

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ

**КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМ. И. РАЗЗАКОВА**

Факультет информационных технологий

Кафедра «Прикладная математика и информатика»

«УТВЕРЖДАЮ»

Декан ФИТ
д.ф.-м.н., проф. Кабаева Г.Дж.

(подпись)

« ____ » _____ 2020г.

**УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС
ПО ДИСЦИПЛИНЕ**

Б.3.В.1. Объектно- ориентированное программирование
(код, название)

| | |
|-------------------------------|--|
| <u>Направление:</u> | 680200 Биотехнические системы и технологии |
| <u>Профиль:</u> | Медицинская информатика |
| <u>Квалификация:</u> | бакалавр |
| <u>Форма обучения:</u> | очная |

Бишкек 2020 г.

Лист согласования

Учебно-методический комплекс по дисциплине «Объектно- ориентированное программирование» разработан в соответствии с требованиями ГОС ВПО по подготовки бакалавров и предназначен для студентов, обучающихся по направлению 680200 Биотехнические системы и технологии профилю/программе медицинская информатика.

Автор: ст. преп. Кыштобаева Г.К.

| Процесс рассмотрения и утверждения УМКД | № протокола | Подписи (печать) |
|---|--|---|
| Учебно-методический комплекс дисциплины рассмотрен на заседании кафедры _____ <hr/> (наименование учебного подразделения) | протокол № _____ от « _____ » _____ 20__ г. | Зав. профилирующей кафедры: <hr/> (подпись) <u>Ф.И.О.</u> _____ |
| *Учебно-методический комплекс дисциплины рассмотрен на заседании кафедры _____ — <hr/> (наименование учебного подразделения) | протокол № _____ от « _____ » _____ 20__ г. | Зав. непрофилирующей кафедры: <hr/> <u>Ф.И.О.</u> _____ |
| Учебно-методический комплекс дисциплины одобрен руководителем ООП _____ по направлению _____ <hr/> (наименование учебного подразделения) | Дата: | Руководитель ООП: <hr/> (подпись) <u>Ф.И.О.</u> _____ |
| Учебно-методический комплекс дисциплины согласован на заседании Учебно-методической комиссии факультета/института _____ — (наименование учебного подразделения) | протокол № _____ от « _____ » _____ 20__ г., | Председатель УМК: <hr/> (подпись) <u>Ф.И.О.</u> _____ |
| **Учебно-методический комплекс дисциплины согласован (или обсуждался/рецензирован) _____ (указать наименование предприятия/ учреждения/организации) | Дата: согласования / обсуждения/ рецензия | (должность) _____ <hr/> (подпись) <u>Ф.И.О.</u> _____ — |

СОДЕРЖАНИЕ УМК

| | стр |
|---|-----|
| 1. Титульный лист..... | 1 |
| 2. Содержание учебно-методического комплекса..... | 3 |

Часть 1

| | |
|---|----|
| 3. Рабочая программа..... | 4 |
| 3.1.Содержание дисциплины по ГОС | 5 |
| 3.2.Состав разделов Рабочей программы (цели и задачи изучения дисциплины, ее значение в учебном процессе)..... | 5 |
| 3.3.Межпредметные связи. Перечень дисциплин и разделов, усвоение которых необходимо при изучении данной дисциплины..... | 5 |
| 3.4.Структура дисциплины с разбивкой по модулям и видам учебных занятий..... | 6 |
| 3.5. Содержание программы курса по темам..... | 6 |
| 3.6. Задания для самостоятельной работы студентов | 7 |
| 3.7. Распределение баллов по модулям и видам учебных занятий..... | 8 |
| 3.8. Перечень контрольных вопросов теоретического курса..... | 8 |
| 3.10.Список литературы..... | 9 |
| 4. Силлабус..... | 10 |
| 5. Глоссарий..... | 15 |

Часть 2

| | |
|---|--|
| 5.1.Учебно – методические материалы | |
| 5.2.Формы текущего и итогового контроля..... | |
| 5.3.Учебно–методическая литература по дисциплине, разработанная преподавателями кафедр..... | |

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ
КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМ. И. РАЗЗАКОВА

Факультет информационных технологий

Кафедра «Прикладная математика и информатика»

«УТВЕРЖДАЮ»

Декан ФИТ
д.ф.-м.н., проф. Кабаева Г.Дж.

