

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
(МИНТРАНС РОССИИ)
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА
(РОСАВИАЦИЯ)
ФГБОУ ВО «САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ»
(ФГБОУ ВО СПбГУ ГА)

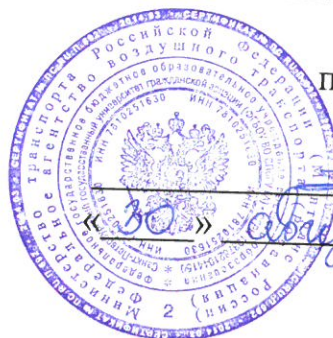
УТВЕРЖДАЮ

Первый

проректор-проректор
по учебной работе

Н.Н. Сухих

2017 года



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

**Объектно-ориентированное программирование
транспортных систем**

Направление подготовки
01.03.04 Прикладная математика

Направленность программы (профиль)
Математическое и программное обеспечение систем управления

Квалификация выпускника
бакалавр

Форма обучения
очная

Санкт-Петербург
2017

1 Цели освоения дисциплины

Целью освоения дисциплины «Объектно-ориентированное программирование транспортных систем» являются приобретение обучающимися знаний, умений и практических навыков в области объектно-ориентированного программирования на C++.

Задачами освоения дисциплины являются:

- изучение объектно-ориентированного синтаксиса языка C++;
- изучение шаблонов классов библиотеки STL и ее возможностей;
- изучение библиотек функций и классов C и C++;
- изучение объектно-ориентированной парадигмы на примере языка C++;
- овладение приемами объектно-ориентированного и функционально-ориентированного программирования на языке C++ для решения математических и естественнонаучных задач.

Дисциплина обеспечивает подготовку выпускника к научно-исследовательскому виду профессиональной деятельности.

2 Место дисциплины в структуре ОПОП ВО

Дисциплина «Объектно-ориентированное программирование транспортных систем» представляет собой дисциплину, относящуюся к Вариативной части Блока 1 «Дисциплины (модули)».

Дисциплина «Объектно-ориентированное программирование транспортных систем» базируется на результатах обучения, полученных при изучении дисциплин: «Операционные системы и сети электронно-вычислительных машин», «Базы данных», «Программирование для электронно-вычислительных машин» и Учебной практике.

Дисциплина «Объектно-ориентированное программирование транспортных систем» является обеспечивающей для дисциплин: «Программирование в сети Internet», «Современные системы программирования», «Интеллектуальные системы», Производственной практики.

Дисциплина изучается в 5 и 6 семестрах.

3 Компетенции обучающегося, формируемые в результате освоения дисциплины

Процесс освоения дисциплины «Объектно-ориентированное программирование транспортных систем» направлен на формирование следующих компетенций:

Перечень и код компетенций	Перечень планируемых результатов обучения по дисциплине
Готовность к самостоятельной работе (ОПК-1)	<p><i>Знать:</i></p> <ul style="list-style-type: none"> – самостоятельно определять технические и эксплуатационные характеристики компьютеров; <p><i>Уметь:</i></p> <ul style="list-style-type: none"> – самостоятельно осуществлять контроль над правильной эксплуатацией программно-аппаратных средств; <p><i>Владеть:</i></p> <ul style="list-style-type: none"> – Навыками настройки и обслуживания программно-аппаратных средств.
Способность использовать современные математические методы и современные прикладные программные средства и осваивать современные технологии программирования (ОПК-2)	<p><i>Знать:</i></p> <ul style="list-style-type: none"> – Основы построения и архитектуры электронно-вычислительных машин и вычислительных систем; <p><i>Уметь:</i></p> <ul style="list-style-type: none"> – использовать методы и средства языка C++; <p><i>Владеть:</i></p> <ul style="list-style-type: none"> – навыками программирования на языке C++.
Готовность применять математический аппарат для решения поставленных задач, способность применить соответствующую процессу математическую модель и проверить ее адекватность, провести анализ результатов моделирования, принять решение на основе полученных результатов (ПК-10)	<p><i>Знать:</i></p> <ul style="list-style-type: none"> – основы алгоритмизации, схемы алгоритмов, правила структурного и объектно-ориентированного программирования; <p><i>Уметь:</i></p> <ul style="list-style-type: none"> – разрабатывать схемы алгоритмов и программы на изучаемом языке программирования для решения профессиональных задач; <p><i>Владеть:</i></p> <ul style="list-style-type: none"> – навыками использования стандартного системного и прикладного программного обеспечения для решения профессиональных задач.
Готовность применять знания и навыки управления информацией (ПК-11)	<p><i>Знать:</i></p> <ul style="list-style-type: none"> – Виды информации и способы ее представления в электронно-вычислительных машинах; <p><i>Уметь:</i></p> <ul style="list-style-type: none"> – Осуществлять выбор способа представления информации в соответствии с поставленной задачей; <p><i>Владеть:</i></p> <ul style="list-style-type: none"> – Навыками управления ресурсами электронно-вычислительных машин и вычислительных систем, чтения и проектирования комбинационных

Перечень и код компетенций	Перечень планируемых результатов обучения по дисциплине
	схем.

4 Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 8 зачетных единиц, 288 академических часа.

