и их модификаций на языке программирования C/ C++;
- сформировать знания о сложности вычислительных алгоритмов и методах из программной реализации.

Задачами дисциплины являются:

- научить студентов навыкам и приемам практического применения стандартных конструкций языка C/C++, библиотек программных модулей, классов, визуальных компонентов среды программирования C++ Builder;
- сформировать у студентов знания о приемах написания программных приложений из курсов математического анализа, линейной алгебры, вычислительных операций с комплексными числами, статистической обработки экспериментальных данных.

2. Место дисциплины в структуре образовательной программы

Дисциплина относится к обязательной части Блока 1 ОП ВО. Для изучения курса требуются знания дисциплин «Математика», «Программирование и основы алгоритмизации».

Приобретенные знания в рамках данной дисциплины необходимы в дальнейшем при изучении дисциплин, ориентированных на проектирование систем управления, программирования аппаратуры регулирующих устройств автоматики (контроллеров) и при выполнении выпускной квалификационной работы.

3. Планируемые результаты обучения по дисциплине

3.1 Планируемые результаты обучения по дисциплине

Компетенции, закрепленные за дисциплиной в ОХОП:

ОПК-6. Способен разрабатывать и использовать алгоритмы и программы, современные информационные технологии, методы и средства контроля, диагностики и управления, пригодные для практического применения в сфере своей профессиональной деятельности

Индикаторы компетенций, закреплённых за дисциплиной в ОХОП:

ИОПК-6.2. Разрабатывает и использует алгоритмы и программы, пригодные для практического применения в сфере своей профессиональной деятельности.

Показатели оценивания индикаторов достижения компетенций

Знать:

31. Основные вычислительные алгоритмы обработки данных, используемых при управлении техническими системами.

32. Приемы отладки, тестирования и сопровождения разработанных программных средств.

Уметь:

У1. Разрабатывать приложения на языке C/C++, отлаживать и тестировать разработанные программные средства.

У2. Разрабатывать программы для устройств управления и встраиваемых систем.

У3. Осуществлять выбор эффективных проектных подходов к синтезу структур данных и алгоритмов их обработки в условиях конкретных практических приложений.

У4. Применять структуры данных и основные алгоритмы их обработки, а также библиотеки готовых подпрограмм, используемых для решения прикладных задач.

У5. Производить контроль параметров и элементов систем управления с применением программ диагностики и технической документации.

3.2. Технологии, обеспечивающие формирование компетенций

Проведение лекционных, практических и лабораторных занятий, самостоятельная работа под руководством преподавателя.

4. Трудоемкость дисциплины и виды учебной работы ОЧНАЯ ФОРМА ОБУЧЕНИЯ

Таблица 1а. Распределение трудоемкости дисциплины по видам учебной работы

Вид учебной работы	Зачетные единицы	Академические часы
Общая трудоемкость дисциплины	5	180
Аудиторные занятия (всего)		60
В том числе:		
Лекции		30
Практические занятия (ПЗ)		30
Самостоятельная работа обучающихся (всего)		120=84+36 (экз.)
В том числе:		
Курсовая работа		не предусмотрена
Курсовой проект		не предусмотрен
Расчетно-графические работы		не предусмотрены
Реферат		не предусмотрен
Другие виды самостоятельной работы:		
- изучение теоретической части дисциплины		44
- выполнение заданий по практическим занятиям		40
Текущий контроль успеваемости и промежуточная аттестация (экзамен)		36 (экз.)
Практическая подготовка при реализации дисциплины (всего)		0

ЗАОЧНАЯ ФОРМА ОБУЧЕНИЯ

Таблица 1б. Распределение трудоемкости дисциплины по видам учебной работы

Вид учебной работы	Зачетные единицы	Академические часы
Общая трудоемкость дисциплины	5	180
Аудиторные занятия (всего)		12
В том числе:		
Лекции		4
Практические занятия (ПЗ)		4
Лабораторные работы (ЛР)		4
Самостоятельная работа обучающихся (всего)		168=159+9(экз.)
В том числе:		
Курсовая работа		не предусмотрена
Курсовой проект		не предусмотрен
Расчетно-графические работы		не предусмотрены
Реферат		не предусмотрен
Другие виды самостоятельной работы: - изучение теоретической части дисциплины и выполнение контрольной работы; - подготовка к лабораторным работам - выполнение заданий по практическим занятиям		89 30 40
Текущий контроль успеваемости и промежуточная аттестация (экзамен)		9 (экз.)
Практическая подготовка при реализации дисциплины (всего)		0