(подпись)

« _____ » _____ 2020г.

РАБОЧАЯ ПРОГРАММА ПО ДИСЦИПЛИНЕ
Б.3.В.1. Объектно- ориентированное программирование
(код, название)

| | | |
|-------------------------------|--|---------|
| <u>Направление:</u> | 680200 Биотехнические системы и технологии | |
| <u>Профиль:</u> | Медицинская информатика | |
| <u>Квалификация:</u> | бакалавр | |
| <u>Форма обучения:</u> | очная | |
| <u>Семестр</u> | 6 | |
| <u>Всего кредитов</u> | 4 кредита | 120 час |
| <u>Лекции</u> | 2 кредита | 32 час |
| <u>Практические</u> | 2 кредита | 32 час |
| <u>СРС</u> | - | 56 час |

Бишкек 2020 г.

Лист согласования

Рабочая программа по дисциплине «Объектно- ориентированное программирование» разработана в соответствии с требованиями ГОС ВПО по подготовки бакалавров и предназначена для студентов, обучающихся по направлению 680200 Биотехнические системы и технологии профилю/программе медицинская информатика.

Автор: ст.преп. Кыштобаева Г.К.

| Процесс рассмотрения и утверждения РПД | № протокола | Подписи (печать) |
|---|---|---|
| Рабочая программа дисциплины рассмотрена на заседании кафедры _____ _____ (наименование учебного подразделения) | протокол № _____ от « _____ » _____ 20__ г. | Зав. профилирующей кафедры: _____ _____ (подпись) Ф.И.О. _____ |
| *Рабочая программа дисциплины рассмотрена/согласована на заседании кафедры _____ _____ (наименование учебного подразделения) | протокол № _____ от « _____ » _____ 20__ г. | Зав. не/профилирующей кафедры: _____ _____ Ф.И.О. _____ |
| Рабочая программа дисциплины одобрена руководителем ООП по направлению _____ _____ (наименование учебного подразделения) | Дата: _____ | Руководитель ООП: _____ _____ (подпись) Ф.И.О. _____ |
| Рабочая программа дисциплины согласована на заседании Учебно-методической комиссии факультета/института _____ _____ (наименование учебного подразделения) | протокол № _____ от « _____ » _____ 20__ г., | Председатель УМК: _____ _____ (подпись) Ф.И.О. _____ |
| **Рабочая программа дисциплины согласована _____ (или обсуждалась/рецензирована) _____ (указать наименование предприятия/учреждения/организации) | Дата: _____ согласования/ обсуждения/ рецензия | (должность) _____ _____ (подпись) Ф.И.О. _____ |

Лист изменений и дополнений в РПД

| № п/п | Номер и название раздела РПД | Описание изменений/дополнений в РПД | Дата изменений | № протокола заседания кафедры | Подписи (печать) преподавателя, зав. кафедрой |
|----------|------------------------------------|--|-------------------|--|--|
| | | | | | _____ _____ |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

ВЫПИСКА ИЗ ГОСУДАРСТВЕННОГО ОБРАЗОВАТЕЛЬНОГО СТАНДАРТА

| Код дисциплин. | Наименование дисциплин и их основные разделы | Всего часов |
|----------------|--|-------------|
| ОП 3.5 | Объектно –ориентированное программирование | 120 |
| | Основы объектно – ориентированного программирования; структурные конструкции в различных языках; основные логические функции и их запись в коде; методы структурного нисходящего программирования; | |

1. Цели и задачи дисциплины

1.1. Цель дисциплины

Целями освоения дисциплины «Объектно-ориентированное программирование» являются овладение основами объектного подхода, объектно-ориентированного и обобщённого программирования на языке C++, приобретение навыков разработки программного кода с использованием современных инструментальных средств для MS Windows.