Наименование	Всего часов	Семестры	
		5	6
Общая трудоемкость дисциплины	288	144	144
Контактная работа:	128	56	72
лекции	64	28	36
практические занятия	60	28	32
семинары	–	–	–
лабораторные работы	–	–	–
курсовой проект (работа)	4	–	4
Самостоятельная работа студента	88	52	36
Промежуточная аттестация:	72	36	36

5 Содержание дисциплины

5.1 Соотнесения тем дисциплины и формируемых компетенций

Темы (разделы) дисциплины	Количество часов	Компетенции				Образовательные технологии	Оценочные средства
		ОПК-1	ОПК-2	ПК-10	ПК-11		
Тема 1. Ввод и вывод в C++.	9		+	+	+	ВК, Л, ПЗ, СРС	ПрЗ
Тема 2. Типы данных в С и C++.	7	+	+	+	+	Л, СРС	ПрЗ
Тема 3. Строки, указатели и одномерные массивы в С.	9	+	+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 4. Синтаксис языков программирования С и C++.	17	+	+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 5. Адресное пространство процесса на С.	9	+	+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 6. Препроцессор С/С++.	9			+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 7. Структурное программи-	13		+	+	+	Л, ПЗ,	У,

Темы (разделы) дисциплины	Количество часов	Компетенции				Образовательные технологии	Оценочные средства
		ОПК-1	ОПК-2	ПК-10	ПК-11		
рование на C/C++.						СРС	ПрЗ
Тема 8. Указатели и ссылки в C++.	9		+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 9. Файловый ввод-вывод в C и C++.	17		+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 10. Обработка исключений.	9		+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 11. Перечисления и структуры в языках: C и C++.	8	+	+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 12. Пространства имен в C++.	6		+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 13. Классы и объекты в C++.	8		+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 14. Перегрузка функций и методов.	6	+	+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 15. Конструкторы классов.	6		+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 16. Дружественные функции и дружественные классы.	6	+	+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 17. Статические поля и статические методы.	6	+	+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 18. Ключевое слово this.	6		+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 19. Строки в C++.	6	+		+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 20. Шаблонный класс std::vector.	6	+	+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 21. Шаблонный класс std::map.	6	+	+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 22. Алгоритмы STL в C++.	6		+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 23. Перегрузка операций в C++.	6	+	+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 24. Правило «трех» в C++.	6	+	+	+	+	Л, ПЗ, СРС	У, ПрЗ
Тема 25. Наследование классов в	10		+	+	+	Л, ПЗ,	У,

Темы (разделы) дисциплины	Количество часов	Компетенции				Образовательные технологии	Оценочные средства
		ОПК-1	ОПК-2	ПК-10	ПК-11		
С++.						СРС	ПрЗ
Тема 26. Виртуальные методы и виртуальные деструкторы.	6	+		+	+	Л, ПЗ, СРС	У, ПрЗ
Всего по дисциплине	212						
Курсовая работа (проект)	4						
Промежуточная аттестация	72						
Итого по дисциплине	288						

ВК – входной контроль, Л– лекция, ПЗ – практическое занятие, СРС – самостоятельная работа студента, У – устный опрос, ПрЗ – практическое задание.

5.2 Темы (разделы) дисциплины и виды занятий

Наименование темы (раздела) дисциплины	Л	ПЗ	С	КР	ЛР	СРС	Всего часов
5 семестр							
Тема 1. Ввод и вывод в С++.	2	2	-	-	-	5	9
Тема 2. Типы данных в С и С++.	2	-	-	-	-	5	7
Тема 3. Строки, указатели и одномерные массивы в С.	2	2	-	-	-	5	9
Тема 4. Синтаксис языков программирования С и С++.	6	6	-	-	-	5	17
Тема 5. Адресное пространство процесса на С.	2	2	-	-	-	5	9
Тема 6. Препроцессор С/С++.	2	2	-	-	-	5	9
Тема 7. Структурное программирование на С/С++.	4	4	-	-	-	5	13
Тема 8. Указатели и ссылки в С++.	2	2	-	-	-	5	9
Тема 9. Файловый ввод-вывод в С и С++.	4	6	-	-	-	7	17
Тема 10. Обработка исключений.	2	2	-	-	-	5	9
Всего за семестр 5	28	28	-	-	-	52	108
Промежуточная аттестация							36
Итого за семестр 5							144

Наименование темы (раздела) дисциплины	Л	ПЗ	С	КР	ЛР	СРС	Всего часов
6 семестр							
Тема 11. Перечисления и структуры в языках: С и С++.	4	2	-	2	-	2	10
Тема 12. Пространства имен в С++.	2	2	-	-	-	2	6
Тема 13. Классы и объекты в С++.	2	2	-	-	-	4	8
Тема 14. Перегрузка функций и методов.	2	2	-	-	-	2	6
Тема 15. Конструкторы классов.	2	2	-	-	-	2	6
Тема 16. Дружественные функции и дружественные классы.	2	2	-	-	-	2	6
Тема 17. Статические поля и статические методы.	2	2	-	-	-	2	6
Тема 18. Ключевое слово this.	2	2	-	-	-	2	6
Тема 19. Строки в С++.	2	2	-	-	-	2	6
Тема 20. Шаблонный класс std::vector.	2	2	-	-	-	2	6
Тема 21. Шаблонный класс std::map.	2	2	-	-	-	2	6
Тема 22. Алгоритмы STL в С++.	2	2	-	-	-	2	6
Тема 23. Перегрузка операций в С++.	2	2	-	-	-	2	6
Тема 24. Правило «трех» в С++.	2	2	-	-	-	2	6
Тема 25. Наследование классов в С++.	4	2	-	-	-	4	10
Тема 26. Виртуальные методы и виртуальные деструкторы.	2	2	-	2	-	2	8
Всего за семестр 6	36	32	-	4	-	36	108
Промежуточная аттестация							36
Итого за семестр 6							144
Итого по дисциплине							288

Л – лекция, ПЗ – практическое занятие, СРС – самостоятельная работа студента, С – семинар, ЛР – лабораторная работа, КР – курсовая работа (проект).

