


ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«КАМЧАТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КамчатГТУ»)

Факультет информационных технологий

Кафедра «Системы управления»

УТВЕРЖДАЮ

Декан ФИТ

 И.А. Рыбка

«27» 03 2020г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Объектно-ориентированное программирование»

направление подготовки
09.03.04 «Программная инженерия»
(уровень бакалавриата)

направленность (профиль)
«Разработка программно-информационных систем»

Петропавловск-Камчатский
2020

Рабочая программа разработана в соответствии с ФГОС ВО по направлению подготовки 09.03.04 «Программная инженерия», профиль «Разработка программно-информационных систем», и учебного плана ФГБОУ ВО «КамчатГТУ».

Составитель рабочей программы:
доцент кафедры СУ, к.т.н.



Луковенкова О.О

Рабочая программа рассмотрена на заседании кафедры «Системы управления»
Протокол № 8 от «27» 03 2020 года.

«27» 03 2020 г.



(подпись)

и.о. Заведующий кафедрой
И.А. Рычка
(Ф.И.О.)

1. Цели и задачи учебной дисциплины

Учебная дисциплина «Объектно-ориентированное программирование» относится к вариативной части Блока 1 «Дисциплины (Модули)» основной профессиональной образовательной программы по направлению подготовки 09.03.04 «Программная инженерия», профиль «Разработка программно-информационных систем», предусмотренной учебным планом ФГОУ ВПО «КамчатГТУ».

Цель изучения дисциплины:

Ознакомление студентов с парадигмой и технологией объектно-ориентированного программирования (ООП) и сопутствующих технологий программирования, а так же – обучение студентов основам ООП на языке программирования высокого уровня C++.

Задача изучения дисциплины:

Дать студенту теоретические и практические знания по принципам объектно-ориентированного проектирования и разработки объектно-ориентированного программного обеспечения ЭВМ.

В результате изучения дисциплины:

- студент должен знать основные концепции объектно-ориентированного программирования, лексические и синтаксические основы объектно-ориентированного языка программирования C++, принципы построения классов и объектов, конструкторы и деструкторы, виртуальные методы и классы, абстрактные классы, принципы и виды наследования классов, шаблоны классов, обработку исключительных ситуаций, методику объектно-ориентированного анализа и проектирования.
- студент должен уметь создавать объектно-ориентированные программы с применением классов на языке C++, создавать иерархию классов на C++, использовать полиморфизм, проектировать с учетом множественного наследования, создавать шаблоны функций и классов, использовать механизм обработки исключений, использовать библиотеку потоковых классов, производить отладку программного кода в среде разработчика Visual C++.
- студент должен приобрести навыки по объектно-ориентированному проектированию и разработке объектно-ориентированного программного кода в современных операционных системах.

Компетенции обучающегося, формируемые в результате освоения дисциплины

Дисциплина «Объектно-ориентированное программирование» направлена на освоение следующих компетенций основной профессиональной образовательной программы по направлению 09.03.04 «Программная инженерия» федерального государственного образовательного стандарта высшего образования:

- готовность применять основные методы и инструменты разработки программного обеспечения (ПКС-5).

Планируемые результаты обучения при изучении дисциплины, соотнесенные с планируемыми результатами освоения образовательной программы, представлены в таблице.

Код компетенции	Планируемые результаты освоения образовательной программы	Планируемый результат обучения по дисциплине	Код показателя освоения
-----------------	---	--	-------------------------

ПКС-5	готовность применять основные методы и инструменты разработки программного обеспечения	Знать: – современные инструменты разработки программного обеспечения	З(ПКС-5)1
		Уметь: – применять основные методы разработки программного обеспечения	У(ПКС-5)1
		Владеть: – навыки по применению основных методов и инструментов разработки программного обеспечения	В(ПКС-5)1

2. Место дисциплины в структуре образовательной программы

Связь с предшествующими дисциплинами

Дисциплина изучается студентом после прохождения курса «Информатика и программирование».

Связь с последующими дисциплинами

Данная дисциплина создает знания и навыки по разработке объектно-ориентированного программного обеспечения, которые в дальнейшем используются в дисциплинах «Алгоритмы и структуры данных», «Конструирование программного обеспечения», а также позволяют успешно выполнить выпускную квалификационную работу.