1.2. Основные задачи дисциплины

Основные задачи дисциплины «Объектно-ориентированное программирование» - изучение и усвоение основных понятий, методов и приемов объектно-ориентированного и визуального программирования, развитие навыков алгоритмического мышления, а также общие элементы языков и общую структуру сред программирования.

1.3 Пререквизитами для изучения данной дисциплины является

- «Информатики»;
- «Математический анализ»;
- «Программирование C++»

1.4 Постреквизитами дисциплины

Данная дисциплина является фундаментом изучения дисциплин:

- «Компьютерная безопасность и защита информации»

2. Структура дисциплины

2.1.Содержание лекционных занятий

2.1.1. Основные принципы объектно– ориентированного программирования (ООП) 8 ч.

Предмет курса. Объектно-ориентированный подход. Основные принципы объектно-ориентированного программирования. Объект, сообщение, класс, экземпляр объекта и метод. Абстракция, инкапсуляция, наследование и полиморфизм. Дополнения к C++. Прототипы функций. Операции расширения области видимости. Перегрузка функций.

2.1.2. Классы. Протокол класса

8 ч.

Объектно-ориентированное проектирование. Классы. Описания протокола класса. Классы и методы в языке C++. Ключевое слово this. Функции типа inline. Управление доступом к элементам классов. Функции-члены класса. Конструкторы и деструкторы. Дружественные функции и классы.

2.1.3. Перегрузка операций

8 ч.

Массивы объектов. Конструктор копирования. Дружественные функции и перегрузка операций. Перегрузка унарных и бинарных операторов. Оператор преобразования типов. Оператор присваивания и индексирования. Перегруженные операции new, delete и ->.

2.1.4. Производные классы. Одиночное наследование.

4 ч.

Производные классы: одиночное наследование. Подкласс, подтип и принцип подстановки. Формы наследования. Наследование в языке C++. Повторное использование кода: наследование и композиция. Открытые и закрытые производные классы. Правила доступа для классов и объектов. Правила доступа для друзей классов и производных классов.

2.1.5. Полиморфизм и виртуальные функции.

4 ч.

Виртуальные функции и полиморфизм. Позднее связывание. Замещение и уточнение. Присваивание в C++

2.2. Перечень лабораторных занятий

| Наименование занятий, содержание | Количество часов | Примечание |
|--|------------------|------------|
| Семестр 5 | | |
| Модуль 1 | 16 | |
| 1. Создание абстрактных типов данных с использованием структуры класс языка c++. | 4 часа | |
| 2. Конструкторы и деструкторы. Создание динамических объектов. Конструктор копирования. | 4 часа | |
| 3. Перегрузка операции C++. | 4 часа | |
| 4. Дружественные функции. Различия между дружественными функциями и функциями-членами. | 4 часа | |
| Модуль 2 | 16 | |
| 5. Правила доступа для друзей классов и производных классов к закрытым, защищенным и открытым частям описаний классов. Дружественные и производные классы. | 4 часа | |
| 6. Перегрузка бинарных и унарных операций. Перегрузка операций потокового ввода / вывода. | 4 часа | |
| 7. Использование шаблонов. Параметрический полиморфизм. | 4 часа | |
| 8. Наследование. Одиночное наследование. Повторное использование кода. Наследование и композиция. Виртуальные функции. Множественное наследование. | 4 часа | |

