

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Воронежский государственный технический университет»

УТВЕРЖДАЮ

Декан факультета информационных технологий  
и компьютерной безопасности

  
П.Ю. Гусев/

2023 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**  
**«Объектно-ориентированное программирование»**

Направление подготовки 09.03.02 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И  
ТЕХНОЛОГИИ

Профиль Отраслевые информационные системы

Квалификация выпускника бакалавр

Нормативный период обучения 4 года

Форма обучения очная

Год начала подготовки 2023

Автор программы



/Тишуков Б.Н./

Заведующий кафедрой  
систем  
автоматизированного  
проектирования и  
информационных систем



/Львович Я.Е./

Руководитель ОПОП



/Курипта О.В./

Воронеж 2023

# 1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

## 1.1. Цели дисциплины

изучение объектно-ориентированной методологии программирования, изучение основных понятий объектно-ориентированного программирования.

## 1.2. Задачи освоения дисциплины

– к теоретическим задачам относятся: обучение разработке программного обеспечения в объектно-ориентированной парадигме; изучение основ программирования на языке C++;

– прикладные задачи состоят в приобретении навыков проектирования и разработки программного обеспечения с применением объектно-ориентированного подхода к программированию.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Объектно-ориентированное программирование» относится к дисциплинам базовой части блока Б1.

## 3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Объектно-ориентированное программирование» направлен на 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;

ОПК-4 - Способен участвовать в разработке технической документации, связанной с профессиональной деятельностью с использованием стандартов, норм и правил;

ОПК-6 - Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий.

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ОПК-2	знать методы и технологии программирования, принципы и определения объектно-ориентированного подхода
	уметь описывать основные структуры данных на основе

	объектно-ориентированного подхода
	владеть навыками работы в различных средах программирования
ОПК-4	знать виды технической документации поддержки программного кода
	уметь описывать техническую документацию
	владеть навыками оформления программного кода
ОПК-6	знать основные принципы конструирования алгоритмов
	уметь разрабатывать и реализовывать алгоритмы на языке высокого уровня
	владеть методами описания основных классов и алгоритмов на их основе

#### 4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Объектно-ориентированное программирование» составляет 6 з.е.

Распределение трудоемкости дисциплины по видам занятий

Виды учебной работы	Всего часов	Семестры
		2
<b>Аудиторные занятия (всего)</b>	90	90
В том числе:		
Лекции	36	36
Лабораторные работы (ЛР)	54	54
<b>Самостоятельная работа</b>	90	90
<b>Курсовой проект</b>	+	+
Часы на контроль	36	36
Виды промежуточной аттестации - экзамен	+	+

Общая трудоемкость: академические часы зач.ед.	216 6	216 6
--	----------	----------

## 5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

### 5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего , час
1	Общие сведения, Основы программирования на С++	<p><b>Введение.</b> Необходимость объектно-ориентированное программирование, характеристики объектно-ориентированных языков</p> <p><b>Основы программирования на С++.</b> История С++, сравнение с языком С. Структура программы на языке С++. Директивы, комментарии. Переменные. Преобразование типов. Арифметические операции. Обзор интегрированных сред разработки (IDE).</p>	8	12	18	38
2	Циклы и ветвления, структуры и функции	<p><b>Циклы и ветвления.</b> Операции отношения. Циклы и ветвления. Логические операции. Приоритеты операций.</p> <p><b>Структуры.</b> Использование структур, перечислений.</p> <p><b>Функции в С++</b></p>	8	12	18	38
3	Объектно-ориентированный подход к разработке программных средств	<p><b>Введение в классы и объекты.</b> Описание и свойства объектов. Конструкторы, деструкторы классов, объявления статических объектов, переменных и методов в классе.</p> <p><b>Наследование в ООП.</b> Наследование, виды наследования. Абстрактные классы. Множественное наследование.</p>	8	10	18	36