3. Содержание дисциплины

3.1. Тематический план дисциплины

Наименование разделов и тем	Всего часов	Аудиторные занятия	Контактная работа по видам учебных занятий			Самостоятельная работа	Формы текущего контроля	Итоговый контроль знаний по дисциплине
			ЛК	ПЗ	ЛЗ			
Раздел 1. Основные положения объектно-ориентированной парадигмы	22	7	5	-	2	15		
Тема 1. Введение	4	1	1	-	0	3	Опрос	
Тема 2. Основные концепции объектно-ориентированного программирования	4	1	1	-	0	3	Опрос	
Тема 3. Основные модели объектно-ориентированного программирования	4	1	1	-	0	3	Опрос	

Тема 4. Современные технологии разработки и тестирования программного обеспечения	4	1	1	-	2	3	Опрос, РЗ	
Тема 5. Жизненный цикл программных продуктов	4	1	1	-	0	3	Опрос	
Раздел 2. Объектно-ориентированное программирование на C++	54	27	9	-	18	27		
Тема 1. Реализация концепции объектно-ориентированного программирования в языке программирования C++	6	3	1	-	2	3	Опрос, РЗ	
Тема 2. Общий синтаксис языка C++	6	3	1	-	2	3	Опрос, РЗ	
Тема 3. Структуры данных	6	3	1	-	2	3	Опрос, РЗ	
Тема 4. Общие принципы создания классов	6	3	1	-	2	3	Опрос, РЗ	
Тема 5. Конструкторы и деструкторы	6	3	1	-	2	3	Опрос, РЗ	
Тема 6. Наследование классов	6	3	1	-	2	3	Опрос, РЗ	
Тема 7. Шаблоны классов	6	3	1	-	2	3	Опрос, РЗ	
Тема 8. Поточные классы ввода-вывода	6	3	1	-	2	3	Опрос, РЗ	
Тема 9. Обработка исключительных ситуаций	6	3	1	-	2	3	Опрос, РЗ	
Раздел 3. WinAPI	32	17	3	-	14	15		
Тема 1. Разработка проектов на основе функций Win API	6	1	1	-	0	5	РЗ	
Тема 2. Разработка проектов на основе классов MFC	18	13	1	-	12	5	Опрос, РЗ	
Тема 3. Отладка программ в Visual C++	8	3	1	-	2	5	Опрос, РЗ	
Экзамен	36	-	-	-	-	-		36
Всего	144	51	17	-	34	57		36

*ПЗ – практическое задание, РЗ – решение задач, КС – конкретная ситуация

3.2. Описание содержания дисциплины

Раздел 1. Основные положения объектно-ориентированной парадигмы

Лекция 1.1 Введение.

Предмет дисциплины и ее задачи. Роль и место задач разработки программного обеспечения в интегрированных производственных комплексах, автоматизированных системах управления техническими объектами. Структура и содержание дисциплины, ее связь с другими дисциплинами специальности. Обзор рекомендуемой литературы.

Лекция 1.2 Основные концепции объектно-ориентированного программирования.

Факторы, обусловившие появление и содержание концепции ООП. Основные идеи ООП: использование объекта в качестве основной компоненты программы и децентрализация управления, реализуемое представлением программы как описания взаимодействия объектов. Инкапсуляция, наследование, полиморфизм. Объектно-ориентированный подход к разработке программ. Место и роль ООП в теории и практике разработки программных систем.

Лекция 1.3 Основные модели объектно-ориентированного программирования.

Объект как совокупность данных и набора операций. Семантика объекта. Представление данных. Классификация методов: конструкторы, деструкторы, селекторы и модификаторы. Классы объектов: назначение и семантика. Класс как абстракция совокупности объектов. Классы и абстрактные типы данных. Объекты как экземпляры классов. Основные действия с объектами: создание, инициализация, использование, уничтожение. Отношение наследования для классов. Простое и множественное наследование. Иерархия классов.

Лекция 1.4 Современные технологии разработки и тестирования программного обеспечения.

Объектно-ориентированный стиль программирования. Инструментальные средства автоматизации проектирования программных систем (CASE-средства). Графический подход к решению проблемы автоматизации разработки программного обеспечения. Требования качеству современных программных средств. Среда Visual C++ как пример современного автоматизированного средства разработки. Основные элементы управления в Visual C++ , создание и разработка проекта. Типы мастеров проектов. Debug и Release проекты.

Лабораторная работа 1.1 Изучение основных элементов управления среды разработчика. Разработка первого проекта.

Основные элементы управления проектом. Состав программы. Включение препроцессора. Использование функции main. Ввод данных, решение математического выражения и вывод результата на экран.