5.3 Содержание дисциплины

Тема 1. Ввод и вывод в C++.

Заголовочные файлы. <stdio>. 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 2. 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 3. 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. <stdio>. Функции: gets, fgets, puts и fputs. Отличия 1-ой от 2-ой и 3-ей от 4-ой.

Тема 4. 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... elseif... else. Тернарная операция. Циклические алгоритмы. 3 вида циклов. Теорема Боба – Джакопини.

Тема 5. 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: дикое и висящее указатели.

Тема 6. 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 7. 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 8. 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 9. 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 10. Обработка исключений.

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

Тема 11. Перечисления и структуры в языках: C и C++.

Перечисления enum. Структуры struct. Объявления в стиле C и в стиле C++. 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 12. Пространства имен в C++.

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. namespacestd.

Тема 13. Классы и объекты в C++.

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 14. Перегрузка функций и методов.

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

Тема 15. 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 16. Дружественные функции и дружественные классы.

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

Тема 17. 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 18. 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 19. 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 20. 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 21. 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 22. 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 23. 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 24. 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(constX&)`. Необходимость перегруженной операции `=`.

Тема 25. 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 26. 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.

5.4 Практические занятия

Номер темы дисциплины	Тематика практических занятий	Трудоемкость (часы)
5 семестр		

Номер темы дисциплины	Тематика практических занятий	Трудоемкость (часы)
1	Практическое занятие 1. Ввод и вывод в C++.	2
3	Практическое занятие 2. Строки, указатели и одномерные массивы в C.	2
4	Практическое занятие 3. Операции в C++ и их приоритет.	2
	Практическое занятие 4. Разветвляющиеся алгоритмы.	2
	Практическое занятие 5. Циклические алгоритмы.	2
5	Практическое занятие 6. Адресное пространство процесса на C.	2
6	Практическое занятие 7. Препроцессор C/C++.	2
7	Практическое занятие 8. Обработка массивов.	2
	Практическое занятие 9. Функции. Рекурсия.	2
8	Практическое занятие 10. 3 способа передачи параметра в функцию.	2
9	Практическое занятие 11. Чтение и запись для текстового и бинарного файла средствами C.	2
	Практическое занятие 12. Чтение и запись для текстового и бинарного файла средствами C++.	2
	Практическое занятие 13. Дозапись в текстовый файл средствами C и C++. Лог-файлы.	2
10	Практическое занятие 14. Обработка исключений.	2
Всего за семестр 5		28
6 семестр		
11	Практическое занятие 1. Структуры в C. Линейные односвязные списки.	2
12	Практическое занятие 2. Пространства имен в C++.	2
13	Практическое занятие 3. Классы и объекты в C++.	2
14	Практическое занятие 4. Перегрузка функций и шаблоны.	2
15	Практическое занятие 5. Конструкторы классов.	2
16	Практическое занятие 6. Дружественные функции и дружественные классы.	2
17	Практическое занятие 7. Статические поля и статические методы.	2
18	Практическое занятие 8. Ключевое слово this.	2
19	Практическое занятие 9. Строки в C++.	2

Номер темы дисциплины	Тематика практических занятий	Трудоемкость (часы)
20	Практическое занятие 10. Работа с одномерными и двумерными массивами с помощью шаблонного класса <code>std::vector</code> .	2
21	Практическое занятие 11. Создание словаря с помощью шаблонного класса <code>std::map</code> .	2
22	Практическое занятие 12. Алгоритмы STL в C++.	2
23	Практическое занятие 13. Перегрузка операций класса комплексных чисел.	2
24	Практическое занятие 14. Правило «трех» в C++. Реализация класса динамической матрицы с помощью одномерного динамического массива.	2
25	Практическое занятие 15. Наследование классов в C++. Создание класса-наследника квадратных матриц порядка 2 как производного от класса произвольных матриц.	2
26	Практическое занятие 16. Виртуальные методы и виртуальные деструкторы.	2
Всего за семестр 6		32
Итого по дисциплине		60

5.5 Лабораторный практикум

Лабораторный практикум учебным планом не предусмотрен.

5.6 Самостоятельная работа

Номер темы дисциплины	Виды самостоятельной работы	Трудоемкость (часы)
5 семестр		
1	Изучение теоретического материала [3, 5, 7-11]	5
2	Изучение теоретического материала [3, 5, 7-11]	5
3	Изучение теоретического материала [1, 2, 7-11]	5
4	Изучение теоретического материала [1, 3, 7-11]	5
5	Изучение теоретического материала [5, 6, 7-11]	5
6	Изучение теоретического материала [1-4, 7-11]	5
7	Изучение теоретического материала [1, 3, 7-11]	5
8	Изучение теоретического материала [2, 7-11]	5

