

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»
(КГУ)

Кафедра «Безопасность информационных и автоматизированных систем»

УТВЕРЖДАЮ:
Ректор КГУ
/ Н.В. Дубив/
«31» августа 2020 г.



**ПРОГРАММА
ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ**

образовательной программы высшего образования –
программы специалитета

10.05.03 — Информационная безопасность автоматизированных систем

Направленность: (специализация №7) обеспечение информационной
безопасности распределенных информационных систем

Формы обучения: очная

Программа «Объектно-ориентированное программирование» составлена в соответствии с учебными планами по программе специалитета «Информационная безопасность автоматизированных систем» (обеспечение информационной безопасности распределенных информационных систем), утвержденным для очной формы обучения «28» августа 2020 года.

Рабочая программа дисциплины одобрена на заседании кафедры «Безопасность информационных и автоматизированных систем» 31 августа 2020 года, протокол № 1.

Программу составил:

канд. пед. наук, доцент



Т.А. Никифорова

Согласовано:

Заведующий кафедрой «БИАС»

канд. пед. наук, доцент



Е.Н. Полякова

Начальник Управления

образовательной деятельности



С.Н. Сеницын

Специалист по учебно-методической
работе Учебно-методического
отдела



Г.В. Казанкова

1. ОБЪЕМ ДИСЦИПЛИНЫ

Всего: 4 зачетных единицы трудоемкости (144 академических часа)

Очная форма обучения

Вид учебной работы	На всю дисциплину	Семестр
		6
Аудиторные занятия (контактная работа с преподавателем), всего часов в том числе:	42	42
Лекции	28	28
Лабораторные работы	14	14
Самостоятельная работа, всего часов в том числе:	102	102
Подготовка к зачету	18	18
Другие виды самостоятельной работы (подготовка к лабораторным работам и рубежному контролю)	84	84
Вид промежуточной аттестации	зачет	зачет
Общая трудоемкость дисциплины и трудоемкость по семестрам, часов	144	144

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Объектно-ориентированное программирование» относится к вариативным дисциплинам по выбору Блока 1.

Изучение дисциплины базируется на результатах обучения, сформированных при изучении следующих дисциплин: Информатика, Специальные главы информатики, Языки программирования, Технологии и методы программирования, Математическая логика и теория алгоритмов.

Планируемыми результатами освоения данной дисциплины являются способность применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач; способность ставить и решать задачи комплексного анализа, связанные с созданием и разработкой новых информационных технологий и информационных систем, с использованием базовых и специальных знаний, современных аналитических методов, математических и информационных моделей; способность разрабатывать новые и модернизировать уже существующие информационные технологии и информационные системы в соответствии с техническим заданием, а также выполнение курсовых работ, курсовых проектов и выпускной квалификационной работы.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Целью изучения дисциплины «Объектно-ориентированное программирование» является формирование общепрофессиональных и специальных компетентностей посредством знакомства студентов с парадигмой объектно-ориентированного программирования, с базовыми понятиями объектно-ориентированного программирования; формирование умения анализировать предметную область в соответствии с поставленной задачей; на основе проведенного анализа выбрать соответствующее программное обеспечение или язык объектно-ориентированного программирования для реализации практической задачи. Целью изучения дисциплины «Объектно-ориентированное программирование» является получение навыков написания программ с использованием парадигмы объектно-ориентированного программирования.

Задачами дисциплины являются: изучение основных понятий объектно-ориентированного программирования (ООП), принципов ООП, основ языка программирования C++, использование языка ООП для решения практических задач; формирование умений осуществить все этапы компьютерного моделирования, а именно: анализировать стоящие перед студентом практические задачи; на основе проведенного анализа выбрать соответствующий язык программирования для решения практических задач и реализовать разработанный алгоритм решения с использованием выбранного языка программирования, а также провести тестирование работы ПО; способность применять к решению прикладных задач базовые алгоритмы обработки информации, выполнять оценку сложности алгоритмов, программировать и тестировать разработанные программы.