Цель работы: Изучение принципов разработки программы в среде Visual C++.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №1.

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;
3. Результаты работы программ (копии экранов);
4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лекция 1.5 Жизненный цикл программных продуктов.

Процесс разработки программ. Основные фазы проектирования программного обеспечения. Структура жизненного цикла. Модели жизненного цикла. Этапы разработки объектно-ориентированной системы.

СРС по разделу 1. Проработка теоретического материала по следующим темам:

1. Основные концепции объектно-ориентированного программирования [1,5,7].
2. Основные модели объектно-ориентированного программирования [1,5,7].
3. Современные технологии разработки и тестирования программного обеспечения [1].
4. Жизненный цикл программных продуктов [1].

Раздел 2. Объектно-ориентированное программирование на С++

Лекция 2.1 Реализация концепции объектно-ориентированного программирования в языке программирования С++.

История возникновения языка С++. Место языка С++ во внедрении и развитии средств ООП. Значение языка С++ для профессиональной подготовки программистов. Структура программы. Ввод и вывод информации.

Лабораторная работа 2.1 Разработка программ с операторами, указателями и массивами.

Разработка программ, использующих операторы, указатели и массивы.

Цель работы: Изучение принципов разработки программы, использующей операторы, указатели и массивы.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №2.

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;
3. Результаты работы программ (копии экранов);
4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лекция 2.2 Общий синтаксис языка С++.

Типы данных и операции С++. Операторы языка. Указатели и адресная арифметика. Массивы и функции. Выделение и освобождение памяти. Макросы. Подстановка лексем. Включение файлов. Заголовочные файлы. Способы размещения программы в файлах.

Лабораторная работа 2.2 Разработка программ с операциями ввода-вывода в файл, динамическим выделением памяти и внешними функциями.

Разработка программ, использующих файловый ввод-вывод, динамическое выделение памяти и внешние функции.

Цель работы: Изучение принципов разработки программ с операциями ввода-вывода в файл, динамическим выделением памяти и внешними функциями.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №3.

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;
3. Результаты работы программ (копии экранов);
4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лекция 2.3 Структуры данных.

Структуры и объединения. Доступ к элементам структуры. Объединения. Битовые поля структур и объединений. Динамические структуры данных: стеки, очереди, списки, бинарные деревья.

Лабораторная работа 2.3 Разработка программы, использующей структуры.

Разработка программы, использующей структуры.

Цель работы: Изучение принципов разработки программы, использующей структуры.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №4.

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;
3. Результаты работы программ (копии экранов);
4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лекция 2.4 Общие принципы создания классов.

Способы реализации инкапсуляции. Определение класса. Личная и общая части определения класса. Функции-элементы класса и функции-друзья. Объекты класса. Статические члены объектов класса. Вложенные и локальные классы. Примеры описания и использования классов.

Лабораторная работа 2.4 Разработка программы, использующей общие принципы построения классов.

Разработка программы, использующей элементарные классы.

Цель работы: Изучение принципов разработки программы, использующей элементарные классы.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №5.

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;
3. Результаты работы программ (копии экранов);
4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лекция 2.5 Конструкторы и деструкторы.

Принципы создания и уничтожения объектов. Свойства конструкторов и деструкторов. Конструкторы по умолчанию, с параметрами и копирования. Выделение и уничтожение динамических объектов класса.

Лабораторная работа 2.5 Разработка программы, использующей конструкторы и деструкторы.

Разработка программы, использующей конструкторы и деструкторы классов.

Цель работы: Изучение принципов разработки программы, использующей конструкторы и деструкторы классов.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №6.

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;
3. Результаты работы программ (копии экранов);

4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лекция 2.6 Наследование классов.

Базовый и производный классы. Функции-элементы и функции-друзья. Правила доступа к элементам производного класса. Иерархия классов. Одиночное и множественное наследование. Виртуальные базовые классы. Особенности доступа при множественном наследовании. Виртуальные функции. Чистые виртуальные функции и абстрактные базовые классы.

Лабораторная работа 2.6 Разработка программы, использующей механизм наследования классов.

Разработка программы, использующей механизм наследования классов.

Цель работы: Изучение принципов разработки программы, использующей механизм наследования классов.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №7.

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;
3. Результаты работы программ (копии экранов);
4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лекция 2.7 Шаблоны классов.

