МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Забайкальский государственный университет»
(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

УЧЕБНЫЕ МАТЕРИАЛЫ

для студентов заочной формы обучения

(с полным сроком обучения)

по дисциплине

«Объектно-ориентированное программирование»

для направления подготовки

09.03.01 Информатика и вычислительная техника

Общая трудоемкость дисциплины – 9 зачетных единиц.

Форма текущего контроля в 3, 4 семестре – контрольная работа.

Форма промежуточного контроля в 3,4 семестре – экзамен.

Чита 2020 г.

Краткое содержание курса

Цель освоения дисциплины – формирование умений, навыков и знаний:

- по объектно-ориентированному анализу, объектно-ориентированному проектированию и объектно-ориентированному программированию;
- использования современных фреимворков для объектно-ориентированных языков общего назначения
- работы с современными средами разработки.

Перечень изучаемых разделов дисциплины

Семестр 3

1. Язык C++.
    1. Перечисления, работа с типами данных. Пространства имён.
    2. Ссылки и указатели. Работа с динамической памятью.
    3. Указатели на функции, лямбда-функции.
    4. Обработка исключительных ситуаций.
2. Декомпозиция. Введение в ООП. Абстрактный тип данных. Алгоритмическая и ОО декомпозиция. UML диаграммы классов.
3. Классы в C++. Инкапсуляция.
    1. Поля, методы, конструкторы, деструктор.
    2. Массивы из объектов и указателей. Работа с динамической памятью. Правило пяти.
    3. Перегрузка операторов. Друзья классов.
    4. Статические члены классов.
    5. Контроль постоянства.
    6. Агрегация, композиция, ассоциация.
4. Стандартная библиотека языка C++.
    1. Классы string, vector, list и др.
    2. Итераторы, совместный цикл, алгоритмы.
    3. Ввод-вывод и др. возможности

5. Наследование.
    1. Модификаторы наследования, области видимости.
    2. Переопределение методов.
    3. Множественное наследование.
    4. Виртуальные и абстрактные методы.
6. Полиморфизм.
    1. Динамический полиморфизм. Абстрактные классы. RTTI.
    2. Шаблоны.
7. Возможности сред разработки Qt Creator и Visual Studio. Структура и принцип работы приложения с GUI. Шаблон проектирования «Модель-Представление».

Семестр 4

8. Введение в фреймворк Qt.
    1. Понятие фреймворка.
    2. Метаклассы. Слоты и сигналы.
    3. Основные элементы интерфейса.
9. Фреймворк Qt: многооконные приложения, таблицы, потоки, файлы и др. возможности фреимворка
10. Языки описания интерфейса пользователя. QML
11. Windows Presentation Foundation
12. Принципы SOLID.
13. Паттерны проектирования.
14. Юнит-тестирование

Материалы курса, слайды лекций, примеры: https://github.com/VetrovSV/OOP

Рекомендуемые онлайн курсы:

Программирование на языке C++

Программирование на языке C++ (продолжение)

Форма текущего контроля

Контрольная работа

Контрольная работа оформляется на скреплённых (не скоросшивателем) листах формата А4, (см. МИ 01-02-2018 Общие требования к построению и оформлению учебной текстовой документации).

Для каждого задания приводится:

- непосредственно задание
- диаграмма классов
- исходный код ,
- скриншот окна программы

Каждое задание начинается с нового листа.

Задания выполняются в течении семестра. Защита контрольных работ проходит на лабораторных занятиях во время сессии. Для защиты нужно иметь контрольную работу (исходные тексты, диаграммы) в электронном виде. Исходный код программ должен соответствовать парадигме ООП.

Рекомендованные среды разработки:

- **Qt Creator**
- **Microsoft Visual Studio**
- JetBrains CLion
- Embarcadero RAD Studio

ПО и сервисы для построения диаграмм:

- draw.io
- Microsoft Visio
- Dia

Дистанционное обучение

- Контрольные работы для проверки нужно загрузить на github.com в отдельный репозиторий с именем: `OOP_sem4`.
- Каждую контрольную работу поместить в отдельную папку с именем `task1, task2` и т.д.
- Отправить ссылку преподавателю для проверки. Если репозиторий закрытый (private), то пригласить в репозиторий пользователя VetrovSV.
- Информация по работе с git и github: https://github.com/VetrovSV/Programming/blob/master/git_lec.pdf

Семестр 3

Задание 1. Простой класс

Классы на выбор:

