

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕСПЕЦИАЛЬНОГО
ОБРАЗОВАНИЯ РЕСПУБЛИКИ УЗБЕКИСТАН

БУХАРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ



«УТВЕРЖДАЮ»:
проректор по учебной работе
Бухарского государственного
университета
Жураев Р.Г.
2022 года

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»
(блок 2.13)
4-курс

Область образования: 540000 – Математика и статистика

Направление образования: 5130200 – Прикладная математика и
информатика

Общие учебные часы в 7,8-м семестре - 188 часов из них:

Лекционные занятия - 14 часов (в 7-семестре), 32 часа (в 8-семестре)

Практические занятия - 14 часов (в 7-семестре), 32 часа (в 8-семестре)

Часы самостоятельного образования:

28 часов (в 7-семестре), 68 часов (в 8-семестре)

Бухара-2022

Рабочая учебная программа по предмету « Объектно-ориентированное программирование» утверждена учебно-методическом советом Бухарского государственного университета под №_1_ от «_30_»_августа_2022 года

Составитель:

Атаева Г.И. – ст.преподаватель кафедры Информационные системы и цифровые технологии Бухарского государственного университета

Рецензенты:

Муродова Ф.Р. – доцент кафедры Информатики и Информационные технологии БухИТИ

Сайидова Н.С. – доцент кафедры Информационные системы и цифровые технологии БухГУ, кандидат физико-математических наук.

Декан факультета
информационных технологий
БухГУ:
2022 год



Х.И. Эшанкулов

Заведующий кафедрой
“Информационные системы
и цифровые технологии”:
2022 год

Т.Р.Шафиев

1. Актуальность и место дисциплины в системе образования.

Типовая учебная программа по учебной дисциплине «Объектно-ориентированное программирование и проектирование» разработана для студентов высшего образования, обучающихся по специальности 5330200 «Прикладная математика и информатика» в соответствии с требованиями типового учебного плана вышеуказанной специальности.

Актуальность изучения дисциплины обусловлена устойчивой тенденцией расширения применимости объектных технологий всех уровней систем автоматизированной обработки информации. Это обусловлено необходимостью поддержки итерационного процесса проектирования таких систем, их эволюционного развития и настройки на условия применения. Изучение объектно-ориентированных технологий программирования и проектирования необходимо для решения задач комплексирования и интеграции прикладных и системных программных компонент на основе различных стандартов и протоколов, управления конфигурацией, производительностью, безопасностью использования ресурсов, а также грамотного использования функциональных возможностей и сервисов современных операционных систем при разработке прикладного и системного программного обеспечения.

2. Цели и задачи предмета.

Цель дисциплины: углубленное обучение студентов технологическим основам и практическим навыкам проектирования, реализации и сопровождения больших программных систем современных ЭВМ на основе технологии объектно-ориентированного программирования.

Основные задачи изучения дисциплины: формирование представлений об объектной технологии как примере использования системного подхода и реализации результатов системного анализа; приобретение знаний о возможностях, методах, моделях и средствах поддержки современных промышленных информационных технологий; приобретение навыков практической работы со средствами обеспечения жизненного цикла создания и эволюционного развития сложных программных систем.

Базовыми учебными дисциплинами по курсу «Объектно-ориентированное программирование и проектирование» являются «Теория алгоритмов», «Основы программирования», «Системное программное обеспечение».

3. Требования к уровню освоения содержания учебной дисциплины.

В результате изучения дисциплины «Объектно-ориентированное программирование и проектирование» формируются следующие компетенции: *академические*: умение применять базовые научно-теоретические знания для

решения теоретических и практических задач; владение системным и сравнительным анализом; владение исследовательскими навыками; умение работать самостоятельно; способность порождать новые идеи (обладать креативностью); использовать основные законы естественнонаучных дисциплин в профессиональной деятельности;

социально-личностные: способность к межличностным коммуникациям; способность к критике и самокритике; умение работать в команде.