5. Структура и содержание дисциплины

5.1. Структура дисциплины

ОЧНАЯ ФОРМА ОБУЧЕНИЯ

Таблица 2а. Модули дисциплины, трудоемкость в часах и виды учебной работы

№	Наименование модуля	Труд-ть, часы	Лекции	Практич. занятия	Лаб. работы	Сам. работа
1	Логическая организация программы на языке C/C++	14	2	2		5+5 (экз.)
2	Реализация вычислительных операций	33	2	2		24+5 (экз.)
3	Циклы в программах и управляющие структуры	16	2	4		5+5 (экз.)
4	Указатели в программах и их применение	33	4	4		20+5 (экз.)
5	Составные типы данных: структура, объединение, перечисление. Переименование составных типов данных	23	4	4		10+5 (экз.)

6	Функции в языке C/C++.	25	6	4		12+3 (экз.)
7	Объектно-ориентированное программирование	20	6	6		3+5 (экз.)
8	Организация разработки прикладного программного обеспечения	16	4	4		5+3 (экз.)
Всего на дисциплину		180	30	30		84+36 (экз.)

ЗАОЧНАЯ ФОРМА ОБУЧЕНИЯ

Таблица 2б. Модули дисциплины, трудоемкость в часах и виды учебной работы

№	Наименование модуля	Труд-ть часы	Лекции	Практич. занятия	Лаб. работы	Сам. работа
1	Логическая организация программы на языке C/C++	13	2	1		9+1 (экз.)
2	Реализация вычислительных операций	35		1		33+1 (экз.)
3	Циклы в программах и управляющие структуры.	16			2	13+1 (экз.)
4	Указатели в программах и их применение	30	2	2		25+1 (экз.)
5	Составные типы данных: структура, объединение, перечисление. Переименование составных типов данных.	18				16+2 (экз.)
6	Функции в языке C/C++	32			2	29+1 (экз.)
7	Объектно-ориентированное программирование	20				19+1 (экз.)
8	Организация разработки прикладного программного обеспечения	16				15+1 (экз.)
Всего на дисциплину		180	4	4	4	159+9 (экз.)

5.2. Содержание дисциплины

МОДУЛЬ 1 «Логическая организация программы на языке C/C++»

Задачи прикладного программирования. Инструментарий программиста. Обзор возможностей языков программирования C/C++/C#. Преемственность языков и сферы их применения, сложности языков, основные стандарты. Рекомендуемая литература и Интернет-ресурсы для самостоятельного изучения.

Алфавит языка. Идентификаторы. Ключевые слова. Структура программы на языке C. Функции. Этапы компиляции программы на языке C. Препроцессор. Директивы препроцессору `#include` и `#define`. Заголовочные файлы. Объектные модули программы. Сборка программы из объектных модулей в исполняемый файл.

Переменные: объявление, определение, инициализация. Комментарии в программах. Группы типов данных. Локальные и глобальные переменные программы. Стек программы. Статическая память программы. Область видимости и область действия переменных. Многомодульные программы. Ключевые слова *static* и *extern*.

МОДУЛЬ 2 «Реализация вычислительных операций»

Арифметические операторы. Оператор присваивания. Составное присваивание. Операции целочисленной арифметики. Совместимость и преобразование типов. Множественное присваивание. Операции инкремента и декремента. Выражения и составные операторы. Использование операторных скобок `{}`. Операции отношения. Константы целого и вещественного типа в языке C/C++. Приоритет и порядок выполнения арифметических операций. Обзор стандартной математической библиотеки языка C. Библиотечные функции ввода–вывода данных.

МОДУЛЬ 3 «Циклы в программах и управляющие структуры»

Операторы циклически выполняемых операций. Вложенные циклы. Досрочное завершение циклов. Операторы *break*, *continue*, *goto*. Метки в программах.

Операторы ветвления `if {} else {}`. Логические операторы `==`, `||`, `&&`, `!=`. Оператор множественного выбора *switch()*. Побитовые операции в языке Си (`|`, `&`, `~`, `<<`, `>>`).