Номер темы дисциплины	Виды самостоятельной работы	Трудо-емкость (часы)
9	Изучение теоретического материала [3-4, 7-11]	7
10	Изучение теоретического материала [1, 7-11]	5
Всего за семестр 5		52
6 семестр		
11	Изучение теоретического материала [4, 7-11] Выполнение курсовой работы (проекта).	2
12	Изучение теоретического материала [2,3,6,7-11] Выполнение курсовой работы (проекта).	2
13	Изучение теоретического материала [1, 3, 7-11] Выполнение курсовой работы (проекта).	4
14	Изучение теоретического материала [3, 5, 7-11] Выполнение курсовой работы (проекта).	2
15	Изучение теоретического материала [2,3,6,7-11] Выполнение курсовой работы (проекта).	2
16	Изучение теоретического материала [3, 5, 7-11] Выполнение курсовой работы (проекта).	2
17	Изучение теоретического материала [2,3,6,7-11] Выполнение курсовой работы (проекта).	2
18	Изучение теоретического материала [1, 3, 7-11] Выполнение курсовой работы (проекта).	2
19	Изучение теоретического материала [4, 5, 7-11] Выполнение курсовой работы (проекта).	2
20	Изучение теоретического материала [2-5, 7-11] Выполнение курсовой работы (проекта).	2
21	Изучение теоретического материала [2-5, 7-11] Выполнение курсовой работы (проекта).	2
22	Изучение теоретического материала [4, 7-11] Выполнение курсовой работы (проекта).	2
23	Изучение теоретического материала [2-5, 7-11] Выполнение курсовой работы (проекта).	2
24	Изучение теоретического материала [1, 3, 7-11] Выполнение курсовой работы (проекта).	2
25	Изучение теоретического материала [4, 6, 7-11] Выполнение курсовой работы (проекта).	4
26	Изучение теоретического материала [4, 6, 7-11] Выполнение курсовой работы (проекта).	2

Номер темы дисциплины	Виды самостоятельной работы	Трудоемкость (часы)
Всего за семестр 6		36
Итого по дисциплине		88

5.7. Курсовые работы (проекты)

Наименование этапа выполнения курсовой работы (проекта)	Трудоемкость (часы)
Этап 1. Выдача задания на курсовую работу (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 курсовой работы (проекта)	2
Защита курсовой работы (проекта)	2
Итого по курсовой работе (проекту):	24
самостоятельная работа студента, отведенная на выполнение курсовой работы (проекта)	20
согласно учебному плану	4

6. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература

1. Огнева, М. В. **Программирование на языке С++: практический курс:** учебное пособие для бакалавриата и специалитета [Электронный ресурс] / М. В. Огнева, Е. В. Кудрина. — М.: Юрайт, 2017. — 335 с. — ISBN 978-5-534-05123-0. — Режим доступа: <http://biblio-online.ru/viewer/04508F33-FB15-49EB-99BF-E1B9FC555F13/programmirovanie-na-yazyke-s-prakticheskiy-kurs>

2. Гниденко, И. Г. **Технологии и методы программирования: учеб. пособие для прикладного бакалавриата** / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — М.: Издательство Юрайт, 2017. — 235 с. — (Серия: Бакалавр. Прикладной курс). — ISBN 978-5-534-02816-4. — Режим доступа: www.biblio-online.ru/book/958BC376-299C-44B4-B3CF-9F0880FF21C1

3. Павловская, Т.А. **Программирование на языке С++** [Электронный ресурс]: учебное пособие / Т.А. Павловская. — Электрон. дан. — Москва: 2016. — 154 с. — Режим доступа: <https://e.lanbook.com/book/100409> . — Загл. с экрана.

б) дополнительная литература

4. Трофимов, В. В. **Алгоритмизация и программирование: учебник для академического бакалавриата** / В. В. Трофимов, Т. А. Павловская; под ред. В. В. Трофимова. — М.: Издательство Юрайт, 2017. — 137 с. — (Серия: Бакалавр. Академический курс. Модуль). — ISBN 978-5-9916-9866-5. — Режим доступа : www.biblio-online.ru/book/8F6D7A82-2198-40F2-8E02-9A1F1CB5893E

5. Тузовский, А. Ф. **Объектно-ориентированное программирование: учеб. пособие для прикладного бакалавриата** / А. Ф. Тузовский. — М.: Издательство Юрайт, 2017. — 206 с. — (Серия: Университеты России). — ISBN 978-5-534-00849-4. — Режим доступа : www.biblio-online.ru/book/5B61CA55-D3ED-4574-977E-B869CAFF31D0

6. Черпаков, И. В. **Основы программирования: учебник и практикум для прикладного бакалавриата** / И. В. Черпаков. — М.: Издательство Юрайт, 2017. — 219 с. — (Серия: Бакалавр. Прикладной курс). — ISBN 978-5-9916-9983-9. — Режим доступа : www.biblio-online.ru/book/1585A8FA-E53D-4789-8107-96795F6F2DF0

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

7. **Microsoft VisualStudio**[Электронный ресурс] — Режим доступа: <https://visualstudio.microsoft.com/ru/vs/community/> свободный (дата обращения: 15.08.2017).

8. **Форум программистов** [Электронный ресурс] — Режим доступа: <http://www.programmersforum.ru/> свободный (дата обращения: 15.08.2017).

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

9. **Единое окно доступа к образовательным ресурсам** [Электронный ресурс]. — Режим доступа: <http://window.edu.ru/> свободный (дата обращения: 15.08.2017).

10. **Электронная библиотека «eLIBRARY.RU»**[Электронный ресурс] **научных публикаций** — Режим доступа: <http://elibrary.ru/>(дата обращения: 15.08.2017).

11. **Электронно-библиотечная система издательства «Лань»**[Электронный ресурс] — Режим доступа: <http://e.lanbook.com/>(дата обращения: 15.08.2017).

7 Материально-техническое обеспечение дисциплины

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

Лицензионное программное обеспечение: Microsoft Windows, Microsoft Office, Microsoft Visual Studio Community.

Информационно-справочные и материальные ресурсы библиотеки СПбГУ ГА.

8 Образовательные и информационные технологии

В структуре дисциплины в рамках реализации компетентностного подхода в учебном процессе используются следующие образовательные технологии: входной контроль, лекции, практические занятия, самостоятельная работа студента.