Компетенции, формируемые в результате освоения дисциплины:

- способность применять языки, системы и инструментальные средства программирования в профессиональной деятельности (ОПК-3);
- способность к освоению новых образцов программных, технических средств и информационных технологий (ОПК-8);
- способность разрабатывать и анализировать проектные решения по обеспечению безопасности автоматизированных систем (ПК-8).

В результате изучения дисциплины студент должен:

знать:

- методы, способы и средства получения, хранения, переработки информации (для ОПК-3);
- способы моделирования и проектирования структуры данных и знаний, прикладных и информационных процессов (для ПК-8);
- опасности и угрозы, возникающие в процессе развития информационного общества (для ПК-8);

уметь:

- применять методы, способы и средства получения, хранения, переработки информации (для ОПК-3, ПК-8);
- формировать архитектуру программных комплексов для информатизации предприятий, разрабатывать программное обеспечение и приложения (для ОПК-8);
- разрабатывать программное обеспечение средствами объектно-ориентированного программирования на языке С++ (для ОПК-8; ПК-8);

владеть:

- навыками программирования в современных средах (для ОПК-3, ОПК-8, ПК-8);
- методами использования современных методов объектно-ориентированного программирования при кодировании программных систем разного уровня сложности (для ОПК-3, ОПК-8, ПК-8).

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Учебно-тематический план. Очная форма обучения

Рубеж	Номер темы	Наименование темы	Количество часов контактной работы с преподавателем	
			Лекции	Лаборат. работы
Рубеж 1	1	Основные конструкции языка С++.	8	8
	2	Классы и их компоненты.	4	4
Рубеж 2	3	Дружественные функции и классы.	4	2
	4	Перегрузка стандартных операций.	4	-
	5	Наследование.	4	-
	6	Шаблоны.	4	-
Всего:			28	14

4.2. Содержание лекционных занятий

Тема 1. Основные конструкции языка C++.

Лексические основы языка C++, Алфавит. Лексемы. Скалярные типы и выражения. Примеры. Использование простейших типов. Примеры. Функции форматного ввода и вывода.

Операции. Знаки операций. Арифметические операции. Примеры. Инкремент и декремент. Примеры. Операции присваивания и отношения. Примеры. Логические операции. Примеры. Условная операция. Примеры. Операции преобразования типов. Примеры.

Условные конструкции. Оператор if...else. Примеры. Оператор switch. Примеры.

Циклические конструкции. Цикл while. Примеры. Цикл do...while. Примеры. Цикл for. Примеры. Оператор безусловного перехода. Примеры. Оператор принудительного выхода из цикла или переключателя. Примеры. Оператор завершения выполнения текущего шага тела цикла. Примеры. Оператор возвращающий значение из функции. Примеры.

Массивы. Одномерные массивы. Имя массива. Двумерные массивы. Многомерные массивы. Работа с массивами с помощью указателей. Примеры. Массивы указателей. Примеры. Массивы динамической памяти. Примеры.

Функции. Определение функции. Примеры. Аргументы функции. Примеры. Передача нескольких значений в функцию. Возврат нескольких значений из функции. Примеры. Ссылочные переменные. Примеры. Рекурсия. Примеры. Функции с переменным числом параметров. Примеры. Подставляемые функции. Способы передачи массивов в функции и их возврата. Примеры.

Строки. Строковая константа. Инициализация строк. Строки и указатели. Примеры. Функции работы со строками. Примеры. Использование строк в командной строке. Примеры.

Работа с файлами. Функции высокоуровневого ввода/вывода. Примеры. Форматный ввод/вывод. Примеры. Функции низкоуровневого ввода/вывода. Примеры.

Структуры и объединения. Описание структур. Примеры. Операции над структурами. Примеры. Использование полей битов в качестве полей структур. Примеры. Понятие объединения. Переменные структуры.

Тема 2. Классы и их компоненты.