2.3 Перечень разделов и тем для самостоятельной работы

| Перечень разделов и тем | Количество часов |
|---|------------------|
| | 56 ч. |
| Инкапсуляция, полиморфизм, наследование – основные принципы ООП. | |
| Класс как абстрактный тип данных(АТД).Объявление абстрактного типа данных. Использование АТД в программе | |
| Перегрузка функций. Операции расширения области видимости. Объявления в операторах. Перегруженные функции. Значения формальных параметров по умолчанию. Инициализация ссылочной переменной. Параметры-ссылки и параметры указатели. const, inline, new, delete. Связывание с сохранением типов. указатель на void | |
| Потоки. Файлы и потоки. Вывод и ввод встроенных типов. Вывод и ввод определяемых пользователем типов. | |
| Перегрузка операций. Оператор присваивания. Присваивания. Инициализация. Перегружаемые операторы new и delete, ->. | |
| Дружественные функции и дружественные классы. Композиция. Дружественные классы. Написание программы с применением контернейных классов, дружественных классов. | |
| Правила доступа для друзей классов и производных классов к закрытым, защищенным и открытым частям описаний классов. Дружественные и производные классы. Процедуры для управления файлами и получения информации о файлах, хранящейся в операционной системе. | |

2.4. РАСПРЕДЕЛЕНИЕ БАЛЛОВ ПО МОДУЛЯМ И ВИДАМ УЧЕБНЫХ ЗАНЯТИЙ

| Модуль | Лк | | Лб | | СРС | | Всего | |
|-----------------|-----|-----|-----|-----|-----|-----|-------|-----|
| | min | max | min | max | min | max | min | max |
| I | 8 | 14 | 3 | 4 | 7 | 12 | 18 | 30 |
| II | 8 | 14 | 3 | 4 | 7 | 12 | 18 | 30 |
| Итоговый модуль | 10 | 16 | 5 | 8 | 10 | 16 | 25 | 40 |
| Всего | 26 | 44 | 11 | 16 | 24 | 40 | 61 | 100 |

Оценки в кредитной технологии обучения

| Оценка по буквенной системе (по 10-балльный) | % ное содержание (баллы) | Цифровой эквивалент баллов | Оценка балльной системе |
|--|--------------------------|----------------------------|-------------------------|
| A | 94 - 100 | 4,0 | отлично |
| A ⁻ | 88 – 93 | 3,67 | |
| B ⁺ | 85 – 87 | 3,33 | |
| B | 80 – 84 | 3,0 | хорошо |
| B ⁻ | 74 – 79 | 2,67 | |
| C ⁺ | 70 – 73 | 2,33 | |
| C | 65 – 69 | 2,0 | удовлетворительно |
| C ⁻ | 61 – 64 | 1,67 | |
| F | 0 – 60 | 0 | |
| | | | неудовлетворительно |

3. Перечень контрольных вопросов теоретического курса

1. Как определяются в языке C++ собственные функции пользователя?
2. Что такое прототип и что такое определение функции?
3. Как определять функции возвращающие значение?
4. Как определять функции, не возвращающие значение?
5. Как вызываются в программе функции возвращающие значение и функции, не возвращающие значение?
6. Каким образом можно использовать элемент массива в качестве параметра функции?
7. Расскажите о средствах отладки функций – заглушках и драйверах.
8. Область видимости переменных и констант.
9. Опишите, как использовать в качестве параметра функции массив целиком.
10. Что такое указатель? Какие операции возможны над указателями?
11. Поясните связь между массивами и указателями.
12. Как создается и уничтожается динамический одномерный массив?
13. Назначение и использование функций new и delete.
14. Создание и уничтожение двумерного динамического массива.
15. Что такое структура?
16. Что такое класс?
17. Что такое абстрактный тип данных?
18. Что такое перечисление?
19. Что такое объединение?
20. Как объявить класс с открытыми функциями и открытыми данными?
21. Как определяется функция доступа к закрытым переменным членам класса?
22. Что такое объект, как он объявляется и используется.
23. Как использовать структуру в качестве аргумента функции?
24. Каким образом осуществляется доступ к элементам данных и методам класса или объекта?
25. Что такое массив объектов?
26. Перечисления и объединения для чего они предназначены?
27. Каковы различия модульного и объектно- ориентированного программирования?
28. Назначение конструктора класса. Как вызывается конструктор?
29. Зачем нужен деструктор?
30. Как создается конструктор по умолчанию и конструктор с параметрами?
31. Что такое перегрузка функции?
32. Как создается перегрузка бинарных операций?
33. Как создается перегрузка унарных операций?
34. Как создается перегрузка операций потокового ввода /вывода?
35. Каковы правила перегрузки операций?
36. Какие исключения бывают при перегрузке операций?
37. Как создается конструктор копирования и зачем он используется?
38. Назовите концепции ООП?
39. Что такое полиморфизм?
40. Что такое инкапсуляция?
41. Что такое наследование?
42. Как создается одиночное наследование?
43. Что такое множественное наследование?
44. Как можно управлять доступом производных классов?
45. Как взаимодействуют конструкторы базовых и производных классов?
46. Что такое виртуальная функция?