Шаблоны классов и функций. Наследование шаблонных классов. Правила отождествления параметров шаблона. Контейнерные классы. Применение шаблонных классов для создания контейнерных классов.

Лабораторная работа 2.7 Разработка программы, использующей шаблоны классов.

Разработка программы, использующей шаблоны классов.

Цель работы: Изучение принципов разработки программы, использующей шаблоны классов.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №8.

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;
3. Результаты работы программ (копии экранов);
4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лекция 2.8 Потокосые классы ввода-вывода.

Использование функций ввода-вывода языка C++. Форматы ввода-вывода. Стандартный ввод-вывод. Ввод-вывод в файл. Строковый ввод-вывод. Ошибочные состояния потоков ввода-вывода.

Лабораторная работа 2.8 Разработка программы, использующей потоковые классы ввода-вывода в файл.

Разработка программы, осуществляющей ввод-вывод в файл с помощью потоков.

Цель работы: Изучение принципов разработки программы, осуществляющей ввод-вывод в файл с помощью потоков.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №9.

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;
3. Результаты работы программ (копии экранов);
4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лекция 2.9 Обработка исключительных ситуаций.

Анализ различных моделей обработки исключительных ситуаций. Стандартные средства контроля подтверждений. Проверка предусловий и постусловий, вычисления инвариантов. Контроль асинхронных событий. Реализация модели обработки синхронных ситуаций с завершением в C++. Возбуждение ситуации, Описание блоков с контролем и реакцией на ситуации. Система классов для описания исключительных ситуаций.

Лабораторная работа 2.9 Разработка программы, использующей технологии обработки исключительных ситуаций.

Разработка программы, осуществляющей обработку исключений.

Цель работы: Изучение принципов разработки программы, осуществляющей обработку исключений.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №10.

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;
3. Результаты работы программ (копии экранов);
4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

СРС по разделу 2. Проработка теоретического материала по следующим темам:

1. Реализация концепции объектно-ориентированного программирования в языке программирования C++ [1-7].
2. Общий синтаксис языка C++ [1-7].
3. Структуры данных [1-7].
4. Общие принципы создания классов [1, 3-7].
5. Конструкторы и деструкторы [1, 3-7].
6. Наследование классов [1, 3-7].
7. Шаблоны классов [1, 3-7].
8. Поточковые классы ввода-вывода [1].
9. Обработка исключительных ситуаций [1].

Раздел 3. WinAPI

Лекция 3.1 Разработка проектов на основе функций Win API.

Виды интерфейсов API. Принципы разработки проекта на основе Win API. Типы мастеров проекта. Технологии разработки консольных и оконных интерфейсов. Использование в проекте DLL.

Лекция 3.2 Разработка проектов на основе классов MFC.

Типы мастеров проекта. Изменение установок проекта. Иерархия классов MFC. Функции MFC. Обработка сообщений в MFC. Анализ классов проекта. Создание ресурсов.

Включение инструмента ClassWizard. Типы диалогов. Введение элементов в диалоговое окно. Виды применяемых элементов. Управление мышью. Поддержка вывода на принтер. Запись в файл.

Лабораторная работа 3.1 Разработка текстового редактора.

Разработка программы-прототипа текстового редактора.

Цель работы: Изучение принципов разработки текстового редактора в среде Microsoft Visual C++.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №11.

Содержание отчета:

Цель работы;

Исходные тексты разработанных программ;

Результаты работы программ (копии экранов);

Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лабораторная работа 3.2 Включение ресурсов в приложение.

Разработка программы-прототипа с различными ресурсами.

Цель работы: Изучение принципов включения ресурсов в разрабатываемое приложение.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №12.

Содержание отчета:

Цель работы;

Исходные тексты разработанных программ;

Результаты работы программ (копии экранов);

Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лабораторная работа 3.3 Создание приложения на базе диалогового окна.

Разработка программы на базе диалогового окна.

Цель работы: Изучение принципов создания приложения на базе диалогового окна.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №13.

Содержание отчета:

Цель работы;

Исходные тексты разработанных программ;

Результаты работы программ (копии экранов);

Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лекция 3.3 Отладка программ в Visual C++.

Структура отладчика программ. Установка метки. Окна просмотра промежуточных результатов. Пошаговое выполнение программы.

Лабораторная работа 3.4 Отладка программ с помощью встроенных средств отладчика.

Устранение ошибок с помощью отладчика программ.

Цель работы: Изучение принципов отладки программ в Visual C++.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №14.