Входной контроль проводится преподавателем в начале изучения дисциплины с целью коррекции процесса усвоения студентами дидактических единиц. Он осуществляется по вопросам дисциплин, на которых базируется дисциплина «Объектно-ориентированное программирование транспортных систем» (п.2).

Лекция. Составляет основу теоретического обучения в рамках дисциплины и направлена на систематизированное изложение накопленных и актуальных научных знаний.

Практическое занятие: предназначено для отработки навыков использования приобретенных на лекционных занятиях теоретических знаний для решения прикладных и практических задач. На практических занятиях студент получает очередной блок заданий для самостоятельной работы и имеет возможность отчитаться по ранее выполненным заданиям.

Самостоятельная работа студента: является составной частью учебной работы. Ее основной целью является формирования навыка самостоятельного приобретения новых знаний по вопросам теоретического курса, закрепление и углубление полученных знаний, работа с периодическими изданиями и научно-популярной литературой, в том числе находящейся в информационных сетях.

В рамках изучения дисциплины «Объектно-ориентированное программирование транспортных систем» предполагается использовать в качестве информационных технологий среду MicrosoftOffice и MicrosoftVisualStudioCommunity.

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

Фонд оценочных средств дисциплины «Объектно-ориентированное программирование транспортных систем» 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 экзамена.

Фонд оценочных средств дисциплины «Объектно-ориентированное программирование транспортных систем» для текущего включает: устные опросы, курсовую работу и практические задания.

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 – 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 работ, которые приведены в п. 9.3. Написание и защита курсовой работы запланирована на 6 семестр.

Промежуточная аттестация по итогам освоения дисциплины проводится в виде экзамена в 5-6 семестрах. 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

5 семестр

Тема/вид учебных занятий (оценочных заданий), позволяющих студенту продемонстрировать достигнутый уровень сформированности компетенций	Количество баллов		Срок контроля (порядковый номер недели с начала семестра)	Примечание
	минимальное значение	минимальное значение		
Контактная работа				
Лекции 1-14		10,5	1-14	
Практические задания 1-14	37	45,5	1-14	
Устные опросы 1-8	8	12	1-14	
Посещаемость		2	1-14	
Итого по обязательным видам занятий	45	70		
Экзамен	15	30		
Итого по дисциплине	60	100		
<i>Премияльные виды деятельности (для учета при определении рейтинга)</i>				
Научные публикации по темам дисциплины		10		
Участие в конференциях по темам дисциплины		10		
Итого дополнительно премиальных баллов		20		
Всего по дисциплине для рейтинга		120		
Перевод баллов балльно-рейтинговой системы в оценку по «академической» шкале				
Количество баллов по БРС		Оценка (по «академической» шкале)		
90 и более		5 – «отлично»		
75÷89		4 – «хорошо»		
60÷74		3 – «удовлетворительно»		

Тема/вид учебных занятий (оценочных заданий), позволяющих студенту продемонстрировать достигнутый уровень сформированности компетенций	Количество баллов		Срок контроля (порядковый номер недели с начала семестра)	Примечание
	минимальное значение	минимальное значение		
	менее 60		2 – «неудовлетворительно»	

6 семестр

Тема/вид учебных занятий (оценочных заданий), позволяющих студенту продемонстрировать достигнутый уровень сформированности компетенций	Количество баллов		Срок контроля (порядковый номер недели с начала семестра)	Примечание
	минимальное значение	минимальное значение		
Контактная работа				
Лекции 1-18		2,5	1-18	
Практические задания 1-16	28,5	45	1-18	
Устные опросы 1-16	16	20	1-18	
Посещаемость	0,5	2,5	1-18	
Итого по обязательным видам занятий	45	70		
Экзамен	15	30		
Итого по дисциплине	60	100		
Премияльные виды деятельности (для учета при определении рейтинга)				
Научные публикации по теме дисциплины		5		
Участие в конференциях по теме дисциплины		5		
Участие в предметной олимпиаде		5		
Прочее		5		
Итого дополнительно премияльных баллов		20		
Всего по дисциплине (для рейтинга)		120		
Перевод баллов балльно-рейтинговой системы в оценку по «академической» шкале				
Количество баллов по БРС	Оценка (по «академической» шкале)			
90 и более	5 – «отлично»			
75÷89	4 – «хорошо»			
60÷74	3 – «удовлетворительно»			
менее 60	2 – «неудовлетворительно»			

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

5 семестр:

Посещение лекционного занятия с ведением конспекта оценивается в 0,75 баллов. Посещение практических занятий с ведением конспекта оценивается от 0,5 до 0,75 баллов. Выполнение практического задания от 3 до 3,5 баллов. Устный опрос от 1 до 1,5 баллов.

Если обучающийся посещает лекционные и практические занятия он получает дополнительные 2 балла.

6 семестр:

Посещение лекционных занятия с ведением конспекта оценивается в 2,5 балла. Посещение и активная работа на практических занятиях оценивается от

0,5 до 1 балла. Практическое задание от 1 до 1,25 баллов. Устный опрос от 1 до 1,25 баллов.

Если обучающийся посещает лекционные и практические занятия он получает от 0,5 до 2,5 баллов.

9.3 Темы курсовых работ (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 C или C++, 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.