3. Учебно - методическая поддержка курса

1. Бадд Т. Объектно-ориентированное программирование в действии: Пер. с англ. СПб.: Питер, 1997. 464 с,
2. Уолтер Сэвитч С++ в примерах. Издательство Эком, г. Москва, 1997 г.
3. Б. Страуструп. Язык программирования С++. Издательство Бином. Москва. (любое издание).
4. В.В. Подбельский. Язык С++. Издательство Финансы и статистика. г. Москва. 1999 г.
5. Д. Нейл, Ч. Уимз, М. Хедингтон. Учебник. Программирование на С++. Москва. 2000 г.
6. Ю. Тихомиров. Visual С++6. Новые возможности для программистов. Издательство ВHV. г. Санкт – Петербург. 1999 г.
7. Программирование на С++. Учебное пособие. Под редакцией А.Д. Хомоненко. Издательство КОРОНА Принт. г. Санкт – Петербург. 1999 г.
8. Т.А. Павловская. С/С++ программирование на языке высокого уровня. Издательство Питер. г. Санкт – Петербург. 2001 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗКОЙ РЕСПУБЛИКИ

**КЫРГЫЗСКОЙ ГОСУДАРСТВЕННОЙ ТЕХНИЧЕСКОЙ
УНИВЕРСИТЕТ им. И. РАЗЗАКОВА**

«Утвержден»
На заседании Методического Совета
КГТУ им.И.Раззакова

Председатель _____

«__» _____ 2020

Силлабус

**по дисциплине «Объектно-ориентированное программирование» для студентов
направления: 680200 «Биотехнические системы и технологии»**

Дневной формы обучения

| | |
|-------------------------------|-----------------------|
| Факультет | ФИТ |
| Кафедра | Прикладная математика |
| Курс | 3 |
| Семестр | 6 |
| Зачет (семестр) | |
| Экзамен (семестр) | 6 |
| Кол. кредитов | 4 |
| Всего часов по учебному плану | 120 |
| из них: | |
| - лекции | 32 |
| - лабораторных | 32 |
| - самостоятельная работа | 56 |

Силлабус разработан для направления: 680200 «Биотехнические системы и технологии»

Силлабус разработан ст. преподавателем Кыштобаевой Г.К.

2. Структура дисциплины

2.1. Содержание лекционных занятий

2.1.1. Основные принципы объектно– ориентированного программирования (ООП) 8 ч.

Предмет курса. Объектно-ориентированный подход. Основные принципы объектно-ориентированного программирования. Объект, сообщение, класс, экземпляр объекта и метод. Абстракция, инкапсуляция, наследование и полиморфизм. Дополнения к С++. Прототипы функций. Операции расширения области видимости. Перегрузка функций.

2.1.2. Классы. Протокол класса 8 ч.

Объектно-ориентированное проектирование. Классы. Описания протокола класса. Классы и методы в языке С++. Ключевое слово this. Функции типа inline. Управление доступом к элементам классов. Функции-члены класса. Конструкторы и деструкторы. Дружественные функции и классы.

2.1.3. Перегрузка операций 8 ч.

