

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ:
Директор



А. В. Замятин

« 14 » _____ 20 23 г.

Рабочая программа дисциплины

Объектно-ориентированное программирование

по направлению подготовки

09.03.03 Прикладная информатика

Направленность (профиль) подготовки :

Разработка программного обеспечения в цифровой экономике

Форма обучения

Очная

Квалификация

Бакалавр

Год приема

2023

Код дисциплины в учебном плане: Б1.О.04.05

СОГЛАСОВАНО:

Руководитель ОП

С.П. Сущенко

Председатель УМК

С.П. Сущенко

Томск – 2023

1. Цель и планируемые результаты освоения дисциплины

Целью освоения дисциплины является формирование следующих компетенций:

– ОПК-2 – Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности.

ИОПК-2.1 Обладает необходимыми знаниями в области информационных технологий и программных средств, в том числе понимает принципы их работы.

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИОПК-2.1 Владеет необходимыми знаниями в области информационных технологий и программных средств, в том числе понимает принципы их работы.

2. Задачи освоения дисциплины

– Изучить основные методы и средства объектно-ориентированного программирования на языке C++.

– Научиться применять полученные знания для решения практических задач профессиональной деятельности.

3. Место дисциплины в структуре образовательной программы

Дисциплина относится к обязательной части образовательной программы. Дисциплина входит в модуль «Разработка программного обеспечения».

4. Семестр(ы) освоения и форма(ы) промежуточной аттестации по дисциплине

Третий семестр, зачет с оценкой

5. Входные требования для освоения дисциплины

Для успешного освоения дисциплины требуются результаты обучения по следующим дисциплинам: Основы программирования, Математическая логика и теория алгоритмов.

6. Язык реализации

Русский

7. Объем дисциплины

Общая трудоемкость дисциплины составляет 3 з.е., 108 часов, из которых:

-лекции: 32 ч.

-лабораторные: 32 ч.

Практическая подготовка – 32 ч.

Объем самостоятельной работы студента определен учебным планом.

8. Содержание дисциплины, структурированное по темам

Тема 1. Основные принципы ООП

Понятие и структура класса. Отношения между классами. Основные принципы – инкапсуляция, полиморфизм, наследование.

Тема 2. Конструкторы и деструкторы

Объекты – создание и разрушение. Конструкторы с параметрами и без. Конструкторы копий. Деструкторы.

Тема 3. Перегрузка функций и операторов
 Перегрузка функций, указатель this. Перегрузка бинарных и унарных операторов.
 Дружественные функции и операторы. Ссылки.

Тема 4. Наследование и полиморфизм
 Принципы и модификаторы наследования. Конструкторы и деструкторы производных классов. Множественное наследование. Иерархии классов, ссылки и указатели на производные типы. Виртуальные функции. Абстрактные классы. Статический и динамический полиморфизм.

Тема 5. Исключения, управление памятью, ввод-вывод
 Обработка исключений. Генерация и перехват исключений. Работа с динамической памятью в C++. Система ввода-вывода в C++. Поток ввода-вывода. Перегрузка операторов вставки и извлечения. Текстовые и бинарные потоки, произвольный доступ.

Тема 6. Шаблоны и библиотека STL
 Шаблонные функции. Шаблонные классы. Библиотека STL, основные типы контейнеров.

9. Текущий контроль по дисциплине

Текущий контроль по дисциплине проводится путем контроля посещаемости, выполнения трех лабораторных работ и фиксируется в форме контрольной точки один раз в семестр. Лабораторные работы оцениваются по пятибалльной шкале по следующим параметрам:

- полнота реализации программы,
- ответы на вопросы по переменным, функциям, классам программы
- ответы на вопросы по теории из соответствующего раздела курса
- умение исправлять ошибки и оперативно вносить изменения в программу.

Практическая подготовка оценивается по результатам выполненных лабораторных работ.

10. Порядок проведения и критерии оценивания промежуточной аттестации

Теоретические и практические результаты формируются компетенциями ИОПК-2.1 и результатами обучения:

№	Этапы формирования компетенций (разделы дисциплины)	Код и наименование результатов обучения	Вид оценочного средства (тесты, задания, кейсы, вопросы и др.)
1.	Основные принципы ООП	ОР-2.1.1. Знать основы объектно-ориентированного программирования ОР-2.1.2. Уметь применять полученные знания при разработке программ с использованием объектно-ориентированного подхода	Вопросы по теории раздела 1.
2.	Конструкторы и деструкторы	ОР-2.1.1. Знать основы объектно-ориентированного программирования ОР-2.1.2. Уметь применять полученные знания при разработке программ с использованием объектно-ориентированного подхода	Вопросы по теории раздела 2.