Содержание отчета:

Цель работы;
Исходные тексты разработанных программ;
Результаты работы программ (копии экранов);
Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

СРС по разделу 3. Проработка теоретического материала по следующим темам:

1. Разработка проектов на основе функций Win API [9-11].
2. Разработка проектов на основе классов MFC [12].
3. Отладка программ в Visual C++ [1].
1. Подготовка к экзамену по перечню примерных вопросов.

3.3. Учебно-методическое обеспечение для самостоятельной работы обучающихся

В целом внеаудиторная самостоятельная работа обучающегося при изучении курса включает в себя следующие виды работ:

- проработка (изучение) материалов лекций;
- чтение и проработка рекомендованной основной и дополнительной литературы;
- подготовка к практическим занятиям и лабораторным работам;
- поиск и проработка материалов из Интернет-ресурсов, периодической печати;
- выполнение домашних заданий в форме творческих (проблемно-поисковых, групповых) заданий, докладов;
- подготовка к текущему и итоговому (промежуточная аттестация) контролю знаний по дисциплине.

Основная доля самостоятельной работы обучающихся приходится на подготовку к практическим занятиям и лабораторным работам, тематика которых полностью охватывает содержание курса. Самостоятельная работа по подготовке к практическим занятиям и лабораторным работам предполагает умение работать с первичной информацией.

4. Фонд оценочных средств

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине «Объектно-ориентированное программирование» представлен в приложении к рабочей программе дисциплины и включает в себя:

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы;
2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания;
3. Типовые контрольные задания или материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций;
4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Примерный перечень вопросов к промежуточной аттестации

1. Факторы, обусловившие появление и содержание концепции ООП.
2. Основные свойства ООП: абстракция, инкапсуляция, наследование, полиморфизм.
3. Основные фазы проектирования программного обеспечения. Структура жизненного цикла.
4. Модели жизненного цикла. Этапы разработки объектно-ориентированной системы.

5. История возникновения языка C++. Общий синтаксис языка C++. Разделители. Использование комментариев. Структура идентификаторов.
6. Этапы подготовки программы на языке C++ к исполнению на ПЭВМ.
7. Зарезервированные слова.
8. Директивы препроцессора.
9. Основные типы данных. Размер данных.
10. Определение переменных. Локальные, статические и др. типы переменных.
11. Арифметические операции. Операции присваивания.
12. Операции отношения. Логические операции. Побитовые операции.
13. Оператор-выражение. Основные виды операторов.
14. Условные операторы.
15. Операторы цикла.
16. Оператор переключения.
17. Указатели и адреса.
18. Описание массива. Массивы в качестве параметров. Строки. Массивы указателей.
19. Определение внешней функции. Вызов функции. Использование и определение параметров. Функция main.
20. Подключение библиотек. Основные библиотеки языка C++.
21. Библиотеки ввода-вывода. Основные функции ввода-вывода.
22. Ввод-вывод в файл, доступ к файлам. Форматы записи данных.
23. Математическая библиотека. Основные функции.
24. Библиотеки для работы с символами и строками. Основные функции.
25. Библиотека общего назначения. Основные функции.
26. Технологии и функции для динамического выделения памяти.
27. Системные библиотеки. Основные функции.
28. Перечисления.
29. Структуры и объединения.
30. Классы: определение, описание, члены класса. Управление доступом к элементам классов, область определения элементов.
31. Конструкторы и деструкторы. Дружественные функции класса.
32. Основы механизма наследования классов.
33. Шаблоны классов.
34. Средства обработки исключений в языке C++.

5. Рекомендуемая литература

Основная литература

1. *Маратулец Ю.В.* Программирование на языке высокого уровня. Рекомендовано ДВ РУМЦ в качестве учебного пособия. Петропавловск-Камчатский: Издательство КамчатГТУ, 2008 - 189 с.

Дополнительная литература

2. *Карпов Б., Баранова Т.* C++. Специальный справочник. С.-Пб.: Питер, 2001. – 479 с.
3. *Климова Л.М.* C++. Практическое программирование. М.: КУДИЦ-ОБРАЗ, 2001. – 587 с.
4. *Маратулец Ю.В.* Основы программирования на языке C++. П.-Камчатский: КамчатГТУ, 2003. – 157 с.
5. *Подбельский В.В.* Язык Си++. М.: Финансы и статистика, 1996 г. – 559 с.
6. *Романов Е.Л.* Практикум по программированию на C++. С.-Пб.: БХВ-Петербург, 2004. – 427 с.
7. *Страуструп Б.* Язык программирования Си++. М.: Радио и связь, 1991. – 352 с.
8. *Финогенов К.Г.* Win32. Основы программирования. М.: Диалог-МИФИ, 2004. - 416 с.

9. *Рихтер Д.* Windows. Создание эффективных Win32-приложений с учетом специфики 64-разрядной версии Windows. С.-Пб.: Питер, 2005. - 624 с.
10. *Д.Круглински, С.Уингоу, Д.Шефферд.* Программирование на Microsoft Visual C++ для профессионалов. С.-Пб.: Питер, 2004 г.
11. *Щупак Ю.А.* Win32 API. Эффективная разработка приложений. С.-Пб.: Питер, 2007.- 572с.
12. *Черносвитов А.* Visual C++ и MFC. С.-Пб.: Питер, 2005 г.

Перечень методических указаний к проведению учебных, лабораторных занятий и самостоятельной работы студентов

1. Марапулец Ю.В. Объектно-ориентированное программирование. Методическое пособие к выполнению лабораторных работ по направлению подготовки бакалавров 09.03.04 «Программная инженерия» по дисциплине «Объектно-ориентированное программирование» для студентов очной формы обучения – Петропавловск-Камчатский: КамчатГТУ, 2015. – 95 с. (электронная форма).

Перечень обучающих, контролирующих компьютерных программ

1. Microsoft Visual C++.

6. Методические указания для обучающихся по освоению дисциплины

Подготовка к лекционным занятиям

Лекции составляют основу теоретического обучения и дают систематизированные основы научных знаний по дисциплине, концентрируют внимание обучающихся на наиболее сложных и узловых вопросах, стимулируют их активную познавательную деятельность и способствуют формированию творческого мышления.

При подготовке к лекции следует предварительно ознакомиться с учебным материалом по теме занятия и при конспектировании лекции акцентировать внимание на новых теоретических положениях и иных данных, не нашедших отражения в учебной литературе.

Для успешного изучения дисциплины студенту рекомендуется систематически готовиться к каждому занятию по следующей схеме:

- повторить материал предыдущей лекции, используя конспекты, учебную и специальную литературу
- ответить на контрольные вопросы по изучаемой теме.

Подготовка к лабораторным занятиям

Лабораторная работа – это выполнение студентами под руководством преподавателя или по инструкции заданий (решение задач, написание программ) с применением персонального компьютера.

В ходе лабораторных работ студенты воспринимают и осмысливают новый учебный материал. Лабораторные занятия носят систематический характер, регулярно следуя за лекционными занятиями. Лабораторные работы выполняются согласно графику, при этом соблюдается принцип индивидуального выполнения работ.

Обучающийся должен подготовить отчет к каждой лабораторной работе, предусмотренной планом.

При подготовке к лабораторным занятиям необходимо заранее изучить методические рекомендации по его проведению. Обратит внимание на цель занятия, на основные вопросы для подготовки к занятию, на содержание темы занятия.

7. Курсовой проект (работа)

В соответствии с учебным планом не предусмотрено курсовое проектирование по дисциплине «Объектно-ориентированное программирование».

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

В учебном процессе используются следующие информационные технологии:

- презентации;
- облачные хранилища для обмена информацией;
- электронные учебники;
- информационные ресурсы сети Интернет.

Программное обеспечение

Для формирования отчетов по лабораторным работам

1. MS Word
2. Latex

Для просмотра электронных документов

1. MS Word
2. MS PowerPoint
3. STDViewer
4. Браузер (Opera, Google Chrome и др.)

Для программирования на C++

1. MS Visual Studio (Community Ed.)

9. Материально-техническое обеспечение дисциплины

На кафедре имеются аудитории для проведения лекционных занятий и компьютерные классы для проведения лабораторных занятий.

10. Дополнения и изменения в рабочей программе

Дополнения и изменения в рабочей программе за ____ / ____ учебный год

В рабочую программу _____
(наименование дисциплины)

для направления _____
(номер направления)

вносятся следующие дополнения и изменения:

Дополнения и изменения внес _____
(должность, Ф.И.О., подпись)

Рабочая программа пересмотрена и одобрена на заседании кафедры

(наименование кафедры)

« ____ » _____ 20__ г.

Заведующий кафедрой _____
(подпись)

(Ф.И.О.)