профессиональные: профессионально эксплуатировать современные автоматизированные системы управления; обучение и повышение квалификации персонала; выполнять постановку задач на автоматизацию обработки информации; вести подготовку технических заданий на разработку программного обеспечения; проводить инфологическое и даталогическое проектирование баз данных; проводить объектный анализ и проектирование систем обработки информации; конструировать шаблоны типовых решений и компонент систем обработки информации.

В результате изучения учебной дисциплины обучающийся должен:

знать: принципы объектно-ориентированного программирования и проектирования; способы реализации отношений между классами; использование свойств полиморфизма, наследования и инкапсуляции; использование абстрактных классов, интерфейсов и шаблонов программирования; принципы представления и структуризации эффективных проектов систем шаблонами проектирования; *уметь*: создавать программы на основе технологий использования классов с использованием современных систем объектно-ориентированного проектирования; переходить из одной объектно-ориентированной платформы на другую;

владеть: методами объектно-ориентированного моделирования, анализа и проектирования; навыками практической работы в современных системах объектно-ориентированного программирования.

Распределение часов по семестрам

Семестр	Лекции	Практические занятия	Самообразование	Всего
7- семестр	14 ч.	14 ч.	28 ч.	56 ч.
8- семестр	32 ч.	32 ч.	68 ч.	132 ч.
	46 ч.	46 ч.	96 ч.	188 ч.

2. Лекционные занятия

Таблица №1.

№	Темы лекций	Количество часов
<i>7-семестр</i>		
1.	Рекурсивные функции C++.	2
2.	Перегрузка функций. Функции шаблоны C++.	2
3.	Файловые потоки C++	2
4.	Структуры (C++).	2
5.	Сортировки. Метод «пузырька», сортировка вставками, сортировка посредством выбора (C++).	2
6.	Использование STL в C++. Класс-контейнер вектор (C++). Работа с векторами. Итераторы. Алгоритмы STL.	2
7.	Исключения. Механизмы обработки исключений. Применение исключений на практике. (C++)	2
Всего за 7-ой семестр		14ч.
<i>8-семестр</i>		
1.	Классы и объекты. Основные понятия. Конструкторы. Деструкторы. (C++)	2
2.	Идентификация типов во время выполнения. Оператор <code>dynamic_cast</code> . Оператор <code>typeid</code> (C++).	2
3.	Иерархия классов исключений (C++).	2
4.	Статические члены класса. Перегрузка операций. (C++)	2
5.	Наследование. Основные понятия. Наследование конструкторов. (C++).	2
6.	Виртуальные функции (C++).	2
7.	Абстрактные классы (C++).	2
8.	Объектно-ориентированная реализация списков. Основные понятия. Стек. (C++)	2
9.	Применение исключений и шаблонов. Очередь (C++).	2
10.	Однонаправленный список общего вида (C++).	2

11.	Двунаправленный список (C++).	2
12.	Реализация списков с помощью STL C++. Классы-контейнеры <code>stack</code> , <code>queue</code> , <code>list</code> .	2
13.	Паттерны проектирования. Порождающие паттерны проектирования (C++)	2
14.	Паттерны проектирования. Структурные паттерны проектирования (C++)	2
15.	Паттерны проектирования. Поведенческие паттерны проектирования (C++)	2
16.	Паттерны проектирования (C++)	2
Всего часов за 8-ой семестр		32

Лекционные занятия проводятся в аудитории, оснащенной мультимедийными средствами.

3. Темы практических занятий.