9.4 Контрольные задания 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. Что позволяет создавать библиотека OpenGL?
5. Что такое база данных?
6. Что такое СУБД?
7. Каково отличие БД от СУБД?
8. Приоритет операторов в выражениях.
9. Блоки и правила видимости переменных.
10. Условный оператор и оператор перехода (if, goto).
11. Оператор множественной альтернативы (switch).
12. Цикл while. Прерывание цикла. Переход к следующей итерации.
13. Цикл do ... while. Прерывание цикла. Переход к следующей итерации.
14. Цикл for. Прерывание цикла. Переход к следующей итерации.
15. Основы синтаксиса языка Си, структура консольного приложения.
16. Фундаментальные типы данных (bool, char, int, double).
17. Определение переменных и констант.
18. Оператор sizeof().
19. Выражения, операции, комментарии.
20. Оператор приведения типа.
21. Операторы инкремента и декремента.

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

Критерий	Этапы формирования	Показатель
<i>Готовность к самостоятельной работе (ОПК-1)</i>		
Знать: – самостоятельно определять технические и эксплуатационные характеристики компьютеров;	1 этап формирования	– описывает характеристики электронно-вычислительных машин
	2 этап формирования	– перечисляет параметры технических средств центрального процессора, необходимых для решения профессиональных задач
Уметь: – самостоятельно осуществлять контроль	1 этап формирования	– выбирает и тестирует аппаратные средства вычислительных систем

Критерий	Этапы формирования	Показатель
над правильной эксплуатацией программно-аппаратных средств;	2 этап формирования	– производит самостоятельный анализ различных типов электронно-вычислительных машин с целью выбора наиболее приемлемого варианта в соответствии с решаемой задачей
<i>Владеть:</i> – Навыками настройки и обслуживания программно-аппаратных средств.	1 этап формирования	– Объясняет принципы настройки и обслуживания программно-аппаратных средств
	2 этап формирования	– Настраивает программно-аппаратные средства
<i>Способность использовать современные математические методы и современные прикладные программные средства и осваивать современные технологии программирования (ОПК-2)</i>		
<i>Знать:</i> – Основы построения и архитектуры электронно-вычислительных машин и вычислительных систем;	1 этап формирования	– Описывает архитектуру электронно-вычислительных машин и вычислительных систем
	2 этап формирования	– Перечисляет составляющие электронно-вычислительных машин и вычислительных систем
<i>Уметь:</i> – использовать методы и средства языка С++;	1 этап формирования	– Использует использовать методы и средства языка С++
	2 этап формирования	– Выбирает оптимальные методы и средства языка С++
<i>Владеть:</i> – навыками программирования на языке С++.	1 этап формирования	– Решает профессиональные задачи с помощью языка С++
	2 этап формирования	– Оценивает методы и алгоритмы, вычислительную сложность разработанных программных продуктов на языке С++
<i>Готовность применять математический аппарат для решения поставленных задач, способность применить соответствующую процессу математическую модель и проверить ее адекватность, провести анализ результатов моделирования, принять решение на основе полученных результатов (ПК-10)</i>		
<i>Знать:</i> – основы алгоритмизации, схемы алгорит-	1 этап формирования	– Перечисляет правила составления схем алгоритмов
	2 этап	– Перечисляет правила струк-

Критерий	Этапы формирования	Показатель
мов, правила структурного и объектно-ориентированного программирования;	формирования	турного и объектно-ориентированного программирования
<i>Уметь:</i> – разрабатывать схемы алгоритмов и программы на изучаемом языке программирования для решения профессиональных задач;	1 этап формирования	– Называет входные и выходные параметры алгоритма при решении практического задания, описывает используемые математические методы или стандартные алгоритмы
	2 этап формирования	– Демонстрирует умение строить алгоритм для решения задачи и разрабатывать программу на изучаемом алгоритмическом языке
<i>Владеть:</i> – навыками использования стандартного системного и прикладного программного обеспечения для решения профессиональных задач.	1 этап формирования	– Называет системные и прикладные программы, дает им краткую характеристику
	2 этап формирования	– Демонстрирует навык использования данных системных и прикладных программ для решения профессиональных задач
<i>Готовность применять знания и навыки управления информацией (ПК-11)</i>		
<i>Знать:</i> – Виды информации и способы ее представления в электронно-вычислительных машинах;	1 этап формирования	– Перечисляет типы данных электронно-вычислительных машин и вычислительных систем: числовые, нечисловые
	2 этап формирования	– Воспроизводит основной состав команд электронно-вычислительных машин
<i>Уметь:</i> – Осуществлять контроль над правильной эксплуатацией программно-аппаратных средств	1 этап формирования	– Описывает назначение постоянных записывающих устройств
	2 этап формирования	– Определяет проблемы реализации распределенных вычислений
<i>Владеть:</i> – Навыками настройки и	1 этап формирования	– Описывает уровни физических устройств

Критерий	Этапы формирования	Показатель
обслуживания программно-аппаратных средств	2 этап формирования	– Категоризирует комбинационные схемы, оценивает их оптимальность

Характеристики шкалы оценивания приведены ниже.