- Геометрическая фигура. Задание сторон, координат на плоскости. Вычисление площади и периметра.
- Комплексное число. Операторы сложения, вычитания, умножения (на комплексное и действительное число). Вычисление аргумента и модуля.
- Кватернион. Аналогично комплексному числу.
- Вектор. Задаётся своими компонентами. Вычисление длины, углов между осями; операторы сложения и вычитания, умножения на число.
- Время. Сложение, вычитание. Добавление минут, секунд, часов и т.п. Перевод времени в секунды, часы, минуты. Преобразование в строку.
- Дата. Реализовать то же самое, что и для времени.
- Обыкновенная дробь. Операции сложения, вычитания, умножения, деления, сравнения.
- Другой класс по согласованию с преподавателем.

1. Описать АДТ.

2. Представить класс в виде UML диаграммы.

3. Описать класс на C++, C#, Java или любом другом объектно-ориентированном языке программирования (по согласованию с преподавателем). Реализовать методы для доступа и изменения данных, контроль постоянства, конструктор с параметрами. Операторы и генерирование исключительных ситуаций если необходимо.

4. Продемонстрировать работу с классом, с основными методами, операторами. Создать массив из объектов. Записать состояние объектов в файл, загрузить из файла. Программа не обязательна

должна взаимодействовать с пользователем, главная цель — показать пример использования класса.

5. Привести документацию для класса (в формате принятом в языке программирования) описав его назначение, принципы использования, смысл методов и их параметров. Если необходимо привести пример использования класса в документации.

Вопросы

1. Что такое АДТ?
2. Что такое предусловия? Для чего нужны? Что такое постусловия?
3. Что такое класс? Что такое объект?
4. Что такое абстрагирование?
5. Что такое инкапсуляция? Что такое метод? Что такое поле класса? Что такое конструктор?
6. Что такое принцип сокрытия? Что такое «чёрный ящик»?
7. Что такое оператор?
8. Как вызвать метод конкретного объекта находящегося в массиве?
9. Чем отличаются обращения к методам в C++ с использованием объекта, ссылки на объект и указателя на объект?
10. Что такое равенство объектов? Когда объекты идентичны?
11. Чем отличается присваивание объектов от присваивания указателей на объекты?
12. Как обратится к методу или полю объекта находящегося в массиве?
13. Что такое поведение? Что такое состояние?

Ссылки

- Слайды лекции: github.com/VetrovSV/OOP/blob/master/OOP_1.1.pdf

- Слайды с лекции (АДТ и UML): github.com/VetrovSV/OOP/blob/master/OOP_1.0.pdf
- Пример класса: github.com/VetrovSV/OOP/tree/master/examples/simple_class
- Draw.io – создание диаграмм

Задание 2. Наследование

1. Создать UML диаграмму из 3-х (или более) классов имеющих отношение типа наследование.
2. Реализовать классы на C++, C#, Java или любом другом объектно-ориентированном языке программирования(по согласованию с преподавателем). Продемонстрировать работу с ними.
3. Привести документацию для классов описав их назначение, принципы использования, смысл методов и их параметров. При необходимости приведите пример использования классов в документации.

Пример классов для задания:

- Геометрическая фигура на плоскости, квадрат, круг, прямоугольник.
- Шахматная фигура, пешка, ладья, …

Вопросы

1. Что такое наследование? Как оно изображается на UML?
2. Когда стоит использовать наследование?
3. Какие классы называется базовыми и производными?
4. Сколько предков может иметь класс?
5. Что такое множественное наследование? Чем оно опасно? Когда его можно использовать?
6. Что такое перегрузка метода? Что такое переопределение метода?
7. Как работает преобразование типов связанных наследованием?
8. Что такое интерфейс (ООП)?
9. Как влияют области видимости внутри класса на наследуемые методы и поля?

Ссылки

- Слайды лекции (Наследование): https://github.com/VetrovSV/OOP/blob/master/OOP_1.2.pdf

Задание 3. Калькулятор.

Создать калькулятор с графическим интерфейсом пользователя. Калькулятор должен корректно обрабатывать любые входные данные. Хранить историю вычислений. Помимо арифметических операций и возведения в любою степень калькулятор должен вычислять функции: sin, cos, tan, ln, exp. Использовать шаблон проектирования Представление-Модель.