Таблица №2

№	Практические занятия	Количество часов
<i>7-семестр</i>		
1.	Рекурсивные функции C++. Решение практических задач.	2
2.	Перегрузка функций. Функции шаблоны C++. Решение практических задач.	2
3.	Файловые потоки C++. Решение практических задач.	2
4.	Структуры (C++). Решение практических задач.	2
5.	Сортировки. Метод «пузырька», сортировка вставками, сортировка посредством выбора (C++). Решение практических задач.	2
6.	Использование STL в C++. Класс-контейнер вектор (C++). Работа с векторами. Итераторы. Алгоритмы STL. Решение практических задач.	2
7.	Исключения. Механизмы обработки исключений. Применение исключений на практике. (C++). Решение практических задач.	2

Всего за 7-ой семестр		14
8-семестр		
1.	Классы и объекты. Конструкторы. Деструкторы. (C++). Решение практических задач.	2
2.	Идентификация типов во время выполнения. Оператор <code>dynamic_cast</code> . Оператор <code>typeid</code> (C++). Решение практических задач.	2
3.	Иерархия классов исключений (C++). Решение практических задач.	2
4.	Статические члены класса. Перегрузка операций. (C++). Решение практических задач.	2
5.	Наследование. Основные понятия. Наследование конструкторов. (C++). Решение практических задач.	2
6.	Виртуальные функции (C++). Решение практических задач.	2
7.	Абстрактные классы (C++). Решение практических задач.	2
8.	Объектно-ориентированная реализация списков. Основные понятия. Стек. (C++). Решение практических задач.	2
9.	Применение исключений и шаблонов. Очередь (C++). Решение практических задач.	2
10.	Однонаправленный список общего вида (C++). Решение практических задач.	2
11.	Двунаправленный список (C++). Решение практических задач.	2
12.	Реализация списков с помощью STL C++. Классы-контейнеры <code>stack</code> , <code>queue</code> , <code>list</code> . Решение практических задач.	2
13.	Паттерны проектирования. Порождающие паттерны проектирования (C++). Решение практических задач.	2
14.	Паттерны проектирования. Структурные паттерны проектирования (C++). Решение практических задач.	2
15.	Паттерны проектирования. Поведенческие паттерны проектирования (C++). Решение практических задач.	2
16.	Паттерны проектирования (C++). Решение практических задач.	2
Всего часов за 8-ой семестр		32

Практические занятия проводятся в аудитории оснащенной компьютерами, для каждой академической группы отдельно. Занятия проводятся с использованием интерактивных педагогических технологий.

4. Самостоятельная работа

Таблица №3

№	Самостоятельная работа	Количество часов
7-семестр		
1.	Ссылки. Параметры-ссылки. Возвращаемое значение-ссылка	2
2.	Подставляемые (inline) функции. Метки <code>static</code> , <code>static</code> и <code>inline</code> . Анонимные объединения. Перегрузка функций.	2
3.	Операция разрешения видимости. Оператора распределения памяти. Перегрузка операций <code>new</code> и <code>delete</code> .	2
4.	Классы. Элементы-функции. Дополнительные сведения об элементах класса.	2
5.	Инкапсуляция и целостность типа. Конструктор класса. Деструктор.	2
6.	Список инициализации. Область видимости класса. Конструктор копий. Операция присваивания.	2
7.	Константная ссылка и качестве аргумента. Конструктор копий или операция присваивания? Проверка на само-присвоение. Запрещение копирования и присваивания. Операции приведения.	2
8.	Перегрузка операций. Правила перегрузки. Перегрузка C++. Операция индексирования. Операция функция.	2
9.	Друзья. Дружественные классы. Дружественные функции. Правила относительно друзей. Перегрузка операций: функция-элемент vs функция-друга.	2
10.	Статические элементы класса. Статические функции-элементы. Константные объекты и константные функции-элементы. Операции класса <code>new</code> и <code>delete</code> .	2
11.	Наследование. Частное наследование. Множественное наследование. Неоднозначность при множественном наследовании.	2
12.	Виртуальный базовый класс. Конструкторы, деструкторы и наследование.	2
13.	Полиморфизм. Виртуальные функции и позднее связывание. Виртуальный деструктор.	2
14.	Файлы и модули. Чистые виртуальные функции и абстрактные классы. Шаблоны. Шаблоны функций. Перегрузка шаблонов функций. Разрешение ссылки на функцию.	2
Всего за 7-ой семестр		28 ч.