1. Шкала оценивания курсовой работы (проекта) показана в таблице, приведенной ниже:

Наименование этапа выполнения курсовой работы (проекта)	Максимальное количество баллов	Шкала оценивания
Этап 1. Выдача задания на курсовую работу (проект)	–	–
Этап 2. Выполнение раздела «Введение»	10	1-2 балла снимаются за незначительные ошибки в программировании, 3 балл снимается за отсутствие полного решения поставленной задачи, 0,5 балла снимается за отсутствие вывода, 0,3 балла снимается за некорректный вывод, 0,2 балла снимается за неполный вывод, 0,2 балла снимается за допущенные грамматические ошибки
Этап 3. Выполнение раздела «Основная часть»	30	
Этап 4. Выполнение разделов «Заключение», «Выводы».	10	
Этап 5. Оформление курсовой работы (проекта)	10	1-3 балла снимаются за небрежность оформления текста, 1-2 балла снимаются за небрежность оформления использованных источников
Своевременность выполнения	10	За каждый просроченный день по неуважительной причине снимается 0,5 балла.
Итого выполнение курсовой работы (проекта)	70	
Защита курсовой работы (проекта)	30	5 баллов – исследовательский характер; 5 баллов – актуальность работы; 10

Наименование этапа выполнения курсовой работы (проекта)	Максимальное количество баллов	Шкала оценивания
		баллов – ответы на вопросы четкие, ясные и полные; 5 баллов – системная интерпретация полученных в курсовой работе (проекту) результатов; 5 баллов – грамотное ведение полемики.
Всего по курсовой работе (проекту):	100	
Перевод баллов балльно-рейтинговой системы в оценку по «академической» шкале		
Количество баллов по БРС	Оценка (по «академической» шкале)	
90 и более	5 – «отлично»	
75÷89	4 – «хорошо»	
60÷74	3 – «удовлетворительно»	
менее 60	2 – «неудовлетворительно»	

1. Максимальное количество баллов за экзамен – 30. Минимальное количество баллов за экзамен – 15 баллов.
2. При наборе менее 15 баллов – экзамен не сдан по причине недостаточного уровня знаний.
3. Экзаменационная оценка выставляется как сумма набранных баллов за ответы на вопросы билета и за решение задачи.
4. Ответы на вопросы билета оцениваются следующим образом:
 - 1 балл: отсутствие продемонстрированных знаний и компетенций в рамках образовательного стандарта (нет ответа на вопрос) или отказ от ответа;
 - 2 балла: нет удовлетворительного ответа на вопрос, демонстрация фрагментарных знаний в рамках образовательного стандарта, незнание лекционного материала;
 - 3 балла: нет удовлетворительного ответа на вопрос, много наводящих вопросов, отсутствие ответов по основным положениям вопроса, незнание лекционного материала;
 - 4 балла: ответ удовлетворительный, оценивается как минимально необходимые знания по вопросу, при этом студентом продемонстрировано хотя бы минимальное знание всех разделов вопроса в пределах лекционного материала. При этом студентом демонстрируется достаточный объем знаний в рамках образовательного стандарта;

- 5 баллов: ответ удовлетворительный, достаточные знания в объеме учебной программы, ориентированные на воспроизведение; использование научной (технической) терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
 - 6 баллов: ответ удовлетворительный, студент достаточно ориентируется в основных аспектах вопроса, демонстрирует полные и систематизированные знания в объеме учебной программы;
 - 7 баллов: ответ хороший (достаточное знание материала), но требовались наводящие вопросы, студент демонстрирует систематизированные, глубокие и полные знания по всем разделам учебной программы;
 - 8 баллов: ответ хороший, ответом достаточно охвачены все разделы вопроса, единичные наводящие вопросы; студент демонстрирует способность самостоятельно решать сложные проблемы в рамках учебной программы;
 - 9 баллов: систематизированные, глубокие и полные знания по всем разделам учебной программы; студент демонстрирует способность самостоятельно и творчески решать сложные проблемы в нестандартной ситуации в рамках учебной программы;
 - 10 баллов: ответ на вопрос полный, не было необходимости в дополнительных (наводящих вопросах); студент демонстрирует систематизированные, глубокие и полные знания по всем разделам учебной программы, а также по основным вопросам, выходящим за ее пределы.
2. Решение задачи оценивается следующим образом:
- 10 баллов: задание выполнено на 91-100 %, решение и ответ аккуратно оформлены, выводы обоснованы, дана правильная и полная интерпретация выводов, студент аргументировано обосновывает свою точку зрения, уверенно и правильно отвечает на вопросы преподавателя;
 - 9 баллов: задание выполнено на 86-90 %, решение и ответ аккуратно оформлены, выводы обоснованы, дана правильная и полная интерпретация выводов, студент аргументировано обосновывает свою точку зрения, правильно отвечает на вопросы преподавателя;
 - 8 баллов: задание выполнено на 81-85 %, ход решения правильный, незначительные погрешности в оформлении; правильная, но не полная интерпретация выводов, студент дает верные, но не полные ответы на вопросы преподавателя, испытывает некоторые затруднения в интерпретации полученных выводов;
 - 7 баллов: задание выполнено на 74-80 %, ход решения правильный, значительные погрешности в оформлении; правильная, но не полная интерпретация выводов, студент дает правильные, но не полные ответы на вопросы преподавателя, испытывает определенные затруднения в интерпретации полученных выводов;
 - 6 баллов: задание выполнено 66-75 %, подход к решению правильный, есть ошибки, оформление с незначительными погрешностями, неполная интерпретация выводов, не все ответы на вопросы преподавателя правильные, не способен интерпретировать полученные выводы;

- 5 баллов: задание выполнено на 60-65 %, подход к решению правильный, есть ошибки, значительные погрешности при оформлении, неполная интерпретация выводов, не все ответы на вопросы преподавателя правильные, не способен интерпретировать полученные выводы;
- 4 балла: задание выполнено на 55-59 %, подход к решению правильный, есть ошибки, значительные погрешности при оформлении, неполная интерпретация выводов, не все ответы на вопросы преподавателя правильные, не способен интерпретировать полученные выводы;
- 3 балла: задание выполнено на 41-54 %, решение содержит грубые ошибки, неаккуратное оформление работы, неправильная интерпретация выводов, студент дает неправильные ответы на вопросы преподавателя;
- 2 балла: задание выполнено на 20-40 %, решение содержит грубые ошибки, неаккуратное оформление работы, выводы отсутствуют; не может прокомментировать ход решения задачи, дает неправильные ответы на вопросы преподавателя;
- 1 балл: задание выполнено менее, чем на 20 %, решение содержит грубые ошибки, студент не может прокомментировать ход решения задачи, не способен сформулировать выводы по работе.