Классы. Класс как расширение понятия структуры. Примеры. Способы доступа к компонентам класса. Примеры. Использование указателей для доступа к компонентам класса. Примеры. Конструкторы и доступность компонентов класса. Примеры. Деструкторы. Примеры.

Компоненты класса. Данные класса. Статические компоненты класса. Примеры. Указатели на компоненты класса. Примеры. Определение методов класса. Примеры. Указатель this. Примеры.

Тема 3. Дружественные функции и классы.

Дружественные функции. Примеры. Дружественные классы. Примеры.

Тема 4. Перегрузка стандартных операций.

Перегрузка стандартных операций. Пример. Операция-функция. Примеры. Операции, которые не перегружаются. Примеры. Операция инкремент и декремент. Примеры.

Тема 5. Наследование.

Общие сведения о механизме наследования. Определение производного класса. Примеры. Особенности использования деструкторов при наследовании. Пример. Множественное наследование. Примеры. Дублирование классов. Примеры. Виртуальные базовые классы. Примеры. Поведение неvirtуальных компонентных функций при наследовании. Примеры. Виртуальные функции. Статическое и динамическое связывание. Примеры. Абстрактные классы. Примеры. Локальные классы. Примеры.

Тема 6. Шаблоны.

Шаблоны функций. Примеры. Шаблоны классов. Примеры.

4.3 Лабораторные работы

Номер темы	Наименование темы	Наименование тем лабораторных работ	Норматив времени, час.
1	Основные конструкции языка C++	Лекционные основы языка C++, Операции.	1
		Условные конструкции. Циклические конструкции.	1
		Массивы.	2
		Строки.	1
		Работа с файлами.	2
	<i>1-ый рубежный контроль</i>	<i>Тестирование</i>	<i>1</i>
2	Классы и их компоненты.	Структуры и объединения.	1
		Классы и компоненты	2
3	Дружественные функции и классы.	Функции.	2
	<i>2-ый рубежный контроль</i>	<i>Тестирование</i>	<i>1</i>
<i>Итого:</i>			<i>14</i>

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

При прослушивании лекций рекомендуется в конспекте отмечать все важные моменты, на которых заостряет внимание преподаватель, в частности те, которые направлены на качественное выполнение соответствующей лабораторной работы.

Преподавателем запланировано использование при чтении лекций технологии учебной дискуссии. Поэтому рекомендуется фиксировать для себя интересные моменты с целью их активного обсуждения на дискуссии в конце лекции.

Залогом качественного выполнения лабораторных работ является самостоятельная подготовка к ним накануне путем повторения материалов лекций. Реко-

мендуется подготовить вопросы по неясным моментам и обсудить их с преподавателем в начале лабораторной работы.

Преподавателем запланировано применение на лабораторных работах технологий развивающейся кооперации, коллективного взаимодействия, разбора конкретных ситуаций.

Для текущего контроля успеваемости по очной форме обучения преподавателем используется балльно-рейтинговая система контроля и оценки академической активности. Поэтому настоятельно рекомендуется тщательно прорабатывать материал дисциплины при самостоятельной работе, участвовать во всех формах обсуждения и взаимодействия, как на лекциях, так и на лабораторных работах в целях лучшего освоения материала и получения высокой оценки по результатам освоения дисциплины.

Выполнение самостоятельной работы подразумевает самостоятельное изучение разделов дисциплины, подготовку к лабораторным работам, к рубежным контролям и подготовку к зачету.