Массивы объектов. Конструктор копирования. Дружественные функции и перегрузка операций. Перегрузка унарных и бинарных операторов. Оператор преобразования типов. Оператор присваивания и индексирования. Перегруженные операции new, delete и ->.

2.1.4. Производные классы. Одиночное наследование. 4 ч.

Производные классы: одиночное наследование. Подкласс, подтип и принцип подстановки. Формы наследования. Наследование в языке С++. Повторное использование кода: наследование и композиция. Открытые и закрытые производные классы. Правила доступа для классов и объектов. Правила доступа для друзей классов и производных классов.

2.1.5. Полиморфизм и виртуальные функции. 4 ч.

Виртуальные функции и полиморфизм. Позднее связывание. Замещение и уточнение. Присваивание в С++

2.2. Перечень лабораторных занятий

| Наименование занятий, содержание | Количество часов | Примечание |
|--|------------------|------------|
| Семестр 5 | | |
| Модуль 1 | 16 | |
| 1. Создание абстрактных типов данных с использованием структуры класс языка с++. | 4 часа | |
| 2. Конструкторы и деструкторы. Создание динамических объектов. Конструктор копирования. | 4 часа | |
| 3. Перегрузка операции С++. | 4 часа | |
| 4. Дружественные функции. Различия между дружественными функциями и функциями-членами. | 4 часа | |
| Модуль 2 | 16 | |
| 5. Правила доступа для друзей классов и производных классов к закрытым, защищенным и открытым частям | 4 часа | |

| | | |
|--|--------|--|
| описаний классов. Дружественные и производные классы. | | |
| 6. Перегрузка бинарных и унарных операций. Перегрузка операций потокового ввода / вывода. | 4 часа | |
| 7. Использование шаблонов. Параметрический полиморфизм. | 4 часа | |
| 8. Наследование. Одиночное наследование. Повторное использование кода. Наследование и композиция. Виртуальные функции. Множественное наследование. | 4 часа | |

2.3 Перечень разделов и тем для самостоятельной работы

| Перечень разделов и тем | Количество часов |
|---|------------------|
| | 56 ч. |
| Инкапсуляция, полиморфизм, наследование – основные принципы ООП. | |
| Класс как абстрактный тип данных(АТД).Объявление абстрактного типа данных. Использование АТД в программе | |
| Перегрузка функций. Операции расширения области видимости. Объявления в операторах. Перегруженные функции. Значения формальных параметров по умолчанию. Инициализация ссылочной переменной. Параметры-ссылки и параметры указатели. const, inline, new, delete. Связывание с сохранением типов. указатель на void | |
| Потоки. Файлы и потоки. Вывод и ввод встроенных типов. Вывод и ввод определяемых пользователем типов. | |
| Перегрузка операций. Оператор присваивания. Присваивания. Инициализация. Перегружаемые операторы new и delete, ->. | |
| Дружественные функции и дружественные классы. Композиция. Дружественные классы. Написание программы с применением контернейных классов, дружественных классов. | |
| Правила доступа для друзей классов и производных классов к закрытым, защищенным и открытым частям описаний классов. Дружественные и производные классы. Процедуры для управления файлами и получения информации о файлах, хранящейся в операционной системе. | |

2.4. РАСПРЕДЕЛЕНИЕ БАЛЛОВ ПО МОДУЛЯМ И ВИДАМ УЧЕБНЫХ ЗАНЯТИЙ

| Модуль | Лк | | Лб | | СРС | | Всего | |
|-----------------|-----|-----|-----|-----|-----|-----|-------|-----|
| | min | max | min | max | min | max | min | max |
| I | 8 | 14 | 3 | 4 | 7 | 12 | 18 | 30 |
| II | 8 | 14 | 3 | 4 | 7 | 12 | 18 | 30 |
| Итоговый модуль | 10 | 16 | 5 | 8 | 10 | 16 | 25 | 40 |
| Всего | 26 | 44 | 11 | 16 | 24 | 40 | 61 | 100 |