Вопросы

1. Что такое исключительная ситуация?
2. Как работает механизм обработки исключительных ситуаций?
3. Что такое бизнес-логика?
4. В какой части программы должна быть реализована бизнес-логика?
5. Изобразите диаграмму классов для приложения.
6. Какие возможности среды программирования использовались? Автоматическая генерация методов, конструктора? Рефакторинг? Автоматическое изменение сигнатуры методов? Система управления версиями?

Допускается изменение тем лабораторных работ по предварительному согласованию с преподавателем.

Семестр 4

Задание 1. Простая БД

1. Создать простое приложение с графическим интерфейсом пользователя: простая файловая БД с GUI.
2. Описать диаграмму классов для созданной программы. На диаграмме должны присутствовать только название классов и тех методов, знание которых необходимо для понимания диаграммы.
3. Каждый созданный класс должен иметь минимальную документацию: комментарии в коде о назначении класса, его полях и методах.

Можно использовать объектно-ориентированный язык общего назначения на выбор (C++, C#, Java, Kotlin, иной по договорённости с преподавателем)

Требования:

- Одна таблица с 4+ полями.
- Данные хранятся в файле. Собственный формат БД (без использования SQL, noSQL и проч.).
- Разделение представления и модели (данных и методов работы с ними). Интерфейс и бизнес-логика должны быть описаны в отдельных классах.
- Добавление, проверка, изменение данных
- Поиск, сортировка (как минимум по одному полю).
- Документация (в коде) описывающая формат данных в файле.
- Требования к GUI:
  - вывод данных в таблицу
  - меню приложения
  - панель инструментов
  - шрифт и цветовая палитра отличные от задаваемых по умолчанию.

- иконка приложения
- Всплывающая подсказка или подсказка в строке состояния для элементов интерфейса.
- Информация о разработчике.
- *Дополнительно (не обязательно к реализации):*
  - *горячие клавиши*
  - *цветовое кодирование данных в таблице*
  - *использование элементов интерфейса (флажок, числовое поле ввода и тт.п.) в таблице*
  - *хранение изображений в БД*
  - *использовать как минимум одно модальное окно*
  - *Автоматическое сохранение БД через заданные интервалы времени*
  - *краткая справка*

Вопросы

1. Что такое представление и модель?
2. Как представлена модель в программе? Как происходит проверка данных?
3. Какие исключительные ситуации могут возникнуть во время работы программы?
4. Что такое SOLID? Соблюдаются ли эти принципы в вашей программе?

Ссылки

- Считывание данных из файла CSV и их представление через QstandardItemModel https://evileg.com/ru/post/158/

Задание 2. Простой чат бот.

1. Программа реализующая простой чат бот с графическим интерфейсом пользователя.
2. Описать диаграмму классов для созданной программы. На диаграмме должны присутствовать только название классов и тех методов, знание которых необходимо для понимания диаграммы.
3. Каждый созданный класс должен иметь минимальную документацию: комментарии в коде о назначении класса, его полях и методах.

Можно использовать объектно-ориентированный язык общего назначения на выбор (C++, C#, Java, Kotlin, иной по договорённости с преподавателем)

Требования

- Бизнес-логика в отдельном модуле.
- Ответ на несколько реплик заданного шаблона («Привет, Бот!» и т.п.)
- Ответ на простые команды. (Например: «Который час?», вопросы о статистике по обмену сообщениями и т.п.). Ответ на команды с параметрами: Например: «умножь 12 на 157»
- Бот должен хранить историю сообщений, включая время отправки и автора.
- Записывать историю в файл при завершении программы. Загружать из файла при запуске программы.
- Реализовать один или несколько пунктов
    - Получение актуальной информации из интернета (погода, курсы валют, последние новости и т. п.)
    - Запуск отдельных программ, работа с операционной системой и файлами.
    - Сохранение информации о собеседнике. Собеседник предполагается неизменным
    - Опционально: показ изображений (в том числе загрузка из интернета, например APOD)

- Дополнительно: авторизация в отдельном окне.
- Требования к GUI:
  - шрифт и цветовая палитра (опционально: использование фоновых изображений) отличные от задаваемых по умолчанию.
  - иконка приложения

Вопросы

1. Изобразите диаграмму классов для приложения
2. Что такое бизнес-логика?
3. Опишите шаблон проектирования «Модель – представление».
4. Что такое SOLID? Опишите каждый принцип.
5. Ваша лабораторная соблюдает принципы SOLID?
6. Что такое регулярное выражение?