Рекомендуемая трудоемкость самостоятельной работы представлена в таблице:

Рекомендуемый режим самостоятельной работы

Наименование вида самостоятельной работы	Рекомендуемая трудоемкость, акад. час.
Углубленное изучение разделов, тем дисциплины лекционного курса Изучение разделов, тем дисциплины, не вошедших в лекционный курс, а именно: Шаблоны. Общие сведения о механизме наследования. Определение производного класса. Примеры. Особенности использования деструкторов при наследовании. Пример. Множественное наследование. Примеры. Дублирование классов. Примеры. Виртуальные базовые классы. Примеры. Поведение неvirtуальных компонентных функций при наследовании. Примеры. Виртуальные функции. Статическое и динамическое связывание. Примеры. Абстрактные классы. Примеры. Локальные классы. Примеры	52
Подготовка к лабораторным работам (по 4 ч на каждое занятие)	28
Подготовка к рубежным контролям (по 2 часа на каждый рубежный контроль)	4
Подготовка к зачету	18
Всего:	102

6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

6.1. Перечень оценочных средств

1. Балльно-рейтинговая система контроля и оценки академической активности студентов в КГУ.
2. Отчеты студентов по лабораторным работам.
3. Банк тестовых заданий к рубежным контролям № 1, № 2.
4. Вопросы к зачету

6.2. Система балльно-рейтинговой оценки работы студентов по дисциплине

№	Наименование	Содержание					
		Распределение баллов					
1	Распределение баллов за семестры по видам учебной работы, сроки сдачи учебной работы (доводятся до сведения студентов на первом учебном занятии)	Вид учебной работы:	Посещение лекций	Выполнение лабораторной работы	Рубежный контроль №1	Рубежный контроль №2	Зачет
		Балльная оценка:	1 _б x 14 = 14 _б	4 _б x 8 = 32 _б	12	12	30
2	Критерий пересчета баллов в традиционную оценку по итогам работы в семестре и зачета	60 и менее баллов – неудовлетворительно; не зачтено; 61...73 – удовлетворительно; зачтено; 74... 90 – хорошо; 91...100 – отлично					
3	Критерии допуска к промежуточной аттестации, возможности получения автоматического зачета по дисциплине, возможность получения бонусных баллов	<p>Для допуска к промежуточной аттестации (зачету) студент должен набрать по итогам текущего и рубежного контроля не менее 50 баллов и должен выполнить все лабораторные работы.</p> <p>Для получения зачета «автоматически» студенту необходимо набрать 61 балл. По согласованию с преподавателем студенту, могут быть добавлены дополнительные (бонусные) баллы за активность на лабораторных работах, активное участие в научной и методической работе, оригинальность принятых решений в ходе выполнения лабораторных работ, за участие в значимых учебных и внеучебных мероприятиях кафедры.</p>					
4	Формы и виды учебной работы для неуспевающих (восстановившихся на курсе обучения) студентов для получения недостающих баллов в конце семестра	<p>В случае если к промежуточной аттестации (зачету) набрана сумма менее 50 баллов, студенту необходимо набрать недостающее количество баллов за счет выполнения дополнительных заданий, до конца последней (зачетной) недели семестра. При этом необходимо проработать материал всех пропущенных лабораторных работ.</p> <p>Формы дополнительных заданий (назначаются преподавателем):</p> <ul style="list-style-type: none"> - выполнение пропущенной лабораторной работы (при невозможности дополнительного проведения лабораторной работы преподаватель устанавливает форму дополнительного задания по тематике пропущенной работы самостоятельно) – до 4 баллов. <p>Ликвидация академических задолженностей, возникших из-за разности в учебных планах при переводе или восстановлении, проводится путем выполнения дополнительных заданий, форма и объем которых определяется преподавателем.</p>					

6.3. Процедура оценивания результатов освоения дисциплины

Рубежные контроли проводятся в форме письменного тестирования.

Перед проведением каждого рубежного контроля преподаватель прорабатывает со студентами основной материал соответствующих разделов дисциплины в форме краткой лекции-дискуссии. Варианты тестовых заданий состоят

для 1 и 2 рубежного контроля из 12 вопросов. На каждое тестирование при рубежном контроле студенту отводится 1 академических часа.

Преподаватель оценивает в баллах результаты тестирования каждого студента по количеству правильных ответов и заносит в ведомость учета текущей успеваемости.