8-семестр		
1.	Шаблоны классов. Структуры и объединения. Указатели на функциональные элементы. Выключение виртуального механизма.	2
2.	Потоки. Предопределенные потоки. Операция помещения. Сцепление операций помещения. Операция извлечения.	2
3.	Форматирование. Форматирующие функции-элементы. Флаги форматирования. Манипуляторы	2
4.	Ошибки. Способы управления состоянием потока.	2
5.	Перегрузка операций помещения и извлечения для типов пользователя. Функции ввода (поток istream). Функции вывода (поток ostream).	2
6.	Переадресация ввода и вывода. Файловый ввод/вывод. Открытие файла. Режимы доступа к файлу. Закрытие файла.	2
7.	Не форматизируемый ввод/вывод данных. Форматирование и памяти. Иерархия классов потоков.	2
8.	Управление исключениями. Спецификация исключений функций.	2
9.	Использование шаблонов функций. Создание простого шаблона функции. Использование шаблонов функций.	2
10.	Шаблоны, использующие несколько типов. Шаблоны и несколько типов.	2
11.	Создание шаблонов классов. Синтаксис описания шаблона.	2
12.	Использование шаблонов классов. Явная специализация шаблонов. Достоинства и недостатки шаблонов.	2
13.	Обработка исключений. Перехват исключений. Объекты-исключения.	2
14.	Раскрутка стека. Повторное возбуждение исключений. Обработка исключений в C++.	2
15.	Обработка исключений в C++. Исключительные ситуации в конструкторах и деструкторах.	2
16.	Идентификация типов во время выполнения. Иерархия классов исключений.	2
17.	Паттерны проектирования.	2
18.	Использование STL в C++.	2
19.	Применение исключений и шаблонов. Очередь (C++).	2
20.	Объектно-ориентированная реализация списков. Основные понятия. Стек. (C++).	2
21.	Идентификация типов во время выполнения. Оператор dynamic_cast. Оператор typeid (C++).	2
22.	Конструкторы. Деструкторы. (C++).	2

23.	Порождающие паттерны проектирования (C++).	2
24.	Структурные паттерны проектирования (C++).	2
25.	Поведенческие паттерны проектирования (C++).	2
26.	Работа в среде Microsoft Visual Studio.	2
27.	Математические функции в C++.	2
28.	Типы данных в C++: значимые и ссылочные типы.	2
29.	Разработать программу вычисления значений функций одной переменной на заданном отрезке, с заданным шагом.	2
30.	Создание динамического массива заданного размера, выделение из массива указанных в варианте диапазонов.	2
31.	Разработка на C++ приложения для статистической обработки массива случайных данных. Генератор случайных чисел.	2
32.	Разработать шаблонный класс для представления динамических одномерных массивов. Класс обеспечивает хранение данных любого типа, для него предусмотрены конструктор по умолчанию, конструктор копирования и операция присваивания.	2
33.	Разработать программу калькулятор на C++	2
34.	Реализация метода наименьших квадратов для решения уравнения линейной регрессии	2
Всего за 8-ой семестр		68 ч.

Самостоятельная работа выполняется в виде конспекта или в виде презентации.

После прохождения курса дисциплины проводится промежуточный контроль, максимальная оценка – 50 баллов = 40б письменная работа + 10б за самостоятельную работу.

Итоговый контроль проводится в виде письменной работы (3 вопроса по теории и 2 вопроса практических), максимальная оценка 50 баллов.

Студент должен ответить на пять вопросов по предмету, представленных в билете.