Оценки в кредитной технологии обучения

| Оценка по буквенной системе (по 10-балльный) | % ное содержание (баллы) | Цифровой эквивалент баллов | Оценка балльной системе |
|--|--------------------------|----------------------------|-------------------------|
| A | 94 - 100 | 4,0 | отлично |
| A ⁻ | 88 – 93 | 3,67 | |
| B ⁺ | 85 – 87 | 3,33 | хорошо |
| B | 80 – 84 | 3,0 | |
| B ⁻ | 74 – 79 | 2,67 | |
| C ⁺ | 70 – 73 | 2,33 | удовлетворительно |
| C | 65 – 69 | 2,0 | |
| C ⁻ | 61 – 64 | 1,67 | |
| F | 0 – 60 | 0 | неудовлетворительно |

3. Перечень контрольных вопросов теоретического курса

1. Как определяются в языке C++ собственные функции пользователя?
2. Что такое прототип и что такое определение функции?
3. Как определять функции возвращающие значение?
4. Как определять функции, не возвращающие значение?
5. Как вызываются в программе функции возвращающие значение и функции, не возвращающие значение?
6. Каким образом можно использовать элемент массива в качестве параметра функции?
7. Расскажите о средствах отладки функций – заглушках и драйверах.
8. Область видимости переменных и констант.
9. Опишите, как использовать в качестве параметра функции массив целиком.
10. Что такое указатель? Какие операции возможны над указателями?
11. Поясните связь между массивами и указателями.
12. Как создается и уничтожается динамический одномерный массив?
13. Назначение и использование функций new и delete.
14. Создание и уничтожение двумерного динамического массива.
15. Что такое структура?
16. Что такое класс?
17. Что такое абстрактный тип данных?
18. Что такое перечисление?
19. Что такое объединение?
20. Как объявить класс с открытыми функциями и открытыми данными?
21. Как определяется функция доступа к закрытым переменным членам класса?
22. Что такое объект, как он объявляется и используется.
23. Как использовать структуру в качестве аргумента функции?
24. Каким образом осуществляется доступ к элементам данных и методам класса или объекта?
25. Что такое массив объектов?
26. Перечисления и объединения для чего они предназначены?
27. Каковы различия модульного и объектно- ориентированного программирования?
28. Назначение конструктора класса. Как вызывается конструктор?
29. Зачем нужен деструктор?
30. Как создается конструктор по умолчанию и конструктор с параметрами?
31. Что такое перегрузка функций?
32. Как создается перегрузка бинарных операций?
33. Как создается перегрузка унарных операций?
34. Как создается перегрузка операций потокового ввода /вывода?

35. Каковы правила перегрузки операций?
36. Какие исключения бывают при перегрузке операций?
37. Как создается конструктор копирования и зачем он используется?
38. Назовите концепции ООП?
39. Что такое полиморфизм?
40. Что такое инкапсуляция?
41. Что такое наследование?
42. Как создается одиночное наследование?
43. Что такое множественное наследование?
44. Как можно управлять доступом производных классов?
45. Как взаимодействуют конструкторы базовых и производных классов?
46. Что такое виртуальная функция?

4. Учебно - методическая поддержка курса

3. Бадд Т. Объектно-ориентированное программирование в действии: Пер. с англ. СПб.: Питер, 1997. 464 с,
4. Уолтер Сэвитч С++ в примерах. Издательство Эком, г. Москва, 1997 г.
5. Б. Страуструп. Язык программирования С++. Издательство Бином. Москва. (любое издание).
6. В.В. Подбельский. Язык С++. Издательство Финансы и статистика. г. Москва. 1999 г.
7. Д. Нейл, Ч. Уимз, М. Хедингтон. Учебник. Программирование на С++. Москва. 2000 г.
8. Ю. Тихомиров. Visual С++6. Новые возможности для программистов. Издательство ВHV. г. Санкт – Петербург. 1999 г.
9. Программирование на С++. Учебное пособие. Под редакцией А.Д. Хомоненко. Издательство КОРОНА Принт. г. Санкт – Петербург. 1999 г.
10. Т.А. Павловская. С/С++ программирование на языке высокого уровня. Издательство Питер. г. Санкт – Петербург. 2001 г.