Зачет состоит из 2 вопросов. Вопросы к зачету доводятся до студентов на последней лекции в семестре. Каждый вопрос оценивается в 15 баллов. На подготовку ответа студенту отводится 1 астрономический час.

Результаты текущего контроля успеваемости и зачета заносятся преподавателем в зачетную ведомость, которая сдается в организационный отдел института в день зачета, а также выставляются в зачетную книжку студента.

6.4. Примеры оценочных средств для рубежных контролей и зачета

1-ый рубежный контроль

Пример 1. Что напечатает следующая программа?

```
#include<iostream.h>
main ()
{
    int x;
    x = -3+4*5-6; cout<<x<<" "; /* Операция 1 */
    x = 3+4%5-6; cout<<x<<" "; /* Операция 2 */
    x = -3*4%-6/5; cout<<x<<" "; /* Операция 3 */
    x = (7+6)%5/2; cout<<x<<" "; /* Операция 4 */
}
```

Пример 2. Что напечатает следующая программа.

```
#include<iostream.h>
#include<stdio.h>
main()
{
    int a,b,c,v,k;
    cout<<"Задайте целое число: ";
    cin>>v;
    k=v;
    cout<<"      До   Во время   После"<<endl;
    v=k;a=v;b=v++;c=v; cout<<a,b,c;
    v=k;a=v;b=v--;c=v; cout<<a,b,c;
    v=k;a=v;b=++v;c=v; cout<<a,b,c;
    v=k;a=v;b=--v;c=v; cout<<a,b,c;
}
```

Пример 3. Проиллюстрируйте в программе применение логических операций и операций увеличения.

Пример 4. Выдать на печать в обратном порядке цифры целого положительного числа N.

Пример 5. Напишите программу для замены в слове X всех букв "а" на сочетание "ку".

Пример 6. Напишите программу, удваивающую каждую букву слова X.

Пример 7. Вычеркните из слова *X* буквы, стоящие на четных местах.

Пример 8. Составить массив, каждый элемент которого равен максимуму из соответствующих значений двух других массивов.

2-ой рубежный контроль

Рассмотрим класс, описывающий товары на складе магазина. Компонентами класса будут:

- название товара;
- оптовая (закупочная) цена;
- розничная (торговая) наценка;
- функция ввода данных о товаре;
- функция печати (вывода на дисплей) сведений о товаре с указанием розничной цены.

Определение класса:

```
#include <iostream.h>
struct goods // Определение класса "товары".
{ char name[40]; // Наименование товара.
  float price; // Оптовая (закупочная) цена.
  static int percent; // Торговая наценка, в %.
  // Компонентные функции:
  void Input() // Ввод сведений о товаре.
  { cout << "Наименование товара: ";
    cin >> name;
    cout << "Закупочная цена: ";
    cin >> price;
  }
  void Display() // Вывод данных о продаваемом товаре.
  { cout << "\n" << name;
    cout << ", розничная цена: ";
    cout << long(price * (1.0 + goods::percent * 0.01));
  }
};
```

Используя определение данного класса, составить программу для заполнения базы данных товаров на складе магазина.

1. Создать шаблон семейства функций для обмена значений двух передаваемых им параметров.
2. Создать шаблон, который определяет семейство функций, каждая из которых подсчитывает количество нулевых элементов одномерного массива параметризованного типа.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ВОПРОСОВ К ЗАЧЕТУ

1. Алфавит языка C++. Лексемы языка C++.
2. Знаки операций в языке C++.
3. Скалярные типы и выражения в языке C++. Примеры использования простейших типов в языке C++.
4. Функции форматного ввода и вывода. Примеры.