9.6 Типовые контрольные задания для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины

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 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.

2. 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.

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. Необходимость виртуального деструктора базового класса.
5. Множественное наследование. Вызов конструкторов и деструкторов базовых классов.

Типовые вопросы для проведения промежуточной аттестации в форме экзамена (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 экзамена (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.

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. `namespacestd`.
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`.
39. Методы класса `std::map`.
40. Ассоциативные массивы: уникальные и неуникальные ключи, значения.
41. Алгоритмы в C++.
42. Заголовочный файл `<algorithm>`: функции `max`, `swap`, `swap_ranges`, `transform`, `random_shuffle`, `sort`.
43. Заголовочный файл `<numeric>` – функция `accumulate`.
44. Перегрузка операций в C++.
45. Перегруженные операции как методы класса.
46. Перегруженные операции как дружественные функции.
47. Унарные и бинарные операции.
48. Правило «трех» в C++.
49. Динамическое выделение и возврат памяти в C++
50. Утечки памяти и необходимость деструктора.
51. Вызов деструктора.
52. Динамическое создание объекта в куче и необходимость `delete`.
53. Особенности деструктора.
54. Необходимость конструктора копирования.
55. Формы конструктора копирования: `X(X&)` и `X(constX&)`.
56. Необходимость перегруженной операции `=`.
57. Наследование классов в C++.
58. Ключевое слово `protected`.
59. Модификаторы наследования: `public`, `private` и `protected`.
60. Права доступа при наследовании.
61. Конструкторы производных классов и вызов конструкторов базовых классов.
62. Переопределение методов в производных классах.
63. Виртуальные методы.
64. Чисто виртуальные методы.
65. Абстрактные классы.
66. Использование указателей на базовый класс при создании динамического объекта производного класса.
67. Необходимость виртуального деструктора базового класса.

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.

10. Методические рекомендации для обучающихся по освоению дисциплины

Основными видами учебных занятий по дисциплине «Объектно-ориентированное программирование транспортных систем» являются лекции, практические занятия.

Лекции являются одним из важнейших видов учебных занятий и составляют основу теоретической подготовки обучающихся по дисциплине «Объектно-ориентированное программирование транспортных систем». Они должны давать систематизированные основы научных знаний по дисциплине, концентрировать внимание студентов на наиболее сложных, проблемных вопросах, стимулировать их активную познавательную деятельность и способствовать формированию творческого мышления.

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. Лекции должны носить, как правило, проблемный характер. Основным методом в лекции выступает устное изложение лектором учебного материала.

Практическим занятиям предшествует лекции и целенаправленная самостоятельная подготовка студентов, поэтому темы практических занятий и практических заданий заранее сообщаются обучающимся для того, чтобы они имели возможность подготовиться и проработать соответствующие теоретические вопросы дисциплины.

В начале каждого практического занятия преподаватель кратко доводит до обучающихся его цель и задачи и обращает внимание обучающихся на наиболее сложные вопросы, относящиеся к изучаемой теме.

После проведения любого вида занятия студентам выдаются задания на самостоятельную работу. Выдаваемые задания являются частью учебного материала, который студенты должны освоить за время изучения дисциплины.

Самостоятельная работа студента является составной частью учебной работы. Ее основной целью является формирования навыка самостоятельного приобретения знаний по некоторым не особо сложным вопросам теоретического курса, закрепление и углубление полученных знаний, самостоятельная работа со справочниками, периодическими изданиями и научно-популярной литературой, в том числе находящимися в глобальных компьютерных сетях

При изучении тем дисциплины «Объектно-ориентированное программирование транспортных систем» обучающимся необходимо: ознакомиться с изложенным теоретическим материалом; акцентировать внимание на основных понятиях каждой конкретной темы выполнить задания на самостоятельную работу; подготовиться к сдаче промежуточной аттестации в виде экзамена (5 -6 семестры).

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 Информатики

« 13 » января 2016 года, протокол № 8.

Разработчики

к.ф.-м.н.

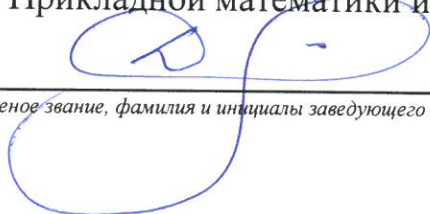


Московкин Д. Л.

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

Заведующий кафедрой № 8 Прикладной математики и информатики

к.т.н., доцент



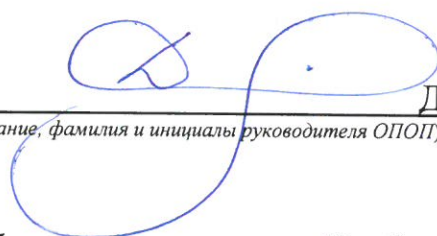
Далингер Я.М.

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

Программа согласована:

Руководитель ОПОП

к.т.н., доцент



Далингер Я.М.

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

Программа рассмотрена и одобрена на заседании Учебно-методического совета Университета « 20 » января 2016 года, протокол № 3.

С изменениями и дополнениями от «30» августа 2017 года, протокол № 10 (в соответствии с Приказом от 5 апреля 2017 г. № 301 «Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры»).