Глоссарий

1. **Абстракция** - процесс изменения уровня детализации программы. Когда мы абстрагируемся от проблемы, мы предполагаем игнорирование ряда подробностей с тем, чтобы свести задачу к более простой.
2. **Диаграмма деятельности, Activity diagram** - методология объектно-ориентированного проектирования, предназначенная для детализации особенностей алгоритмической и логической организации системы. При этом каждое действие расчленяется на фундаментальные процессы. На диаграмме деятельности управление осуществляется:
 - либо через потоки управления (явно);
 - либо через определяемые потоки данных (неявно).
3. **Диаграмма компонентов, Component diagram** - метод объектно-ориентированного проектирования, описывающий особенности физического представления системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, устанавливая зависимости между компонентами.
4. **Инкапсуляция, Encapsulation** - От лат. In - в + Capsula - ящик, в объектно-ориентированном программировании - сокрытие внутренней структуры данных и реализации методов объекта от остальной программы. Другим объектам доступен только интерфейс объекта, через который осуществляется все взаимодействие с ним.
5. **Класс, Class** - Класс - в программировании - множество объектов, которые обладают одинаковой структурой, поведением и отношением с объектами из других классов.
6. **Конструкторы** - Эти операции используют в качестве аргументов объекты соответствующего им типа и создают другие объекты такого же типа. Например, операция сложения матриц создает новую матрицу.
7. **Локальность** - означает, что реализация одной абстракции может быть создана и рассмотрена без необходимости анализа реализации какой-либо другой абстракции. Принцип локальности позволяет составлять программу из абстракций, создаваемых людьми, работающими независимо друг от друга. Один человек может создать абстракцию, которая использует абстракцию, созданную кем-то другим.
8. **Наследование, Inheritance** - Наследование - в объектно-ориентированном программировании - свойство объекта, заключающееся в том, что характеристики одного объекта (объекта-предка) могут передаваться другому объекту (объекту-потомку) без их повторного описания. Наследование упрощает описание объектов.
9. **Объект, Object** - Объект - в программировании - программный модуль:
 - объединяющий в себе данные (свойства) и операции над ними (методы);
 - обладающий свойствами наследования, инкапсуляции и полиморфизма.Объекты взаимодействуют между собой, посылая друг другу сообщения.
10. **Объектно-ориентированное программирование** - технология программирования, при которой программа рассматривается как набор дискретных объектов, содержащих, в свою очередь, наборы структур данных и процедур, взаимодействующих с другими объектами.
11. **Полиморфизм, Polymorphism** - в объектно-ориентированном программировании - способность объекта выбирать правильный метод в зависимости от типа данных, полученных в сообщении.
12. **Свойство объекта** - в объектно-ориентированном программировании - характеристика объекта. Обычно свойства изменяются с помощью методов.
13. **Событийно-управляемое программирование** - объектно-ориентированное программирование, при котором задаются реакции программы на различные события.

14. **Спецификация** - описывает соглашение между разработчиками и пользователями. Разработчик берется написать модуль, а пользователь соглашается не полагаться на знания о том, как именно этот модуль реализован, т.е. не предполагать ничего такого, что не было бы указано в спецификации. Такое соглашение позволяет разделить анализ реализации от собственно использования программы. Спецификации дают возможность создавать логические основы, позволяющие успешно "разделять и властвовать".
15. **Технология программирования, Инжиниринг ПО, Software engineering** - дисциплина, изучающая технологические процессы программирования и порядок их прохождения. (см. онлайн-курс "Технология программирования")
16. **Экземпляр объекта, Instance** - в объектно-ориентированном программировании - конкретный объект из набора объектов данного класса. Все экземпляры одного класса имеют одинаковый набор операций.