МОДУЛЬ 4 «Указатели в программах и их применение»

Тип данных «указатель» Указатели на базовые типы данных. Основные операции, связанные с применением указателей. Указатель на неопределенный тип данных. Механизм доступа (запись/чтение) к переменным программы через указатель. Ссылки в языке Си++. Сравнение ссылки с указателем.

Массивы как пример гомогенной структуры данных. Одномерные и многомерные массивы. Объявление и инициализация массивов. Размещение массивов в памяти, индексация и доступ к элементам массива. Связь массивов и указателей. Динамическая память программы. Выделение и освобождение динамической памяти для массивов данных с помощью операторов *new* и *delete*.

Символьные переменные. Массивы символов (строки) в языке Си. Обзор библиотечных функций проверки класса литер и работы со строками.

Типичные ошибки в программах, возникающие из-за неправильного использования указателей и методы их устранения.

МОДУЛЬ 5 «Составные типы данных: структура, объединение, перечисление. Переименование составных типов данных»

Структурирование неоднородных типов данных. Тип данных *struct* как пример гетерогенной структуры данных. Синтаксис объявления структур данных.

Вложенные структуры. Указатели на структуры данных. Доступ к полям структуры (операция «точка», операция «стрелка»). Битовые поля структур. Тип данных *union*. Применения битовых полей структур с типом данных *union*. Тип данных перечисление (*enum*): синтаксис объявления, примеры и особенности применения. Назначение и применение операции *typedef*. Сравнение операции *typedef* с директивой *define*.

МОДУЛЬ 6 «Функции в языке C/C++»

Объявление функций. Вызов функций. Механизм связи подпрограммы с основной программой (формальные и фактические параметры). Использование указателей и ссылок для связи функций. Передача параметров (простые переменные, массивы, структуры, строки) функциям и типы возвращаемых значений. Назначение прототипов функций. Технологии модульного и структурного программирования.

МОДУЛЬ 7: «Объектно-ориентированное программирование»

Принципы объектно-ориентированного программирования: абстрагирование, инкапсуляция, наследование, полиморфизм. Преимущества использования объектно-ориентированного подхода при создании сложных программных продуктов.

Инкапсуляция. Классы в языке Си++. Синтаксис объявления класса. Отличие структуры от класса. Члены-данные и члены-методы класса. Разграничение прав доступа к членам классов. Оператор привязки метода к классу. Место объявления описания класса, объекта класса и его методов в программе. Конструкторы и деструкторы классов. Назначение и особенности. Назначение ключевого слова *this* в классе.

Наследование классов. Синтаксис объявления класса наследника. Назначение ключевого слова *protected*. Разграничение прав доступа к классу наследнику. Механизм наследования конструкторов и деструкторов классов.

Полиморфизм. Перегрузка функций. Перегрузка операторов (унарного, бинарного, особые случаи).

МОДУЛЬ 8: «Организация разработки прикладного программного обеспечения»

Цикл разработки прикладного программного обеспечения: концептуализация, анализ, проектирование, кодирование, тестирование, сопровождение. Затраты времени на разработку программ по стадиям. Критерии оценки качества программ. Документирование программ. Интегрированные среды разработки приложений. Взаимодействие пользователя с программами (интерфейс командной строки, текстовый интерфейс, оконный интерфейс). Создание собственных и применение файлов-библиотек сторонних производителей. Правила хорошего стиля программирования и представления результатов обработки научных и инженерных данных (визуализация таблиц, графиков зависимостей). Организация разработки программного обеспечения в команде программистов.

5.3. Лабораторные работы ЗАОЧНАЯ ФОРМА ОБУЧЕНИЯ

Таблица 3. Лабораторные работы и их трудоемкость

Порядковый номер модуля. Цели лабораторных работ.	Наименование лабораторных работ	Трудоем- кость в часах
Модуль 3 Цель: научить студентов практическим навыкам разработке вычислительных алгоритмов, программ циклической и ветвящейся структуры	Разработка программы с применением арифметических операторов, операторов for, while, do while	2
Модуль 6 Цель: научить студентов разрабатывать многомодульные приложения	Разработка программы, состоящей из нескольких функций, на примерах решения задач линейной алгебры	2