4	Потоки и файлы. Многофайловые программы.	<b>Потоки и файлы.</b> Потокосые классы. Потокосый ввод/вывод дисковых файлов. Указатели файлов. Файловый ввод/вывод с помощью методов.  <b>Многофайловые программы.</b> Создание многофайловой программы. Межфайловое взаимодействие.	6	10	18	34
5	Шаблоны и исключения. Стандартная библиотека шаблонов.	<b>Шаблоны и исключения.</b> Шаблоны функций и классов. Исключения.  <b>Стандартная библиотека шаблонов.</b> Последовательные контейнеры. Итераторы. Специальные итераторы. Ассоциативные контейнеры. Хранение пользовательских объектов. Функциональные объекты.	6	10	18	34
<b>Итого</b>			<b>36</b>	<b>54</b>	<b>90</b>	<b>180</b>

## 5.2 Перечень лабораторных работ

Лабораторная работа № 1. Базовые конструкции языка C++. Управляющие конструкции.

Лабораторная работа № 2. Работа с массивами.

Лабораторная работа № 3. Подпрограммы.

Лабораторная работа № 4. Работа со строками и файлами.

Лабораторная работа № 5. Классы, свойства, индексаторы.

Лабораторная работа № 6. Наследование.

Лабораторная работа № 7. Виртуальные функции и полиморфизм.

Лабораторная работа № 8. Обработка исключений.

Лабораторная работа № 9. Регулярные выражения. Шаблоны.

Лабораторная работа № 10. Многопоточное программирование.

Лабораторная работа № 11. Абстракция и инкапсуляция.

## 6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсового проекта в 2 семестре.

Примерная тематика курсового проекта: «Разработка программного обеспечения с использованием объектно-ориентированного подхода (по вариантам)».

В рамках выполнения курсового проекта студенту необходимо для предметной области, соответствующей выбранному варианту, провести ее анализ, спроектировать информационную систему, основываясь на основных принципах и методах ООП, разработать ее математическую модель и реализовать систему в виде программного продукта на языке C++.

## 7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

**7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания**

### 7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ОПК-2	знать методы и технологии программирования, принципы и определения объектно-ориентированного подхода	Тестирование Проверка этапов выполнения курсового проекта Результаты проведения коллоквиума Ответы на теоретические вопросы при защите лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

	уметь описывать основные структуры данных на основе объектно-ориентированного подхода	Эффективность использования изученного теоретического материала при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть навыками работы в различных средах программирования	Владение языком программирования C++ при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-4	знать виды технической документации поддержки программного кода	Тестирование Проверка этапов выполнения курсового проекта  Результаты проведения коллоквиума  Ответы на теоретические вопросы при защите лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь описывать техническую документацию	Эффективность использования изученного теоретического материала при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть навыками оформления программного кода	Владение способами описания методики использования программного средства при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-6	знать основные принципы конструирования алгоритмов	Тестирование Проверка этапов выполнения курсового проекта  Результаты проведения коллоквиума  Ответы на теоретические вопросы при защите лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

	уметь разрабатывать и реализовывать алгоритмы на языке высокого уровня	Эффективность использования изученного теоретического материала при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть методами описания основных классов и алгоритмов на их основе	Владение способами описания методики использования программного средства при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

### 7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются во 2 семестре для очной формы обучения, 2 семестре для заочной формы обучения по четырехбалльной системе:

- «отлично»;
- «хорошо»;
- «удовлетворительно»;
- «неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
ОПК-2	знать методы и технологии программирования, принципы и определения объектно-ориентированного подхода	Тест	Выполнение теста на 90- 100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь описывать основные структуры данных на основе объектно-ориентированного подхода	Тес	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены



	владеть навыками работы в различных средах программирования	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-4	знать виды технической документации поддержки программного кода	Тест	Выполнение теста на 90- 100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь описывать техническую документацию	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть навыками оформления программного кода	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-6	знать основные принципы конструирования алгоритмов	Тест	Выполнение теста на 90- 100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь разрабатывать и реализовывать алгоритмы на языке высокого уровня	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

	владеть методами описания основных классов и алгоритмов на их основе	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
--	--	--	--	---	--	------------------