5. Арифметические операции в языке C++. Примеры.
6. Инкремент и декремент. Примеры.
7. Операции присваивания и отношения. Примеры.
8. Логические операции. Примеры.
9. Условная операция. Примеры.
10. Операция sizeof. Примеры.
11. Операции преобразования типов. Примеры.
12. Оператор if...else. Примеры.
13. Оператор switch. Примеры.
14. Цикл while. Примеры.
15. Цикл do...while. Примеры.
16. Цикл for. Примеры.
17. Оператор безусловного перехода. Примеры.
18. Оператор принудительного выхода из цикла или переключателя. Примеры.
19. Оператор завершения выполнения текущего шага тела цикла. Примеры.
20. Оператор возвращающий значение из функции. Примеры.
21. Одномерные массивы. Примеры.
22. Двумерные массивы. Примеры.
23. Многомерные массивы. Примеры.
24. Работа с массивами с помощью указателей. Примеры.
25. Массивы указателей. Примеры.
26. Массивы динамической памяти. Примеры.
27. Определение функции. Примеры.
28. Аргументы функции. Примеры.
29. Передача нескольких значений в функцию. Возврат нескольких значений из функции. Примеры.
30. Ссылочные переменные. Примеры.
31. Рекурсия. Примеры.
32. Функции с переменным числом параметров. Примеры.
33. Подставляемые функции. Примеры.
34. Способы передачи массивов в функции и их возврата. Примеры.
35. Автоматический класс памяти (auto). Примеры.
36. Внешний класс памяти (extern). Примеры.
37. Внешние глобальные объекты (extern). Примеры.
38. Внешние статические объекты (static). Примеры.
39. Регистровый класс памяти (register). Примеры.
40. Строковая константа. Инициализация строк. Строки и указатели. Примеры.
41. Функции работы со строками. Примеры.
42. Использование строк в командной строке. Примеры.
43. Функции высокоуровневого ввода/вывода. Примеры.
44. Форматный ввод/вывод. Примеры.
45. Функции низкоуровневого ввода/вывода. Примеры.
46. Описание структур. Примеры.

47. Операции над структурами. Примеры.
48. Использование полей битов в качестве полей структур. Примеры.
49. Понятие объединения. Переменные структуры.
50. 8.2.2. Примерный список вопросов к экзамену:
51. Класс как расширение понятия структуры. Примеры.
52. Способы доступа к компонентам класса. Примеры.
53. Использование указателей для доступа к компонентам класса. Примеры.
54. Конструкторы и доступность компонентов класса. Примеры.
55. Деструкторы. Примеры.
56. Данные класса. Статические компоненты класса. Примеры.
57. Указатели на компоненты класса. Примеры.
58. Определение методов класса. Примеры.
59. Указатель this. Примеры.
60. Дружественные функции. Примеры.
61. Дружественные классы. Примеры.
62. Перегрузка стандартных операций. Пример.
63. Операция-функция. Примеры.
64. Операции, которые не перегружаются. Примеры.
65. Операция инкремент и декремент. Примеры.
66. Общие сведения о механизме наследования.
67. Определение производного класса. Примеры.
68. Особенности использования деструкторов при наследовании. Пример.
69. Множественное наследование. Примеры.
70. Дублирование классов. Примеры.
71. Виртуальные базовые классы. Примеры.
72. Поведение неvirtуальных компонентных функций при наследовании. Примеры.
73. Виртуальные функции. Статическое и динамическое связывание. Примеры.
74. Абстрактные классы. Примеры.
75. Локальные классы. Примеры.
76. Шаблоны функций. Примеры.
77. Шаблоны классов. Примеры.

6.5. Фонд оценочных средств

Полный банк заданий для текущего, рубежных контролей и промежуточной аттестации по дисциплине, показатели, критерии, шкалы оценивания компетенций, методические материалы, определяющие процедуры оценивания образовательных результатов, приведены в учебно-методическом комплексе дисциплины.

7. ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ УЧЕБНАЯ ЛИТЕРАТУРА

7.1. Основная учебная литература

1. Литвиненко Н.А. Технология программирования на С++. Начальный курс [Электронный ресурс]. - БХВ-Петербург, 2010. – 281 с. – Доступ из ЭБС «znanium.com»
2. Романов Е.Л. Си++. От дилетанта до профессионала [Электронный ресурс]. - М.: Финансы и статистика, 2014. -600 с. – Доступ из ЭБС «znanium.com»
3. Страуструп Б. Язык программирования С++. Специальное издание. – М.Ж БИНОМ. Лаборатория знаний, 2011. – 1136 с. – Доступ из ЭБС «znanium.com»
4. Уильямс Э. Параллельное программирование на С++ в действии [Электронный ресурс]. - М.: Финансы и статистика, 2012. – 672 с. – Доступ из ЭБС «znanium.com»

7.2 Дополнительная учебная литература:

1. Подбельский В.В. Язык С++ [Электронный ресурс]: Учебное пособие. – 5-е изд. – М.: Финансы и статистика, 2003. – 560 с.: ил. – Доступ из ЭБС «Консультант студента»
2. Страуструп Б. Язык программирования С++ [Электронный ресурс]. – М.: Радио и связь, 1991. – 352 с. URL: <http://lib.ru/%3E%3C/CPPII/cpptut.txt>
3. Уэйт М., Прата С., Мартин Д. Язык Си - руководство для начинающих [Электронный ресурс]. - М.: Мир, 1988. – 512 с. URL: <https://studfiles.net/preview/4273213/> (Дата обращения: 16.02.2017)
4. Швецкий М.В., Лебедева И.А., Медведев А.А. Динамические структуры данных в языке программирования Си++ [Электронный ресурс]: Учебное пособие. – Курган: Изд-во Курганского гос. ун-та, 2001. – 220 с. – Доступ из ЭБС КГУ.

7.3 Учебно-методическая литература:

1. Медведев А.А. Основы языка программирования Си++ [Электронный ресурс]. Методические рекомендации для проведения лабораторных работ - Курган: Изд-во РИЦ КГУ, 2014. – 62 с. – Доступ из ЭБС КГУ.
2. Швецкий М.В., Медведев А.А. Динамические структуры данных в языке программирования Си++ [Электронный ресурс]. Методические рекомендации для проведения лабораторных работ - Курган: Изд-во РИЦ КГУ, 2014. – 91 с. – Доступ из ЭБС КГУ.

8. РЕСУРСЫ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫЕ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. Электронно-библиотечная система научно-издательского центра «ИНФРА-М». – Режим доступа: <http://znanium.com/>. – загл. с экрана.
2. Электронно-библиотечная система издательства «Лань». – Режим доступа: <http://e.lanbook.com/>. – загл. с экрана.
3. <https://code-live.ru/tag/cpp-manual/>. Самоучитель по основам языка С++.
4. avolberg.ru/theory/oop. Теоретические основы ООП для начинающих.

9. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ

Лекционные занятия проходят в мультимедийной аудитории, оснащенной компьютером и проектором.

Лабораторные занятия проходят в компьютерном классе. Программное обеспечение: ОС Windows 2000, XP и выше; Microsoft Office 2000 и выше; Borland C++ 4.5.

10. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

1. Переносной проектор BENQ PB6110 с экраном, локальная сеть компьютеров на базе Intel Core i3-2120 - 16 шт. с выходом в Internet, коммутатор 2-го уровня D-LINK DGS-101D/E1A.

2. Программное обеспечение: Microsoft Windows XP/2000, Windows 7.

Аннотация к рабочей программе дисциплины
«Объектно-ориентированное программирование»

образовательной программы высшего образования –
программы специалитета

10.05.03 – Информационная безопасность автоматизированных систем

Специализация:

Обеспечение информационной безопасности распределенных
информационных систем

Трудоемкость дисциплины: 4 з.е. (144 академических часа)

Семестр: 6 (очная форма обучения)

Форма промежуточной аттестации: зачет

Содержание дисциплины. Основные разделы

Отличительные особенности языка программирования C++. Основные конструкции C++. Классы и их компоненты. Дружественные функции и классы. Перегрузка стандартных операций. Наследование. Шаблоны.