5.4. Практические занятия ОЧНАЯ ФОРМА ОБУЧЕНИЯ

Таблица 4а. Тематика, форма практических занятий (ПЗ) и их трудоемкость

Порядковый номер модуля. Цели практических занятий.	Примерная тематика занятий и форма их проведения	Трудое мкость в часах
Модуль 1,2 Цель: ознакомить студентов с методами разработки программного обеспечения с использованием стандартных интерфейсных компонентов ОС Windows в среде C++ Builder и научиться строить приложения, управляемые событиями.	Знакомство с средой программирования. Разработка вычислительной программы с графическим интерфейсом пользователя.	4
Модуль 3 Цель: научить студентов практическим навыкам разработке вычислительных алгоритмов, программ циклической и ветвящейся структуры.	Разработка программы с применением арифметических операторов, операторов for, while, do while.	4
Модуль 4 Цель: научить студентов практическим навыкам разработке алгоритмов и программ с применением указателей Изучить основные приемы работы с массивами данных.	Разработка программы работы с динамического выделения памяти для многомерных массивов данных. Разработка программы для работы с массивами на примере выполнения матричных операций.	4

Модуль 5 Цель: научить студентов разрабатывать приложения с применением структурированных типов данных.	Разработка программы с применением структурированных типов данных на примерах программирования арифметических операций с комплексными числами	4
Модуль 6 Цель: научить студентов разрабатывать многомодульные приложения.	Разработка программы, состоящей из нескольких функций, на примерах решения задач линейной алгебры	4
Модуль 7 Цель: научить студентов методам разработки классов и производных от базовых классов	Разработка программы с использованием классов производных от базовых классов на примере программирования задач работы с комплексными числами	6
Модуль 8 Цель: научить методике подключения к проекту программы файлов-библиотек программных модулей. Освоить методы создания собственных файлов-библиотек.	Разработка программы с применением библиотечных модулей на примере обработки статистических данных	4

ЗАОЧНАЯ ФОРМА ОБУЧЕНИЯ

Таблица 4б. Тематика, форма практических занятий (ПЗ) и их трудоемкость

Порядковый номер модуля. Цели практических занятий.	Примерная тематика занятий и форма их проведения	Трудоемкость в часах
Модуль 1 Цель: ознакомить студентов с методами разработки программного обеспечения с использованием стандартных интерфейсных компонентов ОС Windows в среде C++ Builder и научиться строить приложения, управляемые событиями	Знакомство с средой программирования. Разработка вычислительной программы с графическим интерфейсом пользователя	2
Модуль 2 Цель: научить студентов практическим навыкам разработке вычислительных алгоритмов, программ циклической и ветвящейся структуры	Разработка программы с применением арифметических операторов, операторов for, while, do while	1
Модуль 4 Цель: научить студентов практическим навыкам разработке алгоритмов и программ с применением указателей Изучить основные приемы работы с массивами данных.	Разработка программы работы с динамического выделения памяти для многомерных массивов данных Разработка программы для работы с массивами на примере выполнения матричных операций	1

6. Самостоятельная работа обучающихся и текущий контроль успеваемости

6.1. Цели самостоятельной работы

Формирование способностей к самостоятельному познанию и обучению, поиск литературы, обобщение, оформление и представление полученных результатов, их критический анализ, разработка документации.

6.2. Организация и содержание самостоятельной работы

Самостоятельная работа заключается в изучении отдельных тем по заданию преподавателя по рекомендуемой учебной литературе, в подготовке к практическим занятиям, к текущему контролю успеваемости и подготовке к экзамену.

Для очной формы обучения лабораторные работы не предусмотрены. Для заочной формы обучения выполняются 2 лабораторные работы, которые защищаются посредством тестирования или устным опросом (по желанию обучающегося). Максимальная оценка за каждую выполненную лабораторную работу – 5 баллов, минимальная – 3 балла. Выполнение всех лабораторных работ обязательно.

В ходе изучения дисциплины предусмотрены 8 практических занятий по очной форме обучения и 2 по заочной форме обучения, которые защищаются посредством тестирования или устным опросом (по желанию обучающегося). Максимальная оценка за каждую выполненную лабораторную работу – 5 баллов, минимальная – 3 балла. Выполнение всех практических занятий обязательно.

В случае невыполнения практического занятия по уважительной причине студент должен выполнить пропущенные занятия в часы, отведенные на консультирование с преподавателем.

В течение семестра студентами выполняются две письменные контрольные работы. В контрольной работе предлагается дать развернутые ответы на два вопроса из числа предлагаемых на экзамене. Знания оцениваются по пятибалльной шкале. Полученные оценки служат для более объективной промежуточной аттестации студентов, предъявляемой в деканат.

Подготовка к практическим занятиям заключается в чтении рекомендованной преподавателем литературы и разработке программ для закрепления знаний полученных на практических занятиях. Задания для самостоятельной разработки программ содержатся в тексте методических указаний для выполнения каждого практического занятия.

7. Учебно-методическое и информационное обеспечение дисциплины

7.1. Основная литература по дисциплине

1. Семакин, И.Г. Основы программирования: учебник для среднего проф. образования: в составе учебно-методического комплекса / И.Г. Семакин, А.П. Шестаков. - 6-е изд.; стер. - М.: Академия, 2007. - 204 с. - (УМК-У). - Библиогр.: с. 428- 429.-Текст: непосредственный. - ISBN 5-294-00054-7: 127 р. 60 к.-(ID=64416-10)

2. Кудрина, Е.В. Основы алгоритмизации и программирования на языке С#: учебное пособие для вузов / Е.В. Кудрина, М.В. Огнева. - Москва: Юрайт, 2022. - (Высшее образование). - Образовательная платформа Юрайт. - Текст: электронный. - ISBN 978-5-534-09796-2. - URL: <https://urait.ru/bcode/494874>. - (ID=145304-0)

3. Павлов, Л.А. Структуры и алгоритмы обработки данных: учебник для вузов / Л.А. Павлов, Н.В. Первова. - 3-е изд. - Санкт-Петербург [и др.]: Лань, 2021. - ЭБС Лань. - Текст: электронный. - ISBN 978-5-8114-7259-8. - URL: <https://e.lanbook.com/book/156929>. - (ID=144941-0)

4. Трофимов, В.В. Алгоритмизация и программирование: учебник для вузов / В.В. Трофимов, Т.А. Павловская; Трофимов В.В., Павловская Т.А.; под редакцией В.В. Трофимова. - Москва: Юрайт, 2022. - Образовательная платформа Юрайт. - URL: <https://urait.ru/bcode/491215>. - (ID=145176-0)

7.2. Дополнительная литература по дисциплине

1. Апанасевич, С.А. Структуры и алгоритмы обработки данных. Линейные структуры: учебное пособие / С.А. Апанасевич. - Санкт-Петербург [и др.]: Лань, 2019. - ЭБС Лань. - Текст: электронный. - ISBN 978-5-8114-3366-7. - URL: <https://e.lanbook.com/book/113934>. - (ID=144942-0)

2. Вирт, Н. Алгоритмы и структуры данных / Н. Вирт. - Москва: Мир, 1989. - 360 с. - 6786-00. - (ID=48989-3)

3. Маер, А.В. Введение в структуры и алгоритмы обработки данных: учебное пособие / А.В. Маер, О.С. Черепанов; Курганский государственный университет. - Курган: Курганский государственный университет, 2021. - ЭБС Лань. - Текст: электронный. - ISBN 978-5-4217-0576-5. - URL: <https://e.lanbook.com/book/177907>. - (ID=144949-0)

4. Павловская, Т.А. С/С++. Структурное и объектно-ориентированное программирование: практикум / Т.А. Павловская, Ю.А. Щупак. - Санкт-Петербург: Питер, 2011. - 347 с. - (Учебное пособие). - Текст: непосредственный. - ISBN 978-5-459-00613-1: 225 p. - (ID=93703-4)

5. Сыромятников, В.П. Структуры и алгоритмы обработки данных: практикум: учебное пособие / В.П. Сыромятников; МИРЭА - Российский технологический университет. - Москва: МИРЭА - Российский технологический университет, 2020. - ЭБС Лань. - Текст: электронный. - URL: <https://e.lanbook.com/book/163915>. - (ID=144952-0)

6. Хорев, П.Б. Объектно-ориентированное программирование: учебное пособие для вузов по направлению "Информатика и вычислительная техника" / П.Б. Хорев. - 3-е изд.; испр. - Москва: Академия, 2011. - 447 с. - (Высшее профессиональное образование. Информатика и вычислительная техника). - Текст: непосредственный. - ISBN 978-5-7695-8091-8: 430 p. 10 к. - (ID=89010-5)

7. Хусаинов, Б.С. Структуры и алгоритмы обработки данных. Примеры на языке Си: учеб. пособие по напр. 654600 - Информатика и вычисл. техника / Б.С. Хусаинов. - Москва: Финансы и статистика, 2004. - 464 с.: ил. + 1 электрон. опт. диск (CD-ROM). - (Учебное пособие). - Библиогр.: с. 462 - 464. - Текст: непосредственный. - ISBN 5-279-02775-8: 226 p. 88 к. - (ID=57401-5)

7.3. Методические материалы

1. Андреева, О.В. Информатика и программирование: основы алгоритмизации и программирования: лабораторный практикум / О.В. Андреева. - Москва: МИСиС, 2014. - ЦОР IPR SMART. - Текст: электронный. - ISBN 978-5-87623-779-8. - URL: <https://www.iprbookshop.ru/98856.html>. - (ID=142785-0)

2. Кирнос, В.Н. Информатика 2. Основы алгоритмизации и программирования на языке C++: учебно-методическое пособие / В.Н. Кирнос. - Томск: Томский государственный университет систем управления и радиоэлектроники: Эль Контент, 2013. - ЦОР IPR SMART. - Текст: электронный. - URL: <https://www.iprbookshop.ru/14011.html>. - (ID=142716-0)

3. Кораблин, Ю.П. Структуры и алгоритмы обработки данных: учебно-методическое пособие / Ю.П. Кораблин, В.П. Сыромятников, Л.А. Скворцова; МИРЭА - Российский технологический университет. - Москва: МИРЭА - Российский технологический университет, 2020. - ЭБС Лань. - Текст: электронный. - URL: <https://e.lanbook.com/book/163860>. - (ID=144959-0)

4. Основы алгоритмизации и программирования: лабораторный практикум. Направление подготовки 230400.62 – Информационные системы и технологии. Профили подготовки: «Информационные системы в бизнесе», «Безопасность информационных систем». Бакалавриат / составитель Е.И. Николаев; Северо-Кавказский федеральный университет. - Ставрополь: Северо-Кавказский федеральный университет, 2015. - ЭБС Лань. - Текст: электронный. - URL: <https://e.lanbook.com/book/155245>. - (ID=146162-0)

7.4. Программное обеспечение по дисциплине

Операционная система Microsoft Windows: лицензии № ICM-176609 и № ICM-176613 (Azure Dev Tools for Teaching).

Microsoft Office 2007 Russian Academic: OPEN No Level: лицензия № 41902814.

Для проведения практических занятий используется среда разработки программного обеспечения C++ Builder версии 5 или старше.

7.5. Специализированные базы данных, справочные системы, электронно-библиотечные системы, профессиональные порталы в Интернет

ЭБС и лицензионные ресурсы ТвГТУ размещены:

ЭБС и лицензионные ресурсы ТвГТУ размещены:

1. Ресурсы: <https://lib.tstu.tver.ru/header/obr-res>
2. ЭК ТвГТУ: <https://elib.tstu.tver.ru/MegaPro/Web>
3. ЭБС "Лань": <https://e.lanbook.com/>
4. ЭБС "Университетская библиотека онлайн": <https://www.biblioclub.ru/>
5. ЭБС «IPRBooks»: <https://www.iprbookshop.ru/>
6. Электронная образовательная платформа "Юрайт" (ЭБС «Юрайт»): <https://urait.ru/>
7. Научная электронная библиотека eLIBRARY: <https://elibrary.ru/>
8. Информационная система "ТЕХНОРМАТИВ". Конфигурация "МАКСИМУМ": сетевая версия (годовое обновление): [нормативно-технические, нормативно-правовые и руководящие документы (ГОСТы, РД, СНИПы и др.). Диск 1, 2, 3, 4. - М.: Технорматив, 2014. - (Документация для профессионалов). - CD. - Текст: электронный. - 119600 р. - (105501-1)
9. База данных учебно-методических комплексов: <https://lib.tstu.tver.ru/header/umk.html>

УМК размещен: <https://elib.tstu.tver.ru/MegaPro/GetDoc/Megapro/116781>

8. Материально-техническое обеспечение дисциплины

Практические занятия и лабораторные работы проводятся в компьютерных классах ХТ-201, где каждый студент может работать на отдельной ЭВМ.

Для проведения практических занятий используется среда разработки программного обеспечения C++ Builder версии 5 или старше.

9. Оценочные средства для проведения промежуточной аттестации

9.1. Оценочные средства для проведения промежуточной аттестации в форме экзамена

1. Экзаменационный билет соответствует форме, утвержденной Положением о рабочих программах дисциплин, соответствующих федеральным государственным образовательным стандартам высшего образования с учетом профессиональных стандартов. Типовой образец экзаменационного билета приведен в Приложении. Обучающемуся даётся право выбора заданий из числа, содержащихся в билете, принимая во внимание оценку, на которую он претендует.

Число экзаменационных билетов – 15. Число вопросов (заданий) в экзаменационном билете – 3 (1 вопрос для категории «знать» и 2 вопроса для категории «уметь»).

Продолжительность экзамена – 60 минут.

2. Шкала оценивания промежуточной аттестации в форме экзамена – «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

3. Критерии оценки за экзамен:

для категории «знать»:

выше базового – 2;

базовый – 1;

ниже базового – 0;

критерии оценки и ее значение для категории «уметь»:

отсутствие умения – 0 баллов;

наличие умения – 2 балла.

«отлично» - при сумме баллов 5 или 6;

«хорошо» - при сумме баллов 4;

«удовлетворительно» - при сумме баллов 3;

«неудовлетворительно» - при сумме баллов 0, 1 или 2.

4. Вид экзамена – письменный экзамен, включающий решение задач с использованием ЭВМ.

5. База заданий, предъявляемая обучающимся на экзамене.

1. История языка программирования С. Область применения. Элементы языка Си: алфавит, ключевые слова, идентификаторы. Комментарии в программах. Структура программы на языке С. Заголовочные h-файлы.
2. Группы типов данных. Базовые типы данных в языке С/С++. Объявление и инициализация переменных. Ключевое слово unsigned. Локальные и глобальные переменные программы. Область действия переменных.
3. Арифметические операторы в Си. Оператор присваивания. Множественное присваивание Совместимость и преобразование типов. Выражения и составные операторы. Использование операторных скобок {}.

4. Многомодульные программы. Ключевые слова `static` и `extern`.
5. Назначение директив `include` и `define`. Использование аргументов в директиве `define`. Константы в языке C++.
6. Указатели на типы данных в языке C/C++. Основные операции, связанные с применением указателей. Использование указателей для связи функций в Си-программах.
7. Массивы указателей. Пример создания и инициализации многомерных массивов с применением массива указателей.
8. Основные ошибки, связанные с неправильным применением указателей. Методы их устранения.
9. Циклы в языке C. Использование операторных скобок `{}`. Правила записи и примеры типовых циклических алгоритмов. Вложенные циклы. Досрочное завершение циклов. Операторы `break` и `continue`. Оператор `goto`. Метки в программах.
10. Объявление и инициализация одномерных массивов. Связь массивов и указателей. Индексы и доступ к элементам массива. Передача массивов в качестве формальных параметров функциям.
11. Объявление и инициализация многомерных массивов. Передача двумерных массивов в качестве формальных параметров функциям.
12. Указатели на функции в языке Си. Примеры применения в программах.
13. Ссылки в языке Си++. Сравнение ссылки с указателем. Передача аргументов функциям с использованием ссылок.
14. Двойная и тройная косвенная адресация. Привести механизм доступа (запись/чтение) к переменной программы. Пример применения для создания многомерных массивов.
15. Функции в Си, C++. Объявление функций. Вызов функций. Назначение прототипов функций. Механизм связи подпрограммы с основной программой (формальные и фактические параметры). Передача параметров в функции и типы возвращаемых значений.
16. Объявление символьных переменных. Символьные константы. Объявление массива символов (строки) в языке Си. Ноль-символ в строке.
17. Объявление массива символьных строк. Привести пример выдачи на компонент Метод всего массива строк с применением функции `printf`.
18. Структурирование неоднородных типов данных. Тип данных `struct` в языке Си. Синтаксис описания структур. Доступ к элементам структур. Вложенные структуры. Передача структур в качестве формальных параметров функциям.
19. Массивы структур. Объявление, инициализация и доступ к элементам массива структур. Привести пример применения.
20. Битовые поля структур. Пример применения битовых полей структур с типом данных `union`.
21. Побитовые операции в языке Си (`|`, `&`, `~`, `<<`, `>>`). Примеры применения.
22. Условный оператор `if () {} else {}`. Логические операторы `==`, `||`, `&&`, `!=`.
23. Оператор множественного выбора `switch()`. Синтаксис оператора. Его преимущества в сравнении с условным оператором `if () {} else {}`. Привести пример применения в программе.

24. Динамическая память: выделение и освобождение памяти, размещение данных в динамической памяти. Выделение и освобождение памяти при работе с одиночными переменными и с массивами. Операторы `new` и `delete`.
25. Назначение операции `typedef`. Сравнение операции `typedef` с директивой `define`. Примеры применения.
26. Библиотечные функции. Заголовочные файлы и их назначение. Использование и подключение библиотек готовых модулей к файлу проекта приложения. Основные функции математической библиотеки языка C. Функция `printf` библиотеки ввода-вывода и ее основные команды форматирования для вывода данных.
27. Объектно-ориентированное программирование. В чем преимущество ООП в сравнении с модульным принципом разработки программ. Классы в языке Си++. Синтаксис объявления класса. Чем класс отличается от структуры языка C?
28. Члены-данные и члены-методы класса. Разграничение прав доступа к членам классов. Оператор привязки метода к классу. Место объявления объекта класса, самого класса и его методов в программе.
29. Конструкторы и деструкторы классов. Назначение и особенности. Назначение ключевого слова `this` в классе и пример применения.
30. Наследование классов в C++. Синтаксис объявления класса наследника. Назначение ключевого слова `protected`. Разграничение прав доступа к классу наследнику.

При ответе на вопросы экзамена допускается пользование справочными данными. Пользование различными техническими устройствами не допускается. При желании студента покинуть пределы аудитории во время экзамена экзаменационный билет после его возвращения заменяется.

Преподаватель имеет право после проверки письменных ответов на экзаменационные вопросы задавать студенту в устной форме уточняющие вопросы в рамках содержания экзаменационного билета, выданного студенту.

Иные нормы, регламентирующие процедуру проведения экзамена, представлены в Положении о текущем контроле успеваемости и промежуточной аттестации студентов.

9.2. Оценочные средства для проведения промежуточной аттестации в форме зачета

Учебным планом зачет по дисциплине не предусмотрен.

9.3. Оценочные средства для проведения промежуточной аттестации в форме курсового проекта или курсовой работы

Учебным планом курсовая работа (проект) по дисциплине не предусмотрены.

10. Методические рекомендации по организации изучения дисциплины.

Студенты перед началом изучения дисциплины ознакомлены с системами кредитных единиц и балльно-рейтинговой оценки.

Студенты, изучающие дисциплину, обеспечиваются электронными изданиями или доступом к ним, учебно-методическим комплексом по дисциплине, включая методические указания к выполнению практических работ и всех видов самостоятельной работы.

11. Внесение изменений и дополнений в рабочую программу дисциплины

Содержание рабочих программ дисциплин ежегодно обновляется протоколами заседаний кафедры по утвержденной «Положением о структуре, содержании и оформлении рабочих программ дисциплин по образовательным программам, соответствующим ФГОС ВО с учетом профессиональных стандартов» форме.

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тверской государственный технический университет»

Направление подготовки бакалавров 27.03.04 Управление в технических системах

Направленность (профиль) – Автоматизация и управление технологическими процессами и производствами

Кафедра «Автоматизация технологических процессов»

Дисциплина «Структуры и алгоритмы обработки данных»

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1

1. Вопрос для проверки уровня «ЗНАТЬ» – 0 или 1 или 2 балла:

Группы типов данных. Базовые типы данных в языке C/C++. Объявление и инициализация переменных. Ключевое слово *unsigned*. Локальные и глобальные переменные программы. Область действия переменных.

2. Задание для проверки уровня «УМЕТЬ» - 0 или 2 балла:

Разработать простое приложение в среде C++ Builder, в котором объявить переменные программы известные вам типов, их инициализировать и вывести на экран монитора отводимую для них память в байтах.

3. Задание для проверки уровня «УМЕТЬ» – 0 или 2 балла:

Продемонстрировать владение отладчиком программ.

Критерии итоговой оценки за экзамен:

«отлично» - при сумме баллов 5 или 6;

«хорошо» - при сумме баллов 4;

«удовлетворительно» - при сумме баллов 3;

«неудовлетворительно» - при сумме баллов 0, 1 или 2.

Составитель: доцент кафедры АТП _____ В.Г. Васильев

Заведующий кафедрой АТП: _____ Б.И. Марголис