3.	Перегрузка функций и операторов	ОР-2.1.1. Знать основы объектно-ориентированного программирования ОР-2.1.2. Уметь применять полученные знания при разработке программ с использованием объектно-ориентированного подхода	Вопросы по теории раздела 3. Задания и вопросы к лабораторной работе №1
4.	Наследование и полиморфизм	ОР-2.1.1. Знать основы объектно-ориентированного программирования ОР-2.1.2. Уметь применять полученные знания при разработке программ с использованием объектно-ориентированного подхода	Вопросы по теории раздела 4. Задания и вопросы к лабораторной работе №2
5.	Исключения, управление памятью, ввод-вывод	ОР-2.1.1. Знать основы объектно-ориентированного программирования ОР-2.1.2. Уметь применять полученные знания при разработке программ с использованием объектно-ориентированного подхода	Вопросы по теории раздела 5.
6.	Шаблоны и библиотека STL	ОР-2.1.1. Знать основы объектно-ориентированного программирования ОР-2.1.2. Уметь применять полученные знания при разработке программ с использованием объектно-ориентированного подхода	Вопросы по теории раздела 6. Задания и вопросы к лабораторной работе №3

Зачет с оценкой в третьем семестре проводится по результатам сдачи лабораторных заданий. При сдаче каждой лабораторной работы проверяются знания и умения по индикаторам всех компетенций дисциплины.

Примерный перечень теоретических вопросов приведен в разделе 8 настоящего документа.

Примеры заданий для лабораторных работ:

1. Перегрузка конструкторов, функций и операторов.

Цель работы: практическая реализация и использование контейнерных классов с перегруженными методами и операторами.

Описание: реализовать упрощенный аналог класса string (STL) и класса «массив строк», включающего, кроме прочих, методы сортировки массива; проверить работу всех методов и операторов.

2. Наследование и полиморфизм.

Цель работы: практическая реализация простых иерархий классов и исследование работы виртуальных функций.

Описание: реализовать иерархию классов, поставляющих числовые значения с помощью виртуальной функции Get, и иерархию классов обработки полученных значений; проверить работу программы для различных вариантов вызова Get.

3. Шаблоны классов.

Цель работы: реализация и использование шаблонных классов.

Описание: реализовать шаблонные классы «вектор» и «матрица (массив векторов)», включающих несколько методов и перегруженных операторов, и проверить их на объектах разных типов.

11. Учебно-методическое обеспечение

- а) Электронный учебный курс по дисциплине в электронном университете «Moodle»
- б) Оценочные материалы текущего контроля и промежуточной аттестации по дисциплине.
- в) Методические указания по проведению лабораторных работ.

12. Перечень учебной литературы и ресурсов сети Интернет

- а) основная литература:
 - Лафоре Р. Объектно-ориентированное программирование в C++. Санкт-Петербург [и др.]: Питер, 2016.
 - Павловская Т.А. C/C++. Процедурное и объектно-ориентированное программирование. Санкт-Петербург [и др.]: Питер, 2015.
- б) дополнительная литература:
 - Кьюоу Дж. Объектно-ориентированное программирование. СПб [и др.]: Питер: Питер принт, 2005.
 - Иванова Г.С., Ничушкина Т.Н. Объектно-ориентированное программирование. М.: Издательство МГТУ им. Н. Э. Баумана, 2014.
 - Лаптев В.В. C++. Объектно-ориентированное программирование: [учебное пособие для студентов вузов]. Санкт-Петербург [и др.]: Питер, 2008.

13. Перечень информационных технологий

- а) лицензионное и свободно распространяемое программное обеспечение:
Программное обеспечение – средства программирования на C++:
 - Microsoft Visual Studio,
 - справочная система – MSDN.
- б) информационные справочные системы:
 - Электронный каталог Научной библиотеки ТГУ – <http://chamo.lib.tsu.ru/search/query?locale=ru&theme=system>
 - Электронная библиотека (репозиторий) ТГУ – <http://vital.lib.tsu.ru/vital/access/manager/Index>
 - ЭБС Лань – <http://e.lanbook.com/>
 - Электронно-библиотечная система Znanium.com [Электронный ресурс] / Научно-издательский центр Инфра-М. – Электрон. дан. – М., 2016- . URL: <http://znanium.com/>

14. Материально-техническое обеспечение

Аудитории для проведения занятий лекционного типа и аудитории для проведения лабораторных занятий.

Аудитория для проведения лекционных занятий должна быть оснащена мультимедийным оборудованием с доступом в интернет (проектор, экран, монитор, системный блок). Для проведения лабораторных занятий требуется наличие компьютерной техники с установленным соответствующим программным обеспечением. При освоении дисциплины используются компьютерные классы ИПМКН ТГУ с доступом к ресурсам Научной библиотеки ТГУ, в том числе отечественным и зарубежным периодическим изданиям и Интернету.

Виртуальные аудитории для проведения занятий лекционного и лабораторного типа, индивидуальных и групповых консультаций, текущего контроля и промежуточной аттестации в смешанном формате («Moodle»).

15. Информация о разработчиках

Фукс Александр Львович, канд. техн. наук, доцент кафедры теоретических основ информатики ТГУ.