---

## 7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

### 7.2.1 Примерный перечень заданий для подготовки к тестированию

1. Дополнительная информация, передающаяся при генерации исключения, может быть помещена в:
  - A. ключевое слово `throw`;
  - B. функцию, вызвавшую ошибку;
  - C. блок-ловушку;
  - D. объект класса исключений.
2. Чтобы определять объекты класса в разных файлах, в каждом из них необходимо:
  - A. объявлять класс;
  - B. определять класс;
  - C. объявлять класс с использованием `extern`;
  - D. определять класс с использованием `extern`.
3. Оператор присваивания может быть перегружен с целью:
  - A. хранения информации о количестве одинаковых объектов;
  - B. присваивания идентификационного номера каждому объекту;
  - C. проверки того, что все компонентные данные скопировались без ошибок;
  - D. уведомления о том, что имело место присваивание.
4. Истинно ли утверждение о том, что указатель на базовый класс может ссылаться на объекты порожденного класса?
  - A. да;
  - B. нет.
5. Оператор разрешения обычно:
  - A. ограничивает видимость переменных для определенных методов;
  - B. обозначает, от какого базового класса создан производный;
  - C. определяет отдельный класс;
  - D. разрешает неопределенности.

## 7.2.2 Примерный перечень заданий для решения стандартных задач

1. Предположим, что класс `X` не использует перегруженные операции. Напишите выражение, в котором вычитается объект `x1` класса `X` из другого объекта `x2` этого же класса. Результат помещается в `x3`.
2. Истинно ли следующее утверждение: переменная типа `char` может хранить значение 301?
3. Опишите структуру, содержащую три переменные типа `int` с названиями `hrs`, `mins` и `secs`. Назовите структуру именем `time`.
4. Напишите функцию `foo()`, выводящую на экран слово `foo`.

5. Дана следующая функция:

```
int times2(int a)
{
    return (a * 2);
}
```

Напишите функцию `main()`, которая будет содержать все необходимое для вызова данной функции.

### 7.2.3 Примерный перечень заданий для решения прикладных задач

1. Истинно ли следующее утверждение: если вы определили конструктор, содержащий определение типа `aclass obj = intvar;`, вы также можете записать выражение типа `obj = intvar;`?

2. Пусть указатель `p` ссылается на объекты базового класса и содержит адрес объекта порожденного класса. Пусть в обоих этих классах имеется не виртуальный метод `ding()`. Тогда выражение `p->ding();` поставит на выполнение версию функции `ding()` из \_\_\_\_\_ класса.

3. Объясните разницу в выполнении следующих двух выражений:  
`person p1(p0);` `person p1 = p0;`

4. Напишите `if`, определяющий, достиг объект `ifstream` под названием `foobar` конца файла или же возникла ошибка.

5. Напишите выражение, записывающее единичный символ в объект `fileOut` класса `ofstream`.

### 7.2.4 Примерный перечень вопросов для подготовки к зачету

Не предусмотрено учебным планом

### 7.2.5 Примерный перечень вопросов для подготовки к экзамену

1. Виды переменных в C++
2. Понятие класса в C++. Применение конструкторов и деструкторов класса.
3. Работа с массивами в C++.
4. Области видимости переменных и методов в классе.
5. Наследование в C++. Примеры простого и множественного наследования.
6. Указатели на объекты.
7. Работа с потоковыми классами в C++.
8. Поточный ввод/вывод файлов.
9. Компиляция программы на C++ с использованием консоли и компилятора `g++`.
10. Стандартная библиотека шаблонов (STL).

### **7.2.6. Методика выставления оценки при проведении промежуточной аттестации**

Оценка при проведении промежуточной аттестации учитывает результаты тестирования. Экзамен проводится по экзаменационным билетам, каждый из которых содержит два теоретических вопроса и одно практическое задание, тестирование предполагает получение ответов на 10 вопросов.

1. Оценка «Неудовлетворительно» ставится в случае, если студент дал неправильные ответы на два экзаменационных вопроса, не решил практическое задание и ответил менее чем на 60% тестовых вопросов.

2. Оценка «Удовлетворительно» ставится в случае, если студент неправильно ответил на один экзаменационный вопрос или на все вопросы дал неполные ответы, не решил полностью практическое задание и ответил на 60-80 % тестовых вопросов.