Ссылки

https://stackoverflow.com/questions/46943134/how-do-i-write-a-qt-http-get-request

Форма промежуточного контроля

Экзамен

Экзамен проводится в третьем и четвертом семестрах. До экзамена допускаются студенты, сдавшие и защитившие все контрольные работы.

Экзаменационный билет состоит из трёх вопросов: двух теоретических, одной задачи. Задачи выбираются подобными тем, что решались в контрольных работах или разбирались на занятиях.

Перечень примерных вопросов для подготовки к экзамену

Семестр 3

1. Исключительные ситуации
2. ОО декомпозиция
3. Отношения между классами. Диаграмма классов
4. Алгоритмы стандартной библиотеки
5. Стандартная библиотека
6. Создание и использование объектов класса. Массивы объектов.
7. Абстрагирование
8. Агрегирование. Композиция
9. Друзья классов
10. Наследование. Виды
11. Приведение типов
12. Виртуальные методы и классы
13. Перегрузка методов, переопределение методов
14. Перегрузка операторов
15. Полиморфизм. Виды. Примеры
16. Динамический полиморфизм
17. Средства преобразования типов. Явные преобразования static_cast, dynamic_cast,reinterpret_cast.
18. Абстрактные классы.
19. Статический полиморфизм
20. Указатели на функции и методы.
21. Лямбда функции.
22. Статические члены класса
24. Шаблоны
25. Создание библиотек
26. Многопоточные приложения
27. SOLID
28. Паттерны проектирования
29. Юнит-тестирование
30. Фреймворк Qt
31. Языки описания интерфейса пользователя. QML

Семестр 4

23. Вопросы из семестра 3

Критерии формирования оценок экзамена

Экзамен проводится в устной форме: обсуждается теоретический материал и приводится решение практических заданий с объяснением.

Оценки *"отлично"* заслуживает студент, обнаруживший всестороннее, систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой.

Оценки *"хорошо"* заслуживает студент обнаруживший полное знание учебно-программного материала, успешно выполняющий предусмотренные в программе задания. Ответ содержит в целом правильное, но не всегда точное и аргументированное изложение материала.

Оценки *"удовлетворительно"* заслуживает студент, обнаруживший знания основного учебно-программного материала в объёме, необходимом для дальнейшей учёбы и предстоящей работы по специальности. Эта оценка выставляется студентам, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя.

Оценка *"неудовлетворительно"* выставляется студенту, обнаружившему пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий.

Наличие сертификата онлайн-курса (соответствующего содержанию дисциплины) может быть учтено на экзамене.

Учебно-методическое и информационное обеспечение дисциплины

Основная литература

1. Объектно-ориентированное программирование в C++: лекции и упражнения [Электронный ресурс] : Учебное пособие для вузов / Ашарина И.В. - М. : Горячая линия - Телеком, 2012. - http://www.studentlibrary.ru/book/ISBN9785991270014.html
2. Объектно Ориентированное Программирование. Хорошая книга для Хороших Людей [Электронный ресурс] / Комлев Н.Ю. - М. : СОЛОН-ПРЕСС, 2015. - http://www.studentlibrary.ru/book/ISBN9785913591388.html

Дополнительная литература

1. Б. Страуструп Язык программирования C++. 1136 с. 2015 г.
2. Г. Буч. Объектно-ориентированный анализ и проектирование с примерами приложений. 720 с. 2010 г.
3. "Программирование на C++ [Электронный ресурс] / Дейл Н., Уимз Ч., Хедингтон М. Пер. с англ. - М. : ДМК Пресс, 2000. - (Серия "Учебник")" - http://www.studentlibrary.ru/book/ISBN5937000080.html

Базы данных, информационно-справочные и поисковые системы

1. ЭБС Юрайт. – Режим доступа: https://www.biblio-online.ru
2. ЭБС "КОНСУЛЬТАНТ СТУДЕНТА". – Режим доступа: http://www.studentlibrary.ru/
3. Библиотека ЗабГУ. – Режим доступа: http://library.zabgu.ru.
4. Система вопросов и ответов о программировании . – Режим доступа: https://stackoverflow.com
5. https://ru.cppreference.com – информация по языку и стандартной библиотеке C++

6. Программирование на языке C++ - онлайн-курс
7. Программирование на языке C++ (продолжение) - онлайн-курс

Ведущий преподаватель:
старший преподаватель кафедры ИВТ и ПМ
Ветров Сергей Владимирович.

Заведующий кафедрой ИВТ и ПМ к. ф.-м. н., доцент О.В. Валова.