5. Критерии оценивания знаний студентов по дисциплине

Способы оценивания	Экспресс-тесты, письменные работы, устный опрос, презентации
Критерии оценивания	<p>Ответ или работа содержит полный ответ – студент глубоко и прочно усвоил программный материал, последовательно, грамотно и логически стройно его изложил; и ответе тесно связана теория с практикой; студент даёт определения, поясняет их, может ответить на выделенное задание; свободно справляется с задачами, использует графические организаторы, показывает знакомство с информационными технологиями. Самостоятельно справляется с практическим заданием - оценка 5 (отлично);</p> <p>Студент знает программный материал, определения, разбирается в информационных терминах, правильно применяет теоретические</p>

<p>сведения, положения при решении практических вопросов и задач. Но при ответе путается, допускает ошибки при решении практических заданий, часто не собран, в целом, знания не систематизированы -оценка 4 (хорошо);</p> <p>Студент знает общие положения основного материала, но допускает неточности, часто путается, не всегда может дать точное определение того или иного термина, нарушает последовательность в изложении программного материала, испытывает трудности при выполнении тестовых и практических заданий, не увязывает теорию с практикой -оценка 3 (удовлетворительно);</p> <p>Студент не знает значительной части программного материала, допускает существенные ошибки, не справляется с практическими заданиями, не использует графические организаторы при ответе. Часто пропускает занятия без уважительной причины -оценка 2 (неудовлетворительно).</p>
--

6. Форма оценивания студентов по предмету.

№	Вид оценивания/ оценка, форма	ПО-1	ИО*	Итоговая оценка
1.	Максимальная оценка	5	5 (100 баллов)	Итоговая оценка студентов ставится на основе итогового контроля.
2.	Форма:	устно	Письменная работа	

* ИО проводится в соответствии с учебным графиком, составленным деканом факультета или учебно-методической комиссией университета.

Порядок оценивания:

Промежуточный контроль осуществляется по пятибалльной системе, соответственно проходной оценкой считаются – 2,3,4,5. Студентам задаётся по 5 вопросов по пройденным темам.

Суммарная оценка: Каждое из заданий оценивается 1 баллом, суммарная оценка является заключительной.

Итоговая работа проводится в письменном виде. Студентам раздаются билеты с пятью вопросами за каждый из которых студент может заработать 1 балл. Суммарная оценка является *итоговой*.

Затем производится переход из 5-ти балльной системы оценивания, в 100 балльную шкалу, в которой: Проходной балл – 60 баллов. Оценка 3 – 60-69 баллов. Оценка 4 – 70-89 балла. Оценка 5 – 90-100 баллов.

Список основной литературы

1. Огнева М.В., Кудрина Е.В. Программирование на языке C++: практический курс: учебное пособие для ВУЗов. – Москва, 2022 – 335с. / Электронный ресурс.
2. Радченко Г.И., Захаров Е.А. Объектно-ориентированное программирование / Г.И. Радченко, Е.А. Захаров. – Челябинск: Издательский центр ЮУрГУ, 2013. – 167 с./ Электронный ресурс.
3. C++ Core Guidelines Explained: Best Practices for Modern C++ by Rainer Grimm./Электронный ресурс.

Дополнительная литература

1. C++ Notes for Professionals. (Заметки по C++ для профессионалов). / Электронный ресурс.
2. Прата Стивен. Язык программирования C++. Лекции и упражнения. 6-издание. 2012. (Электронный ресурс).
3. Культин Н.В. Microsoft® Visual C++ в задачах и примерах. — 2-е изд., исправл. — СПб.: БХВ-Петербург, 2014. — 272 с. (Электронный ресурс).
4. Мирзиёев Ш.М. Стратегия Нового Узбекистана. – Ташкент -2021-100с.

Электронные ресурсы

1. <https://ru.code-basics.com/languages/cpp/lessons/first-program>
2. <https://www.codecademy.com/>
3. <https://www.coursera.org/courses?query=c%2B%2B>
4. <http://nullpro.info>
5. <https://metamit.com>