3. Оценка «Хорошо» ставится в случае, если студент правильно ответил на один экзаменационный вопрос или на все вопросы дал неполные ответы, решил полностью практическое задание, продемонстрировал понимание материала, но допустил незначительные ошибки, а также выполнил тест на 80-90%.

4. Оценка «Отлично» ставится, если студент дал обоснованные, глубокие и теоретически правильные ответы на экзаменационные вопросы, решил полностью практическое задание, продемонстрировал понимание материала, а также выполнил тест на 90-100%.

Компетенции считаются сформированными, если в ходе изучения дисциплины выполнены и защищены курсовой проект и лабораторные работы. Выполненные лабораторные работы являются допуском к сдаче экзамена.

### **7.2.7 Паспорт оценочных материалов**

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Общие сведения, Основы программирования на C++	ОПК-2, ОПК-6	Тест, экзамен, защита лабораторных работ, защита курсового проекта
2	Циклы и ветвления, структуры и функции	ОПК-4, ОПК-6	Тест, экзамен, защита лабораторных работ, защита курсового проекта

3	Объектно-ориентированный подход к разработке программных средств	ОПК-2, ОПК-4	Тест, экзамен, защита лабораторных работ, защита курсового проекта
4	Потоки и файлы. Многофайловые программы	ОПК-2, ОПК-4, ОПК-6	Тест, экзамен, защита лабораторных работ, защита курсового проекта
5	Шаблоны и исключения. Стандартная библиотека шаблонов.	ОПК-2, ОПК-4, ОПК-6	Тест, экзамен, защита лабораторных работ, защита курсового проекта

### **7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 45 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методике выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется с использованием выданных задач на бумажном носителе. Время решения задач 45 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методике выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется с использованием выданных задач на бумажном носителе. Время решения задач 45 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методике выставления оценки при проведении промежуточной аттестации.

Защита курсового проекта осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20 мин.

## **8. УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

## **8.1. Перечень учебной литературы, необходимой для освоения дисциплины**

Холопкина Л.В. Программирование на C++. Учебное пособие / Л.В. Холопкина. – Воронеж, ВГТУ. – 2011. – 184 с.

Холопкина Л.В. Сборник задач по дисциплине “Объектно-ориентированное программирование” / Л.В. Холопкина, Н.И. Гребенникова, М.П. Носачева. – Воронеж: ВГТУ, – 2012.

Лисицин Д.В. Объектно-ориентированное программирование [Электронный ресурс]: конспект лекций/ Д.В. Лисицин— Электрон. текстовые данные.— Новосибирск: Новосибирский государственный технический университет, 2010.— 88 с.— Режим доступа: <http://www.iprbookshop.ru/44970.html>.— ЭБС «IPRbooks».

Новиков П.В. Объектно-ориентированное программирование [Электронный ресурс]: учебно-методическое пособие к лабораторным работам/ П.В. Новиков— Электрон. текстовые данные.— Саратов: Вузовское образование, 2017.— 124 с.— Режим доступа: <http://www.iprbookshop.ru/64650.html>.— ЭБС «IPRbooks».

**8.2. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:**

ОС Linux (Ubuntu 19.10), Anjuta, g++, MS Visual Studio

## **9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА**

Для проведения лекционных занятий необходима аудитория, оснащенная проектором

Проведение лабораторных работ проводятся в специализированной лаборатории

## **10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)**



По дисциплине «Объектно-ориентированное программирование» читаются лекции, проводятся лабораторные работы, выполняется курсовой проект.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Методика выполнения курсового проекта изложена в учебно-методическом пособии. Выполнять этапы курсового проекта должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится проверкой курсового проекта, защитой курсового проекта.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.

<p>Самостоятельная работа</p>	<p>Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие:</p> <ul style="list-style-type: none"> <li>- работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций;</li> <li>- выполнение домашних заданий и расчетов;</li> <li>- работа над темами для самостоятельного изучения;</li> <li>- участие в работе студенческих научных конференций, олимпиад;</li> <li>- подготовка к промежуточной аттестации.</li> </ul>
<p>Подготовка к промежуточной аттестации</p>	<p>Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед экзаменом, экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.</p>

