

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

Кафедра «Математика и физика»



УТВЕРЖДАЮ:  
Первый проректор  
/Т.Р. Змызгова/  
«01» сентября 2023 г.

**Рабочая программа учебной дисциплины**

Объектно-ориентированное программирование

образовательной программы высшего образования –  
программы специалитета

**01.05.01 Фундаментальные математика и механика**  
**Направленность (профиль) «Математическое и программное**  
**обеспечение информационных систем»**

Формы обучения: очная

Курган 2023

Рабочая программа дисциплины «Объектно-ориентированное программирование» составлена в соответствии с учебными планами по программе специалитета «Фундаментальные математика и механика» (Математическое и программное обеспечение информационных систем), утвержденной:

- для очной формы обучения «30» 06 2023 года;

Рабочая программа дисциплины одобрена на заседании кафедры «Математика и физика» «31» 08 2023 года, протокол № 1.

Рабочую программу составил:  
к. пед. наук, доцент кафедры  
«Математика и физика»



А.В. Чернышова

Согласовано:

Заведующий кафедрой  
«Математика и физика»



М.В. Гаврильчик

Специалист по учебно-методической  
работе учебно-методического отдела



Г.В. Казанкова

Начальник управления  
образовательной деятельности



И.В. Григоренко

### 1. ОБЪЕМ ДИСЦИПЛИНЫ

Всего: 3 зачетные единицы трудоемкости (108 академических часов)

#### Очная форма обучения

| Вид учебной работы  | На всю дисциплину | Семестр      |
|---|-------------------|--------------|
|   |                   | 3            |
| <b>Аудиторные занятия (контактная работа с преподавателем), всего часов</b>             | <b>46</b>         | <b>46</b>    |
| <b>в том числе:</b>   |                   |              |
| Лекции  | 16                | 16           |
| Практические занятия  | 30                | 30           |
| <b>Самостоятельная работа, всего часов</b>  | <b>62</b>         | <b>62</b>    |
| <b>в том числе:</b>   |                   |              |
| Подготовка к зачету   | 18                | 18           |
| Другие виды самостоятельной работы (самостоятельное изучение тем (разделов) дисциплины) | 44                | 44           |
| <b>Вид промежуточной аттестации</b>   | <b>Зачет</b>      | <b>Зачет</b> |
| <b>Общая трудоемкость дисциплины и трудоемкость по семестрам, часов</b>                 | <b>108</b>        | <b>108</b>   |

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Объектно-ориентированное программирование» относится к части, формируемой участниками образовательных отношений блока Б1.

Изучение дисциплины базируется на результатах обучения, сформированных при изучении следующих дисциплин: Информатика, Языки программирования.

Знания, умения и навыки, полученные при освоении дисциплины «Объектно-ориентированное программирование», являются необходимыми для освоения последующих дисциплин: «Технологии разработки Web-приложений», «Разработка мобильных приложений», «Технологии параллельного программирования», а также при выполнении выпускной квалификационной работы в части создания программного продукта.

## 3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Целью освоения дисциплины «Объектно-ориентированное программирование» является: получение фундаментального образования, способствующего развитию личности.

Задачами дисциплины являются: изучение основных понятий объектно-ориентированного программирования, изучение основ языка программирования C++, использование этого языка для решения практических задач.

Компетенции, формируемые в результате освоения дисциплины:

- способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение (ПК-1);

- способностью выполнять работы по созданию (модификации) и сопровождению информационных систем, автоматизирующих задачи организационного управления и бизнес-процессы (ПК-3).

В результате изучения дисциплины обучающийся должен:

**Знать:** основные этапы решения задач с использованием современных программных средств и технологий; математические алгоритмы решения прикладных задач; методы решения задач, построения алгоритмов (ПК-1, ПК-3).

**Уметь:** использовать изученные инструментальные средства для решения практических задач профессиональной деятельности; использовать математические алгоритмы для решения задач в предметных областях; создавать простейшие приложения, используя язык программирования C++ (ПК-1, ПК-3).

**Владеть:** технологиями решения задач профессиональной деятельности с помощью инструментальных средств разработки приложений; навыками выбора программного обеспечения для реализации математических алгоритмов при решении практических задач; способами анализа поставленной задачи (ПК-1, ПК-3).



## 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.1. Учебно-тематический план

| Рубеж дисциплины | Номер раздела, темы | Наименование раздела, темы дисциплины                            | Количество часов по видам учебных занятий |                      |
|------------------|---------------------|--|---|----------------------|
|                  |                     |  | Лекции                                    | Практические занятия |
| <i>3 семестр</i> |                     |  |   |                      |
| Рубеж 1          | 1                   | Общие сведения о программах, лексемах и алфавите.                | 2   | -                    |
|                  | 2                   | Основные управляющие конструкции C++. Указатели и массивы в C++. | 2   | 4                    |
|                  | 3                   | Функции в C++. Строки в C++.                                     | 2   | 4                    |
|                  | 4                   | Работа с файлами в C++.  | 2   | 3                    |
|                  |                     | Рубежный контроль 1  | -   | 1                    |
| Рубеж 2          | 5                   | Структуры и объединения. Классы и объекты.                       | 2   | 4                    |
|                  | 6                   | Конструкторы и перегрузка операций.                              | 2   | 6                    |
|                  | 7                   | Наследование. Полиморфизм и виртуальные функции.                 | 2   | 4                    |
|                  | 8                   | Шаблоны.   | 2   | 3                    |
|                  |                     | Рубежный контроль 2  | -   | 1                    |
| <b>Итого:</b>    |                     |  | <b>16</b>                                 | <b>30</b>            |

### 4.2. Содержание лекционных занятий

#### Тема 1 Общие сведения о программах, лексемах и алфавите.

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.

#### Тема 2 Основные управляющие конструкции языка программирования C++. Указатели и массивы в C++.

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.

Указатели и адреса. Понятие указателя и адреса. Операция косвенной адресации и нахождения адреса. Адресная арифметика. Операция sizeof.

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

### **Тема 3 Функции в C++. Строки в C++.**

Функции. Общие сведения о функциях. Определение функции. Аргументы функции. Примеры. Передача нескольких значений в функцию. Возврат нескольких значений из функции. Примеры.

Ссылки. Общие сведения о ссылках. Ссылки в качестве параметров функций. Ссылки в качестве результатов функций. Классы памяти. Рекурсия. Функции с переменным числом параметров. Подставляемые (inline) функции. Способы передачи массивов в функции и их возврата. Примеры.

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

### **Тема 4 Работа с файлами в C++.**

Работа с файлами. Функции высокоуровневого ввода/вывода. Решение задач с использованием функций высокоуровневого ввода/вывода. Форматный ввод/вывод. Функции низкоуровневого ввода/вывода. Примеры.

### **Тема 5 Структуры и объединения. Классы и объекты.**

Структуры. Описание структур. Операции над структурами. Использование полей битов в качестве полей структур.

Объединения. Понятие объединения. Совместное использование объединений и структур. Переменные структуры.

Класс как расширение понятия структуры. Способы доступа к компонентам класса. Использование указателей для доступа к компонентам класса. Присваивание объектов. Передача объектов функциям. Объекты в качестве возвращаемого значения функций.

### **Тема 6 Конструкторы и перегрузка операций.**

Конструкторы и деструкторы. Конструкторы с параметрами. Встраиваемые функции в объявлении класса. Доступность компонентов класса. Механизм управления доступом.

Данные класса. Статические компоненты класса. Указатели на компоненты класса. Определение методов класса. Указатель this. Операторы new и delete.

Дружественные функции. Дружественные классы. Примеры.

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

### **Тема 7 Наследование. Полиморфизм и виртуальные функции.**

Общие сведения о механизме наследования. Определение производного класса. Особенности использования деструкторов при наследовании. Примеры.

Множественное наследование. Дублирование классов. Примеры.

Виртуальные базовые классы. Поведение не виртуальных компонентных функций при наследовании. Примеры.

Виртуальные функции. Статическое и динамическое связывание. Абстрактные классы. Локальные классы. Примеры.

#### Тема 8. Шаблоны.

Шаблоны классов. Шаблоны классов с шаблонами. Шаблоны функций. Обобщенные алгоритмы и функторы.

### 4.3. Практические занятия

| Номер раздела а, темы | Наименование раздела, темы                                       | Наименование практических работ  | Норматив времени, час. часы |
|-----------------------|--|--|-----------------------------|
| 2                     | Основные управляющие конструкции C++. Указатели и массивы в C++. | Лексические основы языка C++. Операции. Условные конструкции. Операторы цикла. Массивы. Одномерные массивы. Многомерные массивы. | 4                           |
| 3                     | Функции в C++. Строки в C++.                                     | Функции. Строки.   | 4                           |
| 4                     | Работа с файлами в C++.  | Файлы. Функции высокоуровневого ввода вывода   | 3                           |
|                       |  | <b>Рубежный контроль 1</b>   | <b>1</b>                    |
| 5                     | Структуры и объединения. Классы и объекты.                       | Структуры в C++. Объединения в C++. Классы и объекты   | 4                           |
| 6                     | Конструкторы и перегрузка операций.                              | Конструкторы и деструкторы. Перегрузка операций.   | 6                           |
| 7                     | Наследование. Полиморфизм и виртуальные функции.                 | Одиночное и множественное наследование. Полиморфизм и виртуальные функции.   | 4                           |
| 8                     | Шаблоны.   | Шаблоны классов. Шаблоны функций.  | 3                           |
|                       |  | <b>Рубежный контроль 1</b>   | <b>1</b>                    |
|                       |  | <b>Итого</b>   | <b>30</b>                   |

### 4.4. Контрольная работа

Не предусмотрена.



## 5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ОБУЧАЮЩИМСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Дисциплина «Объектно-ориентированное программирование» преподается в виде лекционных и практических занятий, на которых происходит объяснение, усвоение, проверка материала.

На лекционных занятиях рекомендуется использование иллюстративного материала (текстовой, графической и цифровой информации), мультимедийных форм презентаций.

В преподавании дисциплины применяются образовательные технологии: метод проблемного изложения материала; самостоятельное ознакомление обучающихся с источниками информации, использование иллюстративных материалов (видеофильмы, фотографии, компьютерные презентации), демонстрируемых на современном оборудовании, общение в интерактивном режиме.

Самостоятельная работа обучающегося, наряду с практическими занятиями в группе выполняется (при непосредственном или опосредованном контроле преподавателя) по учебникам и учебным пособиям, оригинальной современной литературе по профилю.

При прослушивании лекций рекомендуется в конспекте отмечать все важные моменты, на которых заостряет внимание преподаватель, в частности те, которые направлены на качественное выполнение соответствующих практических заданий.

Залогом качественного выполнения практических работ является самостоятельная подготовка к ним накануне путем повторения материалов лекций. Рекомендуется подготовить вопросы по неясным моментам и обсудить их с преподавателем в начале практической работы.

В качестве форм рубежного контроля используется решение практических задач.

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 практических занятиях в целях лучшего освоения материала и получения высокой оценки по результатам освоения дисциплины.

Выполнение самостоятельной работы подразумевает самостоятельное изучение разделов дисциплины, подготовку к практическим занятиям, к рубежным контролям, подготовку к зачету.

Рекомендуемая трудоемкость самостоятельной работы представлена в таблице:



**Рекомендуемый режим самостоятельной работы**

| Наименование<br>вида самостоятельной работы                                 | Рекомендуемая<br>трудоемкость,<br>акад. час. |
|---|--|
|   | Очная форма<br>обучения                      |
| <b>5 семестр</b>  |  |
| <b>Самостоятельное изучение тем дисциплины:</b>                             | <b>28</b>                                    |
| Использование графических возможностей языка C++                            | 4  |
| Изучение контейнерных классов объектно-ориентированной библиотеки Qt        | 4  |
| Механизм обработки исключений   | 5  |
| Контейнеры  | 5  |
| Многомодульные программы  | 5  |
| Стандартные библиотеки шаблонов   | 5  |
| <b>Подготовка к практическим занятиям<br/>(по 1 часу на каждое занятие)</b> | <b>15</b>                                    |
| <b>Подготовка к рубежным контролям<br/>(по 0,5 часа на каждый рубеж)</b>    | <b>1</b>                                     |
| <b>Подготовка к экзамену</b>  | <b>18</b>                                    |
| <b>Итого</b>  | <b>62</b>                                    |

## 6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

### 6.1. Перечень оценочных средств

1. Балльно-рейтинговая система контроля и оценки академической активности обучающихся в КГУ (для очной формы обучения).
2. Отчеты обучающихся по практическим работам.
3. Банк заданий к рубежным контролям № 1, № 2.
4. Вопросы к зачёту.

### 6.2. Система балльно-рейтинговой оценки работы обучающихся по дисциплине

| № | Наименование   | Содержание   |                     |   |                      |                      |       |
|---|--|--|---------------------|---|----------------------|----------------------|-------|
|   |  | Распределение баллов                               |                     |   |                      |                      |       |
|   |  | Вид учебной работы :                               | Посещение лекций    | Выполнение и защита отчетов по практическим работам | Рубежный контроль №1 | Рубежный контроль №2 | Зачёт |
| 1 | Распределение баллов за семестры по видам учебной работы, сроки сдачи учебной работы (доводятся до сведения обучающихся на первом учебном занятии) | Балльная оценка:                                   | До 16               | До 30   | До 12                | До 12                | До 30 |
|   | Примечания:  |  | 8 лекций по 2 балла | 15 практических занятий по 2 балла                  | На 6 практ. занятия  | На 15 практ. занятия |       |
| 2 | Критерий пересчета баллов в традиционную оценку по итогам работы в семестре и зачета   | 60 и менее баллов – незачёт<br>61... 100 – зачтено |                     |   |                      |                      |       |

|   |  |   |
|---|--|---|
| 3 | <p>Критерии допуска к промежуточной аттестации, возможности получения автоматического зачета (экзаменационной оценки) по дисциплине, возможность получения бонусных баллов</p> | <p>Для допуска к промежуточной аттестации по дисциплине за семестр обучающийся должен набрать по итогам текущего и рубежного не менее 51 балла. В случае если обучающийся набрал менее 51 балла, то к аттестационным испытаниям он не допускается.</p> <p>Для получения зачета без проведения процедуры промежуточной аттестации обучающемуся необходимо набрать в ходе текущего и рубежных контролей не менее 61 балла. В этом случае итог балльной оценки, получаемой обучающимся, определяется по количеству баллов, набранных им в ходе текущего и рубежного контролей. При этом, на усмотрение преподавателя, балльная оценка обучающегося может быть повышена за счет получения дополнительных баллов за академическую активность.</p> <p>Обучающийся, имеющий право на получение оценки без проведения процедуры промежуточной аттестации, может повысить ее путем сдачи аттестационного испытания. В случае получения обучающимся на аттестационном испытании 0 баллов итог балльной оценки по дисциплине не снижается.</p> <p>За академическую активность в ходе освоения дисциплины, участие в учебной, научно – исследовательской, спортивной, культурно – творческой и общественной деятельности обучающемуся могут быть начислены дополнительные баллы. Максимальное количество дополнительных баллов за академическую активность составляет 30 баллов.</p> <p>Основанием для получения дополнительных баллов являются:</p> <ul style="list-style-type: none"> <li>- выполнение дополнительных заданий по дисциплине; дополнительные баллы начисляются преподавателем;</li> <li>- участие в течении семестра в учебной, научно – исследовательской, спортивной, культурно – творческой и общественной деятельности КГУ.</li> </ul> |
|---|--|---|



|   |   |  |
|---|---|--|
| 4 | Формы и виды учебной работы для неуспевающих (восстановившихся на курсе обучения) обучающихся для получения недостающих баллов в конце семестра | <p>В случае если к промежуточной аттестации (зачету) набрана сумма менее 51 балла, обучающемуся необходимо набрать недостающее количество баллов за счет выполнения дополнительных заданий, до конца последней (зачетной) недели семестра.</p> <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 решения задач, зачет проводится в форме беседы по вопросам и решения задач.

Перед проведением каждого рубежного контроля преподаватель прорабатывает с обучающимися основной материал соответствующих разделов дисциплины в форме краткой лекции-дискуссии.

На рубежных контролях нужно решить две задачи из списка задач. Каждая задача оценивается в 6 баллов. На подготовку к ответу обучающемуся отводится время не менее 40 минут. Преподаватель оценивает в баллах полноту решения каждой задачи и заносит результат ответа в ведомость учета текущей успеваемости обучающихся.

Билет для зачета состоит из 2 вопросов и практического задания. Ответы на каждый вопрос оцениваются до 10 баллов, выполнение практического задания оценивается до 10 баллов. Время, отводимое обучающемуся на подготовку к ответу на билет, составляет 1 астрономический час.

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

### 6.4. Примеры оценочных средств для рубежных контролей и зачета

#### *Примерный список 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.
50. Переменные структуры.
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 не виртуальных компонентных функций при наследовании. Примеры.
73. Виртуальные функции. Статическое и динамическое связывание. Примеры.
74. Абстрактные классы. Примеры.
75. Локальные классы. Примеры.

### *Примеры заданий для зачета*

Пример 1. Действия с числами.

1. Найти несколько простых чисел Фибоначчи.
2. Для заданного целого числа  $m$  найти среди первых  $m*m-1$  чисел Фибоначчи хотя бы одно, делящееся на  $m$ .
3. Вычислить  $(N)!!$ , где  $(2N)!!=2*4*6*...*2N$ ,  $(2N+1)!!=1*3*5*...*(2N+1)$ .

Пример 2. Работа с массивами

1. Из заданного множества точек на плоскости выбрать две различные точки так, чтобы количества точек, лежащих по разные стороны прямой, проходящей через эти две точки, различались наименьшим образом.
2. Определить радиус и центр окружности, на которой лежит наибольшее число точек заданного на плоскости множества точек.
3. Определить радиус и центр окружности минимального радиуса, проходящей хотя бы через три различные точки заданного множества точек на плоскости.

Пример 3. Обработка символьной информации.



1. Перечислить все слова заданного предложения, которые состоят из тех же букв, что и первое слово предложения.
2. В заданном предложении найти пару слов, из которых одно является обращением другого.
3. Для каждого из слов заданного предложения указать, сколько раз оно встречается в предложении.

Пример 4. Файлы.

1. Дан файл *f*, компоненты которого являются целыми числами. Найти: 1) сумму компонент файла, 2) произведение компонент файла, 3) сумму квадратов компонент файла, 4) модуль суммы и квадрат произведения компонент файла, 5) последнюю компоненту файла.
2. Дан символьный файл *f*. Получить копию файла в файле *g*.
3. Дан символьный файл *f*. Получить в файле *g* компