



**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА
(РОСАВИАЦИЯ)**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ГРАЖДАНСКОЙ АВИАЦИИ»**

УТВЕРЖДАЮ



Ректор

Ю.Ю. Михальчевский

«21» октября 2021 года

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Программирование

Направление подготовки
01.03.04 Прикладная математика

Направленность программы (профиль)
Математическое и программное обеспечение систем управления

Квалификация выпускника
бакалавр

Форма обучения
очная

Санкт-Петербург
2021

Код компетенции/ индикатора	Результат обучения: наименование компетенции, индикатора компетенции
	ограничений
ИД ¹ _{УК2}	Формулирует конкретные задачи согласно поставленной цели и определяет последовательность действий для решения этих задач, выбирает наиболее оптимальный способ решения.
ИД ² _{УК2}	Рассматривает, оценивает и выбирает оптимальные способы решения задач, учитывая правовые нормы, имеющиеся ресурсы и иные ограничения
УК-6	Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни
ИД ¹ _{УК6}	Определяет цели и задачи собственной деятельности, выбирает способы и последовательность их реализации,
ИД ¹ _{УК6}	Понимает необходимость профессионально-личностного роста посредством непрерывного образования как основу саморазвития, выстраивает и реализует траекторию
ОПК-3	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности
ИД ¹ _{ОПК3}	Строит математические модели при решении научно-исследовательских задач
ИД ¹ _{ОПК3}	Использует аналитические и научные пакеты прикладных программ для создания математических моделей.
ПК-2	Способен разрабатывать алгоритмы и реализовывать их на основе современных парадигм, технологий и языков программирования
ИД ¹ _{ПК2}	Применяет имеющиеся технологии и знания при разработке и реализации алгоритмов в ходе профессиональной деятельности
ИД ² _{ПК2}	Оценивает адекватность и логичность применения разработанного алгоритма в рамках конкретной задачи

Планируемые результаты изучения дисциплины:

В результате освоения данной дисциплины студент должен

Знать:

- основные методы алгоритмизации;
- один из языков программирования;

- основные методы отладки программ;
- основные методы тестирования программ;
-
- Уметь:
- использовать языки и системы программирования для решения профессиональных задач;
- Владеть:
- навыками работы с одной из инструментальных систем программирования;
- навыками программирования для решения профессиональных задач;
- навыками отладки программ в одной из инструментальных систем программирования.
- навыками тестирования программ в одной из инструментальных систем программирования.

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 24 зачетные единицы, 864 академических часа.

Наименование	Всего часов	Семестры					
		3	4	5	6	7	8
Общая трудоемкость дисциплины	864	144	144	144	144	108	180
Контактная работа:	355	70,5	54,5	56,5	48,5	58,5	66,5
Лекции	150	28	18	28	16	28	32
практические занятия	-	-	-	-	-	-	-
Семинары	-	-	-	-	-	-	-
лабораторные работы	194	42	36	28	28	28	32
курсовой проект (работа)	4	-	-	-	4	-	-
Самостоятельная работа студента	381	65	72	70	78	16	80
Промежуточная аттестация:	135	9	18	18	18	36	36
контактная работа	7	0,5	0,5	0,5	0,5	2,5	2,5
самостоятельная работа по подготовке к зачёту с оценкой (3, 4, 5 и 6 семестры), экзамену (7, 8 семестры)	121	8,5	17,5	17,5	17,5	33,5	33,5

5. Содержание дисциплины

5.1. Соотнесения тем (разделов) дисциплины и формируемых компетенций

Темы (разделы) дисциплины	Количество часов	Компетенции				Образовательные технологии	Оценочные средства
		УК-2	УК-6	ОПК-3	ПК-2		
Тема 1. Среда программирования на алгоритмическом языке	45	+	+	+		ВК, Л, СРС, ЛР	ИЗ
Тема 2. Синтаксис алгоритмического языка	45	+	+	+		Л, СРС, ЛР	ИЗ
Тема 3. Составные типы данных	45	+	+	+		Л, СРС, ЛР	ИЗ
Тема 4. Модульное программирование	42	+	+	+		Л, СРС, ЛР	ИЗ
Тема 5. Программирование основных алгоритмов	42	+	+	+		Л, СРС, ЛР	ИЗ
Тема 6. Обработка основных структур данных	42	+	+	+		Л, СРС, ЛР	ИЗ
Тема 7. Ввод и вывод в C++.	8	+	+	+		Л, ЛР, СРС	ПрЗ
Тема 8. Типы данных в С и C++.	8	+	+	+		Л, ЛР, СРС	ПрЗ
Тема 9. Строки, указатели и одномерные массивы в С.	6	+	+	+		Л, ЛР, СРС	У, ПрЗ
Тема 10. Синтаксис языков программирования С и C++.	16	+	+	+		Л, ЛР, СРС	У, ПрЗ
Тема 11. Адресное пространство процесса на С.	6	+	+	+		Л, ЛР, СРС	У, ПрЗ
Тема 12. Препроцессор С/С++.	8	+	+	+		Л, ЛР, СРС	У, ПрЗ
Тема 13. Структурное программирование на С/С++.	8	+	+	+		Л, ЛР, СРС	У, ПрЗ
Тема 14. Указатели и ссылки в C++.	6	+	+	+		Л, ЛР, СРС	У, ПрЗ
Тема 15. Файловый ввод-вывод в С и C++.	10	+	+	+		Л, ЛР, СРС	У, ПрЗ
Тема 16. Обработка исключений.	8	+	+	+		Л, ЛР, СРС	У, ПрЗ
Тема 17. Перечисления и структуры в языках: С и C++.	16	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 18. Пространства имен в C++.	6	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 19. Классы и объекты в C++.	12	+	+	+	+	Л, ЛР, СРС	У, ПрЗ

Темы (разделы) дисциплины	Количество часов	Компетенции				Образовательные технологии	Оценочные средства
		УК-2	УК-6	ОПК-3	ПК-2		
Тема 20. Перегрузка функций и методов.	8	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 21. Конструкторы классов.	24	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 22. Дружественные функции и дружественные классы.	18	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 23. Статические поля и статические методы.	26	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 24. Ключевое слово this.	20	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 25. Строки в C++.	18	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 26. Шаблонный класс std::vector.	20	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 27. Шаблонный класс std::map.	10	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 28. Алгоритмы STL в C++.	12	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 29. Перегрузка операций в C++.	12	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 30. Правило «трех» в C++.	12	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 31. Наследование классов в C++.	12	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 32. Виртуальные методы и виртуальные деструкторы.	14	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 33. Синтаксис JavaScript	14	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 34. Обработка событий в JavaScript.	14	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 35. Массивы в JavaScript.	14	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 36. Дата и время в JavaScript. Глобальный объект window.	16	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 37. Методы в JavaScript.	24	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 38. Замыкания в JavaScript.	18	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 39. Взаимодействие JavaScript, HTML и CSS.	14	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 40. Объекты в JavaScript.	14	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Тема 41. Формы в браузере.	16	+	+	+	+	Л, ЛР, СРС	У, ПрЗ
Всего по дисциплине	729						
Промежуточная аттестация	135						

Темы (разделы) дисциплины	Количество часов	Компетенции				Образовательные технологии	Оценочные средства
		УК-2	УК-6	ОПК-3	ПК-2		
Итого по дисциплине	864						

5.2. Темы (разделы) дисциплины и виды занятий

Наименование темы (раздела) дисциплины	Л	ПЗ	С	ЛР	СРС	КР	Всего часов
Семестр 3							
Тема 1. Среда программирования на алгоритмическом языке	8	-	-	14	23	-	45
Тема 2. Синтаксис алгоритмического языка	10	-	-	14	21	-	45
Тема 3. Составные типы данных	10	-	-	14	21	-	45
Всего за семестр 3	28	-	-	42	65	-	135
Промежуточная аттестация							9
Итого за семестр 3							144
Семестр 4							
Тема 4. Модульное программирование	6	-	-	12	24	-	42
Тема 5. Программирование основных алгоритмов	6	-	-	12	24	-	42
Тема 6. Обработка основных структур данных	6	-	-	12	24	-	42
Всего за семестр 4	18	-	-	36	72	-	126
Промежуточная аттестация							18
Итого за семестр 4							144
Семестр 5							
Тема 7. Ввод и вывод в C++.	2	-	-	2	4	-	8
Тема 8. Типы данных в C и C++.	2	-	-	2	4	-	8
Тема 9. Строки, указатели и одномерные массивы в C.	2	-	-	2	2	-	6
Тема 10. Синтаксис языков программирования C и C++.	2	-	-	2	12	-	16
Тема 11. Адресное пространство процесса на C.	2	-	-	2	2	-	6
Тема 12. Препроцессор C/C++.	2	-	-	2	4	-	8
Тема 13. Структурное программирование на C/C++.	2	-	-	2	4	-	8

Наименование темы (раздела) дисциплины	Л	ПЗ	С	ЛР	СРС	КР	Всего часов
Тема 14. Указатели и ссылки в C++.	2	-	-	2	2	-	6
Тема 15. Файловый ввод-вывод в C и C++.	2	-	-	2	6	-	10
Тема 16. Обработка исключений.	2	-	-	2	4	-	8
Тема 17. Перечисления и структуры в языках: C и C++.	2	-	-	2	12	-	16
Тема 18. Пространства имен в C++.	2	-	-	2	2	-	6
Тема 19. Классы и объекты в C++.	2	-	-	2	8	-	12
Тема 20. Перегрузка функций и методов.	2	-	-	2	4	-	8
Всего за семестр 5	28	-	-	28	70	-	126
Промежуточная аттестация							18
Итого за семестр 5							144
Семестр 6							
Тема 21. Конструкторы классов	4	-	-	4	14	2	24
Тема 22. Дружественные функции и дружественные классы.	2	-	-	4	12	-	18
Тема 23. Статические поля и статические методы.	4	-	-	8	14	-	26
Тема 24. Ключевое слово this.	2	-	-	4	14	-	20
Тема 25. Строки в C++.	2	-	-	4	12	-	18
Тема 26. Шаблонный класс std::vector.	2	-	-	4	12	2	20
Всего за семестр 6	16	-	-	28	78	4	126
Промежуточная аттестация							18
Итого за семестр 6							144
7 семестр							
Тема 27. Шаблонный класс std::map.	4	-	-	4	2	-	10
Тема 28. Алгоритмы STL в C++.	6	-	-	4	2	-	12
Тема 29. Перегрузка операций в C++.	4	-	-	4	4	-	12
Тема 30. Правило «трех» в C++.	4	-	-	4	4	-	12
Тема 31. Наследование классов в C++.	4	-	-	6	2	-	12
Тема 32. Виртуальные методы и виртуальные деструкторы.	6	-	-	6	2	-	14
Всего за семестр 7	28	-	-	28	16	-	72
Промежуточная аттестация							36
Итого за семестр 7							108
Семестр 8							
Тема 33. Синтаксис JavaScript	4	-	-	4	6	-	14
Тема 34. Обработка событий в JavaScript.	2	-	-	2	10	-	14
Тема 35. Массивы в JavaScript.	2	-	-	2	10	-	14
Тема 36. Дата и время в JavaScript. Глобальный объект window.	4	-	-	4	8	-	16
Тема 37. Методы в JavaScript.	4	-	-	4	16	-	24
Тема 38. Замыкания в JavaScript.	4	-	-	4	10	-	18

Наименование темы (раздела) дисциплины	Л	ПЗ	С	ЛР	СРС	КР	Всего часов
Тема 39. Взаимодействие JavaScript, HTML и CSS.	4	-	-	4	6	-	14
Тема 40. Объекты в JavaScript.	4	-	-	4	6	-	14
Тема 41. Формы в браузере.	4	-	-	4	8	-	16
Всего за семестр 8	32	-	-	32	80	-	144
Промежуточная аттестация							36
Итого за семестр 8							180
Итого по дисциплине							864

Сокращения: Л – лекция, ПЗ – практическое занятие, ЛР – лабораторная работа, СРС – самостоятельная работа студента, КР – курсовой проект.

5.3 Содержание разделов дисциплины

Тема 1. Среда программирования на алгоритмическом языке

Постановка задачи. Метод решения задачи. Алгоритм решения. Программирование алгоритма. Отладка. Тестирование. Получение результатов. Верификация результатов.

Этапы выполнения в ЭВМ программы на алгоритмическом языке. Возможности среды программирования. Интерфейс среды. Средства отладки. Состав проекта. Управление проектом.

Структура исходного текста. Алфавит языка. Идентификаторы. Ключевые слова. Знаки операций. Операторы. Ввод и вывод данных. Комментарии.

Тема 2. Синтаксис алгоритмического языка

Концепция типа данных. Встроенные типы. Константы. Переменные. Типы результатов операций. Преобразование типов.

Линейные алгоритмы. Описания переменных. Ввод - вывод значений исходных данных и результатов. Вычислительные выражения. Присваивание. Программирование линейных вычислений.

Логический тип данных. Логические выражения. Операторы перехода, условные и выбора. Программирование условных вычислений и логических задач.

Циклы. Операторы циклов с параметром, предусловием, постусловием и прерыванием. Вложенные циклы. Форматный вывод. Программирование циклических вычислений.

Тема 3. Составные типы данных.

Понятие массива. Одномерные массивы. Многомерные массивы. Описание массива. Инициализация массива. Операции с массивами. Индексированные переменные. Программирование обработки массивов данных.

Понятие указателя. Описание указателей. Инициализация. Операции с указателями. Динамические переменные. Операции с динамическими переменными. Процедуры управления памятью. Динамические массивы.

Символьные типы данных. Строковые типы данных. Описание строки. Инициализация строки. Операции со строками. Стандартные процедуры и функции обработки строк. Программирование обработки строковой информации.

Понятие структуры. Описание структуры. Инициализация структуры. Операции со структурами. Квалифицированные имена Вложенные структуры. Понятие объединения. Особенности использования объединений. Программирование обработки структур данных.

Тема 4. Модульное программирование

Формат функции. Возврат результата. Передача параметров. Параметры – ссылки. Параметры - массивы. Локальные переменные. Программирование функций пользователя.

Понятие препроцессора. Директивы препроцессора. Макроопределения. Условная компиляция.

Понятие модуля. Описание модуля. Межмодульные связи. Классы памяти. Области видимости и существования имен. Использование модулей в программе. Компоновка многомодульных программ. Разработка многомодульных программ.

Тема 5. Программирование основных алгоритмов

Алгоритм и его свойства. Способы описания алгоритмов. Система команд алгоритмов для ЦВМ. Схемы алгоритмов. Структурное программирование. Нисходящее проектирование. Использование псевдокода. Разработка алгоритмов вычислительных задач.

Метод анализа вариантов. Схема алгоритма по методу анализа вариантов. Генерация вариантов. Проверка на решение. Контроль процесса перебора. Программирование решений прикладных задач методом анализа вариантов.

Понятие рекурсии. Рекурсивные функции. Прямая и косвенная рекурсия. Реентерабельность рекурсивного кода. Применение рекурсии при программировании решений прикладных задач.

Постановка задачи поиска. Алгоритмы поиска в неупорядоченных, упорядоченных, частично упорядоченных последовательностях. Применение методов поиска при программировании решений прикладных задач.

Постановка задачи сортировки. Классификация методов сортировки. Алгоритмы простых методов сортировки. Оценка эффективности методов. Сложные методы сортировки. Применение методов сортировки при программировании решений прикладных задач.

Тема 6. Обработка основных структур данных

Классификация динамических структур данных. Линейные динамические структуры данных. Линейные списки. Описание списка. Инициализация списка. Операции со списками. Алгоритмы обработки списков. Применение линейных списков при программировании решений прикладных задач.

Понятие текстового файла. Описание текстовых файлов. Стандартные функции обработки текстовых файлов. Форматированный ввод-вывод. Алгоритмы обработки текстовых файлов. Применение текстовых файлов при программировании решений прикладных задач.

Понятие бинарного файла. Описание бинарных файлов. Стандартные функции обработки бинарных файлов. Блочный ввод-вывод. Алгоритмы обработки бинарных файлов. Применение бинарных файлов при программировании решений прикладных задач.

Понятие строкового потока. Описание строковых потоков. Стандартные функции обработки строкового потока. Ввод-вывод в строковых потоках. Алгоритмы обработки строковых потоков. Применение строковых потоков при программировании решений прикладных задач.

Тема 7. Ввод и вывод в C++.

Заголовочные файлы. `<cstdio>`. Ввод и вывод на консоль в стиле C. Функции: `printf`, `fprintf`, `sprintf`, `scanf`, `fscanf`, `sscanf`. `<iostream>`. Ввод и вывод на консоль в стиле C++. Объекты: `std::cin` и `std::cout`. `'\n'` и `std::endl`. `'\t'`.

Тема 8. Типы данных в C и C++.

Целочисленные типы данных в C: знаковые и беззнаковые; умолчания. Соотношения их размеров и ограничения. Зависимость диапазонов от разрядности. `<limits.h>`. Вещественные типы данных в C. Их размеры, диапазоны и точность. `<float.h>`. 3. Стандарт IEEE 754. Смещенный порядок, усеченная мантисса. Машинное эpsilon. Представления: числовое и нечисловое, нормализованное и денормализованное. Вычисление диапазонов и точностей. Символьный тип в C как целочисленный. Международная 7-битная таблица ASCII. Характерные диапазоны символов. Национальные кодовые страницы: CP 1251, CP 866, KOI8-R. Расположение кириллицы. Тип `bool` в C++. Приведение типов данных в C.

Тема 9. Строки, указатели и одномерные массивы в C.

Строковые литералы в C. Статические массивы. Одномерные статические массивы. Одномерные символьные массивы. Строковые литералы как константные символьные массивы с нуль-терминатором. Указатели. Массивы и указатели: сходства и различия. Типы: `size_t` и `ptrdiff_t`. Адресная арифметика указателей. Приведение типов указателей. `<cstdio>`. Функции: `gets`, `fgets`, `puts` и `fputs`. Отличия 1-ой от 2-ой и 3-ей от 4-ой.

Тема 10. Синтаксис языков программирования C и C++.

Комментарии в C-стиле и C++-стиле. Алгоритмы. Среда программирования. Структура программы. Арифметические выражения. Унарные и бинарные

операции. Приоритет операций в C/C++. Линейные алгоритмы. Блоки `{}`. Блочная область видимости в C++. Разветвляющиеся алгоритмы: `if`, `if... else`, `if... else if... else`. Тернарная операция. Циклические алгоритмы. 3 вида циклов. Теорема Бома – Джакопини.

Тема 11. Адресное пространство процесса на C.

Константы в C. Константные указатели и указатели на константы. Структура адресного пространства процесса на C. Сегменты: `data`: `RO`, `RW`, `BSS`, `heap`; `stack`; `code`. Динамическое выделение и возврат памяти в C и C++. Динамические одномерные массивы. Функции: `malloc`, `calloc`, `realloc` и `free`; операции: `new` и `delete`. `<cstring>`. Функции: `strlen`, `strcmp`, `strcpy`, `strcat`, `strchr` и `strstr`. Ошибки при работе с указателями: дикие и висячие указатели.

Тема 12. Препроцессор C/C++.

Макросы: `#define` и `#undef`. `#include`. Директивы препроцессора: `#ifdef`, `#ifndef` и `#if !defined`, `#else`, `#elif` и `#endif`. Директивы: `#line`, `#error` и `#pragma`.

Тема 13. Структурное программирование на C/C++.

Парадигма структурного программирования. Обработка массивов. Сортировка. Многомерные статические и динамические массивы. Матрицы. Обработка матриц. Функции. Возвращаемые значения. Ключевое слово `void`. Прототипы. Объявления и определения. Возможности отсутствия прототипа в C и C++. Рекурсия. Модули. Единицы трансляции. `<ctype>`. Классификация и прототипы функций для работы с символами.

Тема 14. Указатели и ссылки в C++.

Ссылки и указатели. 3 способа передачи параметра в функцию: по значению, по указателю и по ссылке. Отличия в синтаксисе и удобство. Передача массива в функцию.

Тема 15. Файловый ввод-вывод в C и C++.

Текстовые и бинарные файлы. `<stdio>`. Чтение и запись для текстового файла средствами C. Чтение и запись для бинарного файла средствами C. `<fstream>`. Чтение и запись для текстового файла средствами C++. Чтение и запись для бинарного файла средствами C++. Дозапись в текстовый файл средствами C и C++. Лог-файлы. `<stdlib>`. Функция `getenv`. Функции: `rand` и `srand` – для генерации случайных чисел. Функции: `fread` и `fwrite`. Методы: `read` и `write`. «Перемотка» файлов средствами C и C++. Установка позиции записи/чтения.

Тема 16. Обработка исключений.

Ключевые слова: `try`, `throw` и `catch`. Типы значений исключений. Обработка всех остальных исключений. Возможность обработки `run-time errors`.

Тема 17. Перечисления и структуры в языках: С и С++.

Перечисления enum. Структуры struct. Объявления в стиле С и в стиле С++. Поля структуры и способы их инициализации. Размер в памяти и выравнивание полей по границам. Указатели на структуру. Селекторы. Операции над структурами. Возврат структуры из функции. Линейные односвязные списки. Создание и уничтожение односвязного списка без применения функций и с применением. <ctime>. Функции: time, localtime, mktime и strftime; структура tm.

Тема 18. Пространства имен в С++.

Единицы трансляции. Защита от множественного включения заголовочных файлов в единицы трансляции. Правило одного определения (ODR). Использование ключевых слов: extern и static. Пространство имен namespace. Директива using. Декларации using. Безымянные пространства имен. Встраиваемые inline функции. namespace std.

Тема 19. Классы и объекты в С++.

3 парадигмы ООП. Классы. Объекты. Сравнение ключевых слов: struct и class. Два вида членов класса: поля и методы. Модификаторы доступа к членам классов: private и public. Константные методы. Указатель на объект.

Тема 20. Перегрузка функций и методов.

Реализация определений методов класса в файле реализации .cpp, отделенном от заголовочного файла .h определением класса с прототипами-объявлениями методов. Реализация методов в определении класса. Перегрузка функций. Перегрузка методов. Шаблоны функций. Шаблоны классов. Шаблонные методы шаблонных классов.

Тема 21. Конструкторы классов.

Конструкторы. Пустой конструктор. Два способа инициализации полей. Синтаксис списка инициализации. Использование тернарной операции. Конструктор по умолчанию. Перегрузка конструкторов. Передача в конструктор параметров по умолчанию.

Тема 22. Дружественные функции и дружественные классы.

Дружественные функции. Дружественные классы. Объектно-ориентированное программирование игры «Змейка».

Тема 23. Статические поля и статические методы.

Статические поля. Инициализация статических полей. Статические методы. Передача указателя на статический метод в другой метод. Программирование класса математических функций.

Тема 24. Ключевое слово this.

Трудности при передаче указателя на нестатический метод в другой метод. Причины и обход этих трудностей. Адрес созданного объекта. Ключевое слово `this`.

Тема 25. Строки в C++.

Заголовочный файл `<string>`. Методы класса `std::string`. Отличия параметров функции `getline` и метода `getline`. Преобразования между `char *` и `std::string`. Классы: `std::ostream` и `std::ostringstream`.

Тема 26. Шаблонный класс `std::vector`.

Библиотека STL. Шаблоны, итераторы, алгоритмы. Заголовочный файл `<vector>`. Методы класса `std::vector`. Использование `std::vector` для работы с одномерными и многомерными массивами. Создание матриц и векторов и работа с ними с помощью шаблона `std::vector`.

Тема 27. Шаблонный класс `std::map`.

Заголовочный файл `<map>`. Методы класса `std::map`. Ассоциативные массивы: уникальные и неуникальные ключи, значения. Использование `std::map` для создания словарей и частотных словарей.

Тема 28. Алгоритмы STL в C++.

Алгоритмы STL в C++. Заголовочный файл `<algorithm>`: функции `max`, `swap`, `swap_ranges`, `transform`, `random_shuffle`, `sort`. Заголовочный файл `<numeric>` – функция `accumulate`.

Тема 29. Перегрузка операций в C++.

Перегруженные операции как методы класса. Перегруженные операции как дружественные функции. Унарные и бинарные операции. Программирование класса комплексных чисел. Программирование класса трехмерных векторов.

Тема 30. Правило «трех» в C++.

Динамическое выделение и возврат памяти в C++. Операции: `new` и `delete`. Одномерные и двумерные динамические массивы в C++. Реализация класса динамической матрицы с помощью одномерного динамического массива. Утечки памяти и необходимость деструктора. Вызов деструктора. Динамическое создание объекта в куче и необходимость `delete`. Особенности деструктора. Необходимость конструктора копирования. Формы конструктора копирования: `X(X &)` и `X(const X &)`. Необходимость перегруженной операции `=`.

Тема 31. Наследование классов в C++.

Организация наследования. Ключевое слово `protected`. Модификаторы наследования: `public`, `private` и `protected`. Права доступа при наследовании. Конструкторы производных классов и вызов конструкторов базовых классов. Переопределение методов в производных классах. Создание класса-наследника

квадратных матриц порядка 2 как производного от класса произвольных матриц.

Тема 32. Виртуальные методы и виртуальные деструкторы.

Виртуальные методы. Чисто виртуальные методы. Абстрактные классы. Использование указателей на базовый класс при создании динамического объекта производного класса. Необходимость виртуального деструктора базового класса.

Тема 33. Синтаксис JavaScript

Способы загрузки кода JavaScript в браузер. Классификация типов данных в JavaScript. Место JavaScript среди других языков программирования. Слабая и динамическая типизация. Способы преобразования типов данных. Особенности операции конкатенации. Особенности синтаксиса JavaScript. С-подобный синтаксис. Особенности логических операций и операций отношения в JavaScript.

Тема 34. Обработка событий в JavaScript

Событийные атрибуты. Атрибут id как объект. Объект Math в JavaScript. Числа и строки в JavaScript.

Стандарт IEEE 754. Точность и диапазон числового типа. Методы объекта window: parseInt и parseFloat. Форматы Unicode. UCS-2 и UTF-16. Использование кодовых точек Unicode в JavaScript. Строковые методы.

Тема 35. Массивы в JavaScript

Методы объекта Array. Свойство length. Двумерные массивы в JavaScript.

Тема 36. Дата и время в JavaScript. Глобальный объект window

Методы объекта Date. Глобальный объект window и его методы в JavaScript. Области видимости и времена жизни переменных в JavaScript.

Ключевое слово var. Роль глобального объекта window. Всплытие объявлений (hoisting).

Тема 37. Методы в JavaScript

Методы для вызова стандартных диалоговых окон в JavaScript. Методы для таймеров и задержек в JavaScript. Именованные и анонимные функции в JavaScript.

Тема 38. Замыкания в JavaScript

Замыкания (closures) в JavaScript. Псевдомассив arguments.

Псевдомассив arguments. Проверка переданных фактических параметров. Параметры по умолчанию.

Тема 39. Взаимодействие JavaScript, HTML и CSS

Объект document, его методы. Изменение заголовка документа в коде JavaScript. Метод getElementById. Свойство innerHTML. Взаимодействие JavaScript и CSS. Селекторы CSS. Объект style. Свойства style: display и position.

Тема 40. Объекты в JavaScript

Создание объектов и способы установки их свойств и методов. Цикл for... in.

Создание объектов с помощью функции-конструктора. Ключевое слово this. Создание методов в конструкторе. Наследование объектов.

Свойство proto объекта this. Свойство prototype функции-конструктора.

Тема 41. Формы в браузере

Создание форм в браузере. Создание элементов управления на форме. Управление стилем элементов управления с помощью CSS. Работа с формами с помощью JavaScript. Обработка событий. Чтение и запись данных на форме.

5.4 Практические занятия

Практические занятия учебным планом не предусмотрены.

5.5 Лабораторный практикум

Лабораторный практикум учебным планом не предусмотрен.

Номер темы дисциплины	Наименование лабораторных работ	Трудоемкость (часы)
3 семестр		
1	Лабораторная работа 1-2. Исследование процесса разработки тестов для вычислительной задачи	4
	Лабораторная работа 3-4. Исследование среды программирования C++.	4
	Лабораторная работа 5-7. Исследование структуры консольной программы на языке C++.	6
2	Лабораторная работа 8. Исследование стандартных типов данных C++.	2
	Лабораторная работа 9. Программирование линейных вычислений.	2
	Лабораторная работа 10-11. Программирование условных вычислений и логических задач.	4
	Лабораторная работа 12-14. Программирование циклических вычислений.	6
3	Лабораторная работа 15. Программирование обработки массивов данных.	2

Номер темы дисциплины	Наименование лабораторных работ	Трудоемкость (часы)
	Лабораторная работа 16-17. Программирование обработки матриц данных.	4
	Лабораторная работа 18-19. Программирование обработки динамических массивов.	4
	Лабораторная работа 20 Программирование обработки строковой информации.	2
	Лабораторная работа 21. Программирование обработки структур данных.	2
Всего за семестр 3		42
4 семестр		
4	Лабораторная работа 1. Программирование функций пользователя с параметрами - значениями.	2
	Лабораторная работа 2. Программирование функций пользователя с параметрами - ссылками.	2
	Лабораторная работа 3. Программирование функций пользователя с параметрами - массивами.	2
	Лабораторная работа 4. Программирование директив препроцессора.	2
	Лабораторная работа 5-6. Разработка многомодульных программ.	4
5	Лабораторная работа 7. Разработка алгоритмов вычислительных задач.	2
	Лабораторная работа 8-9. Программирование решений прикладных задач методом анализа вариантов.	4
	Лабораторная работа 10. Исследование применения рекурсии при программировании решений прикладных задач	2
	Лабораторная работа 11. Исследование применения методов поиска при программировании решений прикладных задач.	2
	Лабораторная работа 12. Исследование применения методов сортировки при программировании решений прикладных задач.	2
6	Лабораторная работа 13. Исследование применения линейных списков при программировании решений прикладных задач	2

Номер темы дисциплины	Наименование лабораторных работ	Трудоемкость (часы)
	Лабораторная работа 14-15. Исследование применения текстовых файлов при программировании решений прикладных задач	4
	Лабораторная работа 16. Исследование применения бинарных файлов при программировании решений прикладных задач	2
	Лабораторная работа 17-18. Исследование применения строковых потоков при программировании решений прикладных задач.	4
Всего за семестр 4		36
5 семестр		
7	Лабораторная работа 1. Ввод и вывод в C++.	2
8	Лабораторная работа 2. Типы данных в C и C++.	2
9	Лабораторная работа 3. Строки, указатели и одномерные массивы в C.	2
10	Лабораторная работа 4. Операции в C++. Разветвляющиеся и циклические алгоритмы.	2
11	Лабораторная работа 5. Адресное пространство процесса на C.	2
12	Лабораторная работа 6. Препроцессор C/C++.	2
13	Лабораторная работа 7. Обработка массивов. Функции. Рекурсия.	2
14	Лабораторная работа 8. Три способа передачи параметра в функцию.	2
15	Лабораторная работа 9. Чтение и запись для текстового и бинарного файла средствами C и C++.	2
16	Лабораторная работа 10. Обработка исключений.	2
17	Лабораторная работа 11. Структуры в C. Линейные односвязные списки.	2
18	Лабораторная работа 12. Пространства имен в C++.	2
19	Лабораторная работа 13. Классы и объекты в C++.	2
20	Лабораторная работа 14. Перегрузка функций и шаблоны.	2
Всего за семестр 5		28
6 семестр		
21	Лабораторная работа 1-2. Конструкторы классов.	4

Номер темы дисциплины	Наименование лабораторных работ	Трудоемкость (часы)
22	Лабораторная работа 3-4. Дружественные функции и дружественные классы.	4
23	Лабораторная работа 5-8. Статические поля и статические методы.	8
24	Лабораторная работа 9-10. Ключевое слово this.	4
25	Лабораторная работа 11-12. Строки в C++.	4
26	Лабораторная работа 13-14. Работа с одномерными массивами с помощью шаблонного класса std::vector.	4
Всего за семестр 6		28
7 семестр		
27	Лабораторная работа 1. Создание словаря с помощью шаблонного класса std::map.	2
	Лабораторная работа 2. Создание частотного словаря с помощью шаблонного класса std::map	2
28	Лабораторная работа 3-4. Алгоритмы STL в C++.	4
29	Лабораторная работа 5-6. Перегрузка операций класса трехмерных векторов	4
30	Лабораторная работа 7. Правило «трех» в C++.	2
	Лабораторная работа 8. Реализация класса динамической матрицы с помощью одномерного динамического массива.	2
31	Лабораторная работа 9. Наследование классов в C++.	2
	Лабораторная работа 10-11. Создание класса-наследника квадратных матриц порядка 2 как производного от класса произвольных матриц.	4
32	Лабораторная работа 12-14. Виртуальные методы и виртуальные деструкторы.	6
Всего за семестр 7		28
8 семестр		
33	Лабораторная работа 1-2. Способы загрузки кода JavaScript в браузер.	4
34	Лабораторная работа 3. Обработка событий в JavaScript.	2
35	Лабораторная работа 4. Массивы в JavaScript	2
36	Лабораторная работа 5. Глобальный объект window и его методы в JavaScript.	2
	Лабораторная работа 6. Области видимости и времена жизни переменных в JavaScript.	2

Номер темы дисциплины	Наименование лабораторных работ	Трудоемкость (часы)
37	Лабораторная работа 7. Методы для вызова стандартных диалоговых окон в JavaScript.	2
	Лабораторная работа 8. Методы для таймеров и задержек в JavaScript. Именованные и анонимные функции в JavaScript.	2
38	Лабораторная работа 9. Замыкания в JavaScript.	2
	Лабораторная работа 10. Псевдомассив arguments.	2
39	Лабораторная работа 11-12. Взаимодействие JavaScript и HTML, CSS.	4
40	Лабораторная работа 13-14. Объекты в JavaScript. Наследование объектов.	4
41	Лабораторная работа 15. Создание форм в браузере с помощью JavaScript.	2
	Лабораторная работа 16. Обработка информации, полученной с помощью формы. Передача данных в элементы управления.	2
Всего за семестр 8		32
Итого по дисциплине		194

5.6 Самостоятельная работа

Номер темы дисциплины	Виды самостоятельной работы	Трудоемкость (часы)
3 семестр		
1	1. Изучение теоретического материала [1, 2, 5] 2. Выполнение индивидуального задания [11-14]	23
2	1. Изучение теоретического материала [1, 2, 5] 2. Выполнение индивидуального задания [8, 11-14]	21
3	1. Изучение теоретического материала [1, 2, 5] 2. Выполнение индивидуального задания [8, 11-14]	21
Всего за семестр 3		65
4 семестр		
4	1. Изучение теоретического материала [1, 2, 5] 2. Выполнение индивидуального задания [8, 11-14]	24
5	1. Изучение теоретического материала [1, 2, 5] 2. Выполнение индивидуального задания [8, 11-14]	24

Номер темы дисциплины	Виды самостоятельной работы	Трудоемкость (часы)
6	1. Изучение теоретического материала [1, 2, 5] 2. Выполнение индивидуального задания [8, 11-14]	24
Всего за семестр 4		72
5 семестр		
7	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	4
8	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	4
9	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	2
10	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	12
11	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	2
12	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	4
13	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	4
14	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	2
15	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	6
16	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	4
17	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	12
18	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	2
19	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	8
20	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	4
Всего за семестр 5		70
6 семестр		
21	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14] 3. Выполнение курсовой работы (проекта) [11-14]	14

Номер темы дисциплины	Виды самостоятельной работы	Трудоемкость (часы)
22	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14] 3. Выполнение курсовой работы (проекта) [11-14]	12
23	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14] 3. Выполнение курсовой работы (проекта) [11-14]	14
24	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14] 3. Выполнение курсовой работы (проекта) [11-14]	14
25	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14] 3. Выполнение курсовой работы (проекта) [11-14]	12
26	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14] 3. Выполнение курсовой работы (проекта) [11-14]	12
Всего за семестр 6		78
7 семестр		
27	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	2
28	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	2
29	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	4
30	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	4
31	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	2
32	1. Изучение теоретического материала [1-3, 5] 2. Выполнение практического задания [8, 11-14]	2
Всего за семестр 7		16
8 семестр		
33	1. Изучение теоретического материала [1-5] 2. Выполнение практического задания [6-10]	6
34	1. Изучение теоретического материала [1-5] 2. Выполнение практического задания [6-10]	10
35	1. Изучение теоретического материала [4, 6-7] 2. Выполнение практического задания [9-14]	10

Номер темы дисциплины	Виды самостоятельной работы	Трудоемкость (часы)
36	1. Изучение теоретического материала [4, 6-7] 2. Выполнение практического задания [9-14]	8
37	1. Изучение теоретического материала [4, 6-7] 2. Выполнение практического задания [9-14]	16
38	1. Изучение теоретического материала [4, 6-7] 2. Выполнение практического задания [9-14]	10
39	1. Изучение теоретического материала [4, 6-7] 2. Выполнение практического задания [9-14]	6
40	1. Изучение теоретического материала [4, 6-7] 2. Выполнение практического задания [9-14]	6
41	1. Изучение теоретического материала [4, 6-7] 2. Выполнение практического задания [9-14]	8
Всего за семестр 4		80
Итого по дисциплине		381

5.7 Курсовые проекты

Наименование этапа выполнения курсового проекта	Трудоемкость (часы)
Этап 1. Выдача задания на курсовой проект	2
Этап 2. Выполнение раздела «Введение»	4
Этап 3. Выполнение раздела «Основная часть»	10
Этап 4. Выполнение разделов «Заключение», «Выводы».	4
Этап 5. Оформление курсового проекта	2
Защита курсового проекта	2
Итого по курсовому проекту:	22
самостоятельная работа студента, отведенная на выполнение курсового проекта	20
согласно учебному плану	4

6 Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. Солдатенко, И.С. **Практическое введение в язык программирования Си** [Электронный ресурс]: учебное пособие / И.С. Солдатенко, И.В. Попов. — Электрон. дан. — Санкт-Петербург : Лань, 2018. — 132 с. — Режим доступа: <https://e.lanbook.com/book/109619> — Загл. с экрана.

2. Огнева, М. В. **Программирование на языке с++: практический курс : учебное пособие для бакалавриата и специалитета** / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2019. — 335 с. — (Бакалавр и специалист). — ISBN 978-5-534-05123-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/438987>

3. Тузовский, А. Ф. **Объектно-ориентированное программирование : учебное пособие для прикладного бакалавриата** / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2019. — 206 с. — (Университеты России). — ISBN 978-5-534-00849-4. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://www.biblio-online.ru/bcode/434045>

4. Зудилова Т.В., Буркова М.Л. **Web-программирование JavaScript** [Электронный ресурс]. – СПб.: НИУ ИТМО, 2012. – 68 с. Режим доступа: <http://window.edu.ru/resource/612/76612/files/itmo879.pdf>, свободный (дата обращения: 21.01.2021).

б) дополнительная литература:

5. Павловская, Т.А. **С/С++.Программирование на языке высокого уровня. Для магистров и бакалавров: Учеб. для вузов. Допущ. Минобр. РФ** [Текст] / Т. А. Павловская. - СПб.: Питер, 2013. - 461с. Количество экземпляров: 10.

6. Сысолетин, Е. Г. **Разработка интернет-приложений : учебное пособие для вузов** / Е. Г. Сысолетин, С. Д. Ростунцев ; под научной редакцией Л. Г. Доросинского. — Москва : Издательство Юрайт, 2019 ; Екатеринбург : Изд-во Урал. ун-та. — 90 с. — (Университеты России). — ISBN 978-5-9916-9975-4 (Издательство Юрайт). — ISBN 978-5-7996-1911-4 (Изд-во Урал. ун-та). — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/438148>

7. Захаркина В.В. JavaScript. **Основы клиентского программирования: Учебное пособие** [Электронный ресурс]. – СПб.: СПбГУ, 2007. – 73 с. – Режим доступа: http://window.edu.ru/resource/394/57394/files/VZ_JavaScript_web.pdf свободный (дата обращения: 21.01.2021).

в) перечень ресурсов информационно-телекоммуникационной сети «Интернет»:

8. **Работа в Microsoft Visual Studio: курс интернет-университета информационных технологий** [Электронный ресурс] – Режим доступа: <https://www.intuit.ru/studies/courses/499/355/info> (дата обращения: 21.01.2021).

9. **Современный учебник JavaScript** [Электронный ресурс] — Режим доступа: <https://learn.javascript.ru/>, свободный (дата обращения: 11.03.2019).

10. **Учебный и образовательный сайт о JavaScript** [Электронный ресурс] — Режим доступа: <https://www.javascript.com/learn/> , свободный (дата обращения: 21.01.2021).

г) программное обеспечение (лицензионное), базы данных, информационно-справочные и поисковые системы:

11. **Единое окно доступа к образовательным ресурсам** [Электронный ресурс]. – Режим доступа: <http://window.edu.ru>, свободный (дата обращения: 21.01.2021).

12. **Электронная библиотека научных публикаций «eLIBRARY.RU»** [Электронный ресурс] — Режим доступа: <http://elibrary.ru/>, свободный (дата обращения: 21.01.2021).

13. **Электронно-библиотечная система издательства «Лань»** [Электронный ресурс] — Режим доступа: <https://e.lanbook.com>, свободный (дата обращения: 21.01.2021).

14. **Visual Studio Community** [Программное обеспечение] — Режим доступа: <https://visualstudio.microsoft.com/ru/vs/community/> , свободный (дата обращения: 21.01.2021).

7 Материально-техническое обеспечение дисциплины

Для обеспечения образовательного процесса используются следующие материально-технические ресурсы.

Компьютерные классы кафедры № 8 (ауд.: 800, 801, 803, 804) с доступом в Интернет, переносной проектор.

Информационно-справочные и материальные ресурсы библиотеки СПбГУГА.

Лицензионное программное обеспечение: Microsoft Office, Visual Studio Community, Notepad++, веб-браузеры Internet Explorer и Google Chrome.

8 Образовательные и информационные технологии

В рамках изучения дисциплины предполагается использовать следующие образовательные технологии.

Входной контроль проводится преподавателем в начале изучения дисциплины с целью коррекции процесса усвоения обучающимися дидактических единиц при изучении базовых дисциплин.

Лекция составляет основу теоретического обучения в рамках дисциплины и направлена на систематизированное изложение накопленных и актуальных научных знаний. Лекция предназначена для раскрытия состояния и перспектив

развития знаний в современных условиях. На лекции концентрируется внимание обучающихся на наиболее сложных и узловых вопросах, стимулируется их активная познавательная деятельность.

Ведущим методом в лекции выступает устное изложение учебного материала, который сопровождается одновременной демонстрацией слайдов, при необходимости привлекаются открытые Интернет-ресурсы, а также демонстрационные и наглядно-иллюстрационные материалы и практические примеры.

Лабораторная работа позволяет организовать учебную работу с реальными информационными объектами. Лабораторная работа как образовательная технология реализует следующие функции: овладение системой средств и методов практического исследования обучающимися, развитие творческих исследовательских умений обучающихся и расширение возможностей использования теоретических знаний для решения практических задач.

Курсовой проект по дисциплине представляет собой самостоятельную учебно-исследовательскую работу студента и ставит цель систематизировать, закрепить и углубить теоретические и практические знания, умения и навыки по профилю подготовки с целью их применения для решения профессиональных задач.

Лабораторные работы и курсовой проект по дисциплине являются составляющими практической подготовки обучающихся, так как предусматривают их участие в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

Самостоятельная работа обучающихся является составной частью учебной работы. Ее основной целью является формирование навыка самостоятельного приобретения знаний по некоторым вопросам теоретического курса, закрепление и углубление полученных знаний, самостоятельная работа со справочниками, периодическими изданиями и научно-популярной литературой. Самостоятельная работа включает выполнение учебных заданий, в том числе и индивидуальных, а также работу над курсовым проектом.

В рамках изучения дисциплины «Программирование» предполагается использовать в качестве информационных технологий среду Microsoft Office и Microsoft Visual Studio Community, Notepad++, веб-браузеры Internet.

9 Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации по итогам освоения дисциплины

Фонд оценочных средств дисциплины «Программирование» представляет собой комплекс методических и контрольных измерительных материалов, предназначенных для определения качества результатов обучения и уровня сформированности компетенций обучающихся в ходе освоения данной дисциплины. В свою очередь, задачами использования фонда оценочных средств являются осуществление как текущего контроля успеваемости студентов, так и

промежуточной аттестации в форме зачета с оценкой (экзамена).

Фонд оценочных средств дисциплины «Программирование» для текущего включает: индивидуальные задания, устные опросы, практические задания и темы курсовых работ (проектов).

Устный опрос проводится в течение 15 минут с целью контроля усвоения теоретического материала, излагаемого на лекции. Перечень вопросов определяется уровнем подготовки учебной группы, а также индивидуальными особенностями обучающихся.

Практическое задание предназначено для проверки умений и навыков применять полученные знания для решения профессиональных задач с использованием программных средств.

Курсовая работа (проект) – авторский научно-исследовательский проект студента, направленный на выработку исследовательских навыков, опыта работы с научными источниками и создание законченного самостоятельного исследования. Оценочным средством являются темы курсовых проектов, которые приведены в п. 9.3. Написание и защита курсовой работы (проекта) запланирована на 6 семестр.

Промежуточная аттестация по итогам освоения дисциплины проводится в виде зачета с оценкой (3, 4, 5, 6 семестры) и экзамена в 7 и 8 семестрах. Зачет с оценкой и экзамен предполагают устные ответы на 2 теоретических вопроса из перечня вопросов, вынесенных на промежуточную аттестацию, а также решение задачи.

9.1 Балльно-рейтинговая оценка текущего контроля успеваемости и знаний студентов по дисциплине

Не применяется.

9.2 Методические рекомендации по проведению процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Устный опрос оценивается следующим образом:

«зачтено»: обучающийся дает ответ на поставленный вопрос по существу и правильно отвечает на уточняющие вопросы;

«не зачтено»: обучающийся отказывается отвечать на поставленный вопрос, либо отвечает на него неверно и при формулировании дополнительных (вспомогательных) вопросов.

Решение практических задач на лабораторных работах оценивается:

«зачтено»: обучающийся самостоятельно правильно решает задачу, дает

обоснованную оценку по итогу решения;

«не зачтено»: обучающийся отказывается от выполнения задачи или не способен ее решить самостоятельно, а также с помощью преподавателя.

Доклад:

«зачтено»: грамотное и непротиворечивое изложение сути вопроса при использовании современных источников. Обучающийся способен сделать обоснованные выводы, а также уверенно отвечать на заданные в ходе обсуждения вопросы;

«не зачтено»: неудовлетворительное качество изложения материала и неспособность обучающегося сделать обоснованные выводы или рекомендации.

Письменная аудиторная работа:

«зачтено»: работа зачитывается в том случае, если задание выполнено полностью, в соответствии с поставленными требованиями и сделаны необходимые выводы;

«не зачтено»: работа не зачитывается в том случае, если обучающийся не выполнил задания, или результат выполнения задания не соответствует поставленным требованиям, а в заданиях и (или) ответах имеются существенные ошибки.

9.3 Темы курсовых проектов по дисциплине

1. «Метод Гаусса решения СЛАУ для произвольной матрицы». Рассмотреть: (1) выбор главного элемента по столбцу (схему частичного выбора), (2) выбор главного элемента по всей матрице (схему полного выбора).

2. «Итерационный метод Якоби для решения СЛАУ».

3. «Метод Гаусса решения СЛАУ для матрицы с диагональным преобладанием».

4. «Итерационный метод Зейделя для решения СЛАУ».

5. «Метод прогонки решения СЛАУ для трехдиагональной матрицы».

6. «Метод Холецкого решения СЛАУ для симметричной положительно определенной матрицы».

7. «Итерационный метод релаксации для решения СЛАУ».

8. «Метод Гаусса решения СЛАУ для симметричной положительно определенной матрицы».

9. «Применение метода Гаусса для вычисления определителя и обратной матрицы».

10. «Метод вращений для QR-разложения матрицы».

11. «Применение метода Гаусса для LU-разложения матрицы».

12. «Полиномиальная интерполяция с помощью многочлена Лагранжа и многочлена Ньютона».

Примечания:

1. Все матрицы квадратные порядков 5 и 6 (т.е. проверку производить для двух матриц-массивов входных данных).

2. Ввод матриц (и вектора свободных членов) производить из текстового файла (текстовых файлов) средствами С или С++, на усмотрение студента, в текстовом режиме.

3. Вывод результатов производить в другой текстовый файл (текстовые файлы).

4. Для хранения матриц и векторов использовать, на усмотрение студента, или динамические массивы, или шаблонный класс `std::vector`.

5. Для решения использовать ООП или ФООП, на Ваше усмотрение.

6. Студент может использовать несколько подходов к решению задач курсового проектирования.

9.4 Контрольные вопросы для проведения входного контроля остаточных знаний по обеспечивающим дисциплинам

1. Основные понятия компьютерной графики.
2. Когда и кем был введен координатный метод?
3. Что такое кривые Безье? Назовите основные достоинства.
4. Двоичная система счисления положительных целых чисел.
Заданное десятичное целое положительное число преобразовать в двоичное. Сложить результат с положительным двоичным числом, значение которого задано в десятичной форме. Результат преобразовать в десятичное число. Проверить правильность результата.
5. Шестнадцатеричная система счисления положительных целых чисел.
Заданное десятичное целое положительное число преобразовать в шестнадцатеричное. Сложить результат с положительным шестнадцатеричным числом, значение которого задано в десятичной форме. Результат преобразовать в десятичное число. Проверить правильность результата.
6. Кодирование двоичных целых отрицательных чисел.
Заданное десятичное целое положительное число преобразовать в двоичное. Вычесть из результата с положительное двоичное число, значение которого задано в десятичной форме. Результат преобразовать в десятичное число. Проверить правильность результата и его знака.
7. Основные математические логические операции И, ИЛИ, НЕТ.
Записать математическое логическое выражение для заданной логической фразы, например: "И не то, чтобы да, и не то, чтобы нет". Вычислить это математическое логическое выражение.
8. Логические схемы вычисления логических выражений.
По заданному математическому логическому выражению построить логическую схему реализации этого выражения комбинацией логических элементов И, ИЛИ, НЕТ.
9. Принцип хранимых данных.
Сформулировать принцип. Проиллюстрировать его примерами из информатики.

10. Принцип хранимой программы.
Сформулировать принцип. Проиллюстрировать его примерами из информатики.
11. Принцип двоичной системы счисления.
Сформулировать принцип. Проиллюстрировать его примерами из информатики.
12. Понятие алгоритма.
Сформулировать основные свойства алгоритма: дискретность, определенность, конечность, массовость.
13. Условные обозначения схем алгоритмов.
Простые блоки. Составные блоки. Потоки управления. Потоки данных. Подпрограммы.

9.5 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Компетенции	Показатели оценивания (индикаторы достижения) компетенций	Критерии оценивания
I этап		
УК-2	ИД ¹ _{УК2}	Знает: -основы алгоритмизации программ для ЭВМ. Умеет: -разрабатывать алгоритмы программ для ЭВМ.
УК-6	ИД ¹ _{УК6}	Знает: -основы программирования на алгоритмическом языке. Умеет: -программировать алгоритмы на алгоритмическом языке
ОПК-3	ИД ¹ _{ОПК3}	Знает: -основы отладки программ на алгоритмическом языке. Умеет: -отлаживать программы на алгоритмическом языке.

Компетенции	Показатели оценивания (индикаторы достижения) компетенций	Критерии оценивания
ПК-2	ИД ¹ _{ПК2}	<p>Знает:</p> <ul style="list-style-type: none"> -основы разработки тестов и тестирования программ на алгоритмическом языке. <p>Умеет:</p> <ul style="list-style-type: none"> Разрабатывать тесты и тестировать программ на алгоритмическом языке.

II этап		
УК-2	ИД ² _{УК2}	<p>Знает:</p> <ul style="list-style-type: none"> -методы алгоритмизации программ для изучаемой среды программирования. <p>Умеет:</p> <ul style="list-style-type: none"> -разрабатывать алгоритмы программ для изучаемой среды программирования.
УК-6	ИД ² _{УК6}	<p>Знает:</p> <ul style="list-style-type: none"> -методы программирования в изучаемой среде программирования. <p>Умеет:</p> <ul style="list-style-type: none"> -программировать алгоритмы в изучаемой среде программирования.
ОПК-3	ИД ² _{ОПК3}	<p>Знает:</p> <ul style="list-style-type: none"> -методы отладки программ в изучаемой среде программирования. <p>Умеет:</p> <ul style="list-style-type: none"> -отлаживать программы в изучаемой среде программирования.
ПК-2	ИД ² _{ПК2}	<p>Знает:</p> <ul style="list-style-type: none"> -методы разработки тестов и тестирования программ в изучаемой среде программирования. <p>Умеет:</p> <ul style="list-style-type: none"> -разрабатывать тесты и тестировать программы в изучаемой среде программирования.

Шкала оценивания при проведении промежуточной аттестации «Отлично» выставляется обучающемуся, показавшему всесторонние, систематизированные, глубокие знания по рассматриваемой компетенции и

умение уверенно применять их на практике при решении задач, свободное и правильное обоснование принятых решений. Отвечая на вопрос, может быстро и безошибочно проиллюстрировать ответ собственными примерами. Обучающийся самостоятельно правильно решает задачу, дает обоснованную оценку итогам решения.

«Хорошо» выставляется обучающемуся, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задачи некоторые неточности, хорошо владеет всем содержанием, видит взаимосвязи, но не всегда делает это самостоятельно без помощи преподавателя. Обучающийся решает задачу верно, но при помощи преподавателя.

«Удовлетворительно» выставляется обучающемуся, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы в рамках заданной компетенции, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации. Отвечает только на конкретный вопрос, соединяет знания из разных разделов курса только при наводящих вопросах преподавателя. Ситуационная задача решена не полностью, или содержатся незначительные ошибки в расчетах.

«Неудовлетворительно» выставляется обучающемуся, который не знает большей части основного содержания учебной программы дисциплины в рамках компетенций, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач. Не раскрыты глубина и полнота при ответах. Задача не решена даже при помощи преподавателя.

Шкала оценивания курсового проекта

Шкала оценивания	Составляющие	Признаки
Отлично	Практическая часть	Обучающийся показывает умения и навыки выполнения расчетов необходимых показателей для формирования разделов финансовой отчетности. Расчеты в курсовом проекте обоснованы и выполнены правильно на 90-100 %.
	Выводы	Выводы грамотно сформулированы и обоснованы.
	Использованные источники	Использованные источники подобраны грамотно, имеются

Шкала оценивания	Составляющие	Признаки
		нормативные источники. Их количество соответствует требованиям к курсовому проекту.
	Оформление	Курсовой проект оформлен аккуратно согласно требованиям к оформлению без орфографических и грамматических ошибок.
	Своевременность выполнения	Курсовой проект выполнен и сдан на проверку своевременно.
	Защита	Доступно и ясно представляет результаты курсового проекта. Ответы на вопросы полные, глубокие. Обучающийся всесторонне оценивает и интерпретирует полученные результаты, доказывает их значимость. Грамотно и аргументировано представляет комментарии к расчетам.
Хорошо	Практическая часть	Обучающийся показывает умения и навыки выполнения расчетов необходимых показателей для формирования разделов финансовой отчетности. Расчеты в курсовом проекте обоснованы и выполнены правильно на 80-90 %.
	Выводы	Выводы сформулированы с небольшими неточностями.
	Использованные источники	Использованные источники подобраны грамотно. Их количество соответствует требованиям к курсовому проекту.
	Оформление	Курсовой проект оформлен аккуратно согласно требованиям к оформлению с небольшим количеством орфографических и грамматических ошибок.
	Своевременность выполнения	Курсовой проект выполнен и сдан на проверку своевременно.
	Защита	Доступно и ясно представляет результаты курсового проекта. Ответы на вопросы полные.

Шкала оценивания	Составляющие	Признаки
		Обучающийся оценивает и интерпретирует полученные результаты с незначительными неточностями, Демонстрирует самостоятельное мышление.
Удовлетворительно	Практическая часть	Обучающийся показывает слабые навыки выполнения расчетов необходимых показателей, формирования разделов финансовой отчетности. Расчеты обоснованы и выполнены правильно на 70-80 %.
	Выводы	Выводы сформулированы со значительными неточностями или не все выводы сформулированы.
	Использованные источники	Использованные источники подобраны небрежно. Их количество меньше, чем соответствует требованиям к курсовому проекту.
	Оформление	Курсовой проект оформлен неаккуратно с большим количеством орфографических и грамматических ошибок.
	Своевременность выполнения курсового проекта	Курсовой проект выполнен и сдан на проверку позже указанного срока.
	Защита	Обучающийся с трудом докладывает результаты курсового проекта. Ответы на вопросы неполные. Обучающийся не может оценить полученные результаты и интерпретирует их со значительными неточностями.
Неудовлетворительно	Практическая часть	Обучающийся не демонстрирует умения и навыки расчетов необходимых показателей, расчеты выполнены с большим количеством ошибок или не в полном объеме.
	Выводы	Выводы не сформулированы.
	Использованные источники	Использованные источники не соответствуют теме.
	Оформление	Оформление курсового проекта не соответствует требованиям. Большое

Шкала оценивания	Составляющие	Признаки
		количество орфографических и грамматических ошибок.
	Защита	Обучающийся не может представить результаты курсового проекта. Не отвечает на вопросы или отвечает неверно.

9.6 Типовые контрольные задания для проведения текущего контроля и промежуточной аттестации по итогам обучения по дисциплине

9.6.1. Типовые контрольные задания для проведения текущего контроля и промежуточной аттестации (3 семестр)

Типовые задания для проведения промежуточной аттестации в виде индивидуальных заданий (3 семестр):

1. $e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots$ Вычислить значение с точностью 0,001.

2. $D = \sum_{i=1}^8 p_i x_i^2 - M_x^2$, где $M_x = \sum_{i=1}^8 p_i \cdot x_i$ Вычислить значение

Перечень типовых вопросов к зачету с оценкой (3 семестр) для проведения промежуточной аттестации по дисциплине

1. Стандартные типы данных.
2. Модификаторы типов short, unsigned, long.
3. Неявное преобразование типов.
4. Явное преобразование типов.
5. Стандартные функции.
6. Использование стандартных библиотек.
7. Простые операторы: выражение, последовательность, блок.
8. Управляющие операторы. if, switch, break, continue, goto, return.
9. Операторы циклов. for, while, do-while, break, continue

Типовая задача к зачету с оценкой (3 семестр) для проведения промежуточной аттестации по дисциплине

Дана строка в виде ЧИСЛО+ЧИСЛО+ЧИСЛО , где ЧИСЛО – несколько цифр, составляющих целое число без знака. Подсчитать результат вычисления этого выражения.

9.6.2. Типовые контрольные задания для проведения текущего контроля и промежуточной аттестации (3 семестр)

Типовые задания для проведения промежуточной аттестации в виде индивидуальных заданий (4 семестр):

1. Разработать структуру данных, содержащую фамилию студента, одну экзаменационную оценку и дату сдачи экзамена. Использовать ее для программирования экзаменационной ведомости группы. Запрограммировать описание экзаменационной ведомости в виде:

- массива структур,

- текстового файла структур,
- бинарного файла структур,
- линейного списка структур.

2. Запрограммировать в разработанной ведомости (см. предыдущее задание) задачи:

- вывод ведомости в табличной форме,
- добавление нового студента,
- удаление существующего студента,
- изменение данных существующего студента,
- сортировка ведомости по фамилиям студентов в алфавитном порядке,
- поиск студентов по заданным значениям полей структуры:
 - по фамилии,
 - по оценке,
 - по дате.

Перечень типовых вопросов к зачету с оценкой (4 семестр) для проведения промежуточной аттестации по дисциплине

1. Типы данных языка программирования.
2. Линейные программы.
3. Разветвляющиеся программы.
4. Циклические программы.
5. Массивы.
6. Указатели.
7. Символы и строки.
8. Структуры и объединения.
9. Функции.
10. Препроцессор.
11. Модули.
12. Алгоритмы.
13. Алгоритмы анализа вариантов.
14. Рекурсивные алгоритмы.
15. Алгоритмы поиска.
16. Алгоритмы сортировки.
17. Динамические структуры данных.
18. Файловые текстовые потоки.
19. Файловые бинарные потоки.
20. Строковые потоки.

Типовая задача к экзамену (4 семестр) для проведения промежуточной аттестации по дисциплине

Вычислить число Фибоначчи порядка n :

$$F_n = F_{n-1} + F_{n-2} \quad , \quad F_1 = F_2 = 1 \quad , \quad n = 1, 2, \dots$$

9.6.3. Типовые контрольные задания для проведения текущего контроля и промежуточной аттестации (5 семестр)

Типовые практические задания (5 семестр)

1. Описать класс, включающий заданные поля и методы, двумя способами: без конструктора и с конструктором. Написать тестирующие программы, создающие массив объектов.

2. Объект - предложение. Параметры: массив слов ($n < 10$) и их количество. Методы: инициализирующий поля с использованием символьной строки, метод, определяющий количество слов, длина которых больше 3 символов. Определить процент слов, длиной меньше трех символов в заданном тексте, состоящим из нескольких предложений.

Типовые вопросы для устного опроса (5 семестр)

1. Ключевое слово `protected`.
2. Модификаторы наследования: `public`, `private` и `protected`.
3. Права доступа при наследовании.
4. Конструкторы производных классов и вызов конструкторов базовых классов.
5. Переопределение методов в производных классах.
6. Виртуальные методы.

Типовые вопросы для проведения промежуточной аттестации в форме зачета с оценкой (5 семестр)

1. Константные методы.
2. Перегрузка функций.
3. Перегрузка методов.
4. Шаблоны функций.
5. Шаблоны классов.
6. Шаблонные методы шаблонных классов.
7. Конструкторы.
8. Пустой конструктор.
9. Два способа инициализации полей.
10. Синтаксис списка инициализации.

Типовая практическая задача для проведения промежуточной аттестации в форме зачета с оценкой (5 семестр)

Описать класс, включающий заданные поля и методы, двумя способами: без конструктора и с конструктором. Написать тестирующие программы, создающие массив объектов. Объект - предложение. Параметры: массив слов ($n < 10$) и их количество. Методы: инициализирующий, выводящий данные об

объекте и метод, определяющий количество слов, начинающихся с согласных букв.

9.6.4 Типовые контрольные задания для проведения текущего контроля и промежуточной аттестации (6 семестр)

Типовые практические задания (6 семестр)

1. Разработать и реализовать иерархию классов для описанных объектов предметной области, используя механизмы наследования. Проверить ее на тестовом примере, с демонстрацией всех возможностей разработанных классов на конкретных данных. Даны: объект символ, умеющий печатать свое значение и отвечать на запрос о значении своего символьного поля; объект целое число, умеющий печатать свое значение и отвечать на запрос о значении своего числового поля; объект «запись», состоящий из двух полей – символа и двух целых чисел. Объект умеет выводить на печать значение своих элементов, возвращать значение своих полей по запросу.

Типовые вопросы для устного опроса (6 семестр)

1. Чисто виртуальные методы.
2. Абстрактные классы.
3. Использование указателей на базовый класс при создании динамического объекта производного класса.
4. Что стоит помнить при использовании исключений в конструкторе объекта?

Типовые вопросы для проведения промежуточной аттестации в форме экзамена (6 семестр)

1. 3 способа передачи параметра в функцию
2. Защита от множественного включения заголовочных файлов в единицы трансляции.
3. Правило одного определения
4. Директива using.
5. Декларации using.
6. Безымянные пространства имен.
7. Встраиваемые inline функции.
8. namespace std.
9. Перечисления enum.
10. 3 парадигмы ООП.
11. Два вида членов класса.
12. Модификаторы доступа к членам классов.
13. Константные методы.
14. Перегрузка функций.

15. Перегрузка методов.
16. Шаблоны функций.
17. Шаблоны классов.
18. Шаблонные методы шаблонных классов.
19. Конструкторы.
20. Пустой конструктор.
21. Два способа инициализации полей.
22. Синтаксис списка инициализации.
23. Конструктор по умолчанию.
24. Перегрузка конструкторов.
25. Передача в конструктор параметров по умолчанию.
26. Дружественные функции.
27. Дружественные классы.
28. Статические поля.
29. Инициализация статических полей.
30. Статические методы.
31. Ключевое слово `this`.
32. Методы класса `std::string`.
33. Отличия параметров функции `getline` и метода `getline`.
34. Преобразования между `char *` и `std::string`.
35. Классы: `std::ostream` и `std::ostringstream`.
36. Библиотека STL.
37. Шаблоны, итераторы, алгоритмы.
38. Методы класса `std::vector`.

Типовая практическая задача для проведения промежуточной аттестации в форме экзамена (6 семестр)

Описать класс, включающий заданные поля и методы, двумя способами: без конструктора и с конструктором. Написать тестирующие программы, создающие массив объектов. Объект - квартира. Параметры: площадь и стоимость. Методы: инициализирующий, выводящий на экран данные о полях объекта и метод, который определяет стоимость одного метра.

9.6.5 Типовые контрольные задания для проведения текущего контроля и промежуточной аттестации (7 семестр)

Перечень типовых вопросов для устного опроса (7 семестр)

1. Зачем нужен виртуальный деструктор?
2. Необходимость виртуального деструктора базового класса.
3. Множественное наследование. Вызов конструкторов и деструкторов базовых классов.

Типовое практическое задание (7 семестр)

Написать стековый калькулятор, который принимает в качестве аргумента командной строки имя файла, содержащего команды. Если аргумента нет, то использовать стандартный поток ввода для чтения команд. Использовать вещественные числа.

Реализовать следующий набор команд:

- **#** - строка с комментарием.
- **POP, PUSH** — снять/положить число со/на стек(a).
- **+, -, *, /, SQRT** – арифметические операции. Используют один или два верхних элемента стека, изымают их из стека, помещая результат назад
- **PRINT** — печать верхнего элемента стека (без удаления).
- **DEFINE** — задать значение параметра. В дальнейшем везде использовать вместо параметра это значение.

Пример (должно вывести 2):

```
DEFINE a 4
PUSH a
SQRT
PRINT
```

Перечень типовых вопросов к экзамену для проведения промежуточной аттестации по дисциплине (7 семестр)

1. Методы класса `std::map`.
2. Ассоциативные массивы: уникальные и неуникальные ключи, значения.
3. Алгоритмы в C++.
4. Заголовочный файл `<algorithm>`: функции `max`, `swap`, `swap_ranges`, `transform`, `random_shuffle`, `sort`.
5. Заголовочный файл `<numeric>` – функция `accumulate`.
6. Перегрузка операций в C++.
7. Перегруженные операции как методы класса.
8. Перегруженные операции как дружественные функции.
9. Унарные и бинарные операции.
10. Правило «трех» в C++.
11. Динамическое выделение и возврат памяти в C++
12. Утечки памяти и необходимость деструктора.
13. Вызов деструктора.
14. Динамическое создание объекта в куче и необходимость `delete`.
15. Особенности деструктора.
16. Необходимость конструктора копирования.
17. Формы конструктора копирования: `X(X &)` и `X(const X &)`.
18. Необходимость перегруженной операции `=`.
19. Наследование классов в C++.
20. Ключевое слово `protected`.
21. Модификаторы наследования: `public`, `private` и `protected`.

22. Права доступа при наследовании.
23. Конструкторы производных классов и вызов конструкторов базовых классов.
24. Переопределение методов в производных классах.
25. Виртуальные методы.
26. Чисто виртуальные методы.
27. Абстрактные классы.
28. Использование указателей на базовый класс при создании динамического объекта производного класса.
29. Необходимость виртуального деструктора базового класса.

Типовая практическая задача к экзамену для проведения промежуточной аттестации по дисциплине (7 семестр)

Необходимо реализовать контейнер для хранения больших целых чисел, с поддержкой техники ввода/вывода данных аля поток ввода/вывода. Для представления больших целых чисел необходимо использовать класс, разработанный в предыдущем задании.

Разрабатываемый класс должен предоставлять следующую функциональность:

Как контейнер:

- Возможность узнать размер контейнера (метод **size**).
- Возможность вставить значение в любое место контейнера (метод **insert**). Место определяется по индексу. Индекс **0** - вставить в начало контейнера; индекс **N** или **-1** (где **N** - размер контейнера на момент выполнения операции) - вставить в конец контейнера. В случае недопустимого значения индекса необходимо выбросить исключение **out_of_range**.

- Возможность вставить в любое место контейнера набор значений, представленных других экземпляром (объектом) вашего контейнера (метод **insert**). Место вставки первого элемента определяется по индексу, аналогично предыдущему случаю. Порядок вставляемых элементов сохраняется. В случае попытки вставить экземпляр контейнера "в себя" необходимо выбросить исключение **logic_error**

- Возможность удалить элемент контейнера по произвольному индексу (метод **remove**). В случае недопустимого значения индекса необходимо выбросить исключение **out_of_range**.

Как поток ввода/вывода:

- Возможность вставить значение в *текущий конец* контейнера (оператор **<<**).
- Возможность вставить набор значений в *текущий конец* контейнера (оператор **<<**).
- Возможность извлечь значение из *текущего конца* контейнера (оператор **>>**).

- Возможность извлечь все значения из *текущего конца* контейнера (оператор >>).
- Возможность определить состояние потока ввода - определить есть ли еще элементы - (оператор !).
- Возможность установить *голову* контейнера *текущем концом* контейнера (манипулятор **begin**).
- Возможность установить *хвост* контейнера *текущем концом* контейнера (манипулятор **end**).
- Возможность изменить *текущий конец* контейнера на противоположный (манипулятор **swap**).

9.6.6 Типовые контрольные задания для проведения текущего контроля и промежуточной аттестации (8 семестр)

Перечень типовых вопросов для устного опроса (8 семестр)

1. Области видимости и времена жизни переменных в JavaScript. Ключевое слово var. Роль глобального объекта window. Всплытие объявлений (hoisting).
2. Создание объектов с помощью функции-конструктора. Ключевое слово this. Создание методов в конструкторе.
3. Методы для вызова стандартных диалоговых окон в JavaScript.
4. Глобальный объект window и его методы в JavaScript.
5. Методы для таймеров и задержек в JavaScript.
6. Функции в JavaScript. Псевдомассив arguments. Проверка переданных фактических параметров. Параметры по умолчанию.
7. Создание форм в браузере. Создание элементов управления на форме. Управление стилем элементов управления с помощью CSS. Работа с формами с помощью JavaScript. Обработка событий. Чтение и запись данных на форме.

Типовое практическое задание (8 семестр)

1. Описать класс, включающий заданные поля и методы, двумя способами: без конструктора и с конструктором. Написать тестирующие программы, создающие массив объектов. Объект - предложение. Параметры: массив слов ($n < 10$) и их количество. Методы: инициализирующий, метод вывода информации о полях объекта и метод, определяющий и возвращающий по запросу количество слов, длиннее 5 букв. Определить процент слов длиннее 5 букв в заданном тексте.

Перечень типовых вопросов к экзамену для проведения промежуточной аттестации по дисциплине (8 семестр)

1. Общие сведения о клиентских и серверных Web-технологиях.
 2. Синтаксис языка JavaScript и основные типы данных. Примеры.
 3. Обработка событий в языке JavaScript.
 4. Свойства и методы объектов.
 5. Работа со стандартными структурами данных на языке JavaScript.
- Примеры.
6. Области видимости и замыкания. Примеры.
 7. Работа со стандартными элементами интерфейса браузера.
 8. Объектно-ориентированное программирование в сети Internet.

Типовые практические задания к экзамену для проведения промежуточной аттестации по дисциплине (8 семестр)

1. Найти несколькими способами, применяя стандарт IEEE 754, свойства и методы объекта Math и циклы while или do while, машинную точность и диапазон типа 'number'.

2. С помощью генератора случайных чисел создать два массива: из 30 целочисленных значений среднесуточных атмосферных давлений, равномерно распределенных в диапазоне [750; 770] мм рт. ст. и из 30 целочисленных значений среднесуточных температур, равномерно распределенных в диапазоне [10; 25] °С. Вывести эти массивы в окно браузера в 3 столбца из 30 строк в формате: 1 июня (4 пробела) p = 762 мм рт. ст. (4 пробела) t = 13 °С. С помощью формулы $\rho = \frac{p[\text{Pa}] \cdot \mu}{R \cdot T}$, учитывая, что 1 мм рт. ст. = 133,3 Па, $\mu = 0,029$ кг/моль, $R = 8,31$ Дж/(К моль), $T = t + 273,15$ К, найти число месяца с максимальной плотностью воздуха и саму эту плотность. Затем отсортировать массив давлений по возрастанию, а массив температур по убыванию.

3. С помощью таймеров, строковых методов и замыканий (closures) реализовать «закольцованный заголовок» окна браузера: заголовок «Следующая остановка – ст. м. «Московская»» должен представлять собой бегущую влево строку. Скорость – 2 символа в секунду.

Создать функцию, которая с помощью объекта Date позволит получить название дня недели словом: «Mon», «Tue», ... Ввести глобальную переменную типа Boolean, истинную, если неделя – пятидневка. Значение этой переменной ввести с помощью стандартного диалогового окна, запросив у пользователя. Создать функцию, которая с помощью объекта Date позволит получить название вида дня недели: «Working day» (рабочий) или «Day off» (выходной). Узнать сегодняшний день недели. Узнать вид сегодняшнего дня недели. Узнать, какой день недели и его вид: будут через 2 дня, будут через 3 дня, были 3 дня назад. Узнать, сколько в текущем месяце дней. Узнать, сколько в текущем месяце понедельников, вторников, ..., воскресений. Результаты записать в числовой

массив из семи элементов. Отсортировать этот массив по возрастанию. Результат записать в два столбца в виде: день недели (*4 пробела*) количество.

10 Методические рекомендации для обучающихся по освоению дисциплины

Методика преподавания дисциплины «Программирование» характеризуется совокупностью методов, приемов и средств обучения, обеспечивающих реализацию содержания и учебно-воспитательных целей дисциплины, которая может быть представлена как некоторая методическая система, включающая методы, приемы и средства обучения. Такой подход позволяет более качественно подойти к вопросу освоения дисциплины обучающимися.

Учебные занятия начинаются и заканчиваются по времени в соответствии с утвержденным режимом СПб ГУГА в аудиториях согласно семестровым расписаниям теоретических занятий. На занятиях, предусмотренных расписанием, обязаны присутствовать все обучающиеся.

Лекции являются одним из важнейших видов учебных занятий и составляют основу теоретической подготовки обучающихся по дисциплинам. Лекция имеет целью дать систематизированные основы научных знаний по дисциплине, раскрыть состояние и перспективы прогресса конкретной области науки и экономики, сконцентрировать внимание на наиболее сложных и узловых вопросах. Эта цель определяет дидактическое назначение лекции, которое заключается в том, чтобы ознакомить обучающихся с основным содержанием, категориями, принципами и закономерностями изучаемой темы и предмета обучения в целом, его главными идеями и направлениями развития. Именно на лекции формируется научное мировоззрение обучающегося, закладываются теоретические основы фундаментальных знаний будущего управленца, стимулируется его активная познавательная деятельность, решается целый ряд вопросов воспитательного характера.

Каждая лекция должна представлять собой устное изложение лектором основных теоретических положений изучаемой дисциплины или отдельной темы как логически законченное целое и иметь конкретную целевую установку. Особое место в лекционном курсе по дисциплине занимают вводная и заключительная лекции.

Вводная лекция должна давать общую характеристику изучаемой дисциплины, подчеркивать новизну проблем, указывать ее роль и место в системе изучения других дисциплин, раскрывать учебные и воспитательные цели и кратко знакомить обучающихся с содержанием и структурой курса, а также с организацией учебной работы по нему. Заключительная лекция должна давать научно-практическое обобщение изученной дисциплины, показывать перспективы развития изучаемой области знаний, навыков и практических умений.

Лабораторные работы проводятся в целях выработки практических умений


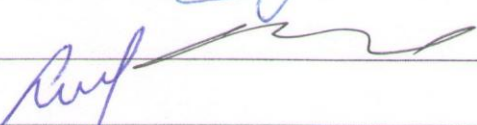
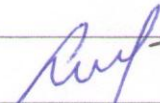
и приобретения навыков при решении управленческих задач. Основным содержанием этих работ является практическая работа каждого обучающегося. Назначение лабораторных работ – закрепление, углубление и комплексное применение на практике теоретических знаний, выработка умений и навыков обучающихся в решении практических задач. Вместе с тем, на этих занятиях, осуществляется активное формирование и развитие навыков и качеств, необходимых для последующей профессиональной деятельности. Лабораторные работы проводятся по наиболее сложным вопросам дисциплины и имеют целью углубленно изучить ее содержание, привить обучающимся навыки самостоятельного поиска и анализа информации, умение делать обоснованные выводы, аргументировано излагать и отстаивать свое мнение. Каждая лабораторная работа заканчивается, как правило, кратким подведением итогов, указаниями преподавателя о последующей самостоятельной работе.

Промежуточная аттестация по итогам освоения дисциплины проводится в виде зачетов с оценкой (3, 4, 5, 6 семестр), курсового проекта (6 семестр), экзамена (7, 8 семестр). К моменту сдачи экзамена должны быть успешно пройдены предыдущие формы контроля. Экзамен, зачеты с оценкой и курсовой проект позволяют оценить уровень освоения компетенций за весь период изучения дисциплины.

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 01.03.04 «Прикладная математика». Программа рассмотрена и утверждена на заседании кафедры №8 «Прикладной математики и информатики»

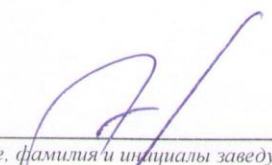
« 15 » сентября 2021 года, протокол № 2

Разработчики

К. Т. Н., доцент		Павлов В. Д.
К.Ф.-М.Н., доцент		Московкин Д. Л.
К.Т.Н., доцент		Земсков Ю.В.

(ученая степень, ученое звание, фамилия и инициалы разработчиков)

И.о. заведующего кафедрой № 8 «Прикладной математики и информатики»

Д.Т.Н., доцент		Костин Г.А.
----------------	---	-------------

(ученая степень, ученое звание, фамилия и инициалы заведующего кафедрой)

Программа согласована:

Руководитель ОПОП

Д.Т.Н., доцент		Костин Г.А.
----------------	--	-------------

(ученая степень, ученое звание, фамилия и инициалы руководителя ОПОП)

Программа рассмотрена и одобрена на заседании Учебно-методического совета Университета « 20 » октября 2021 года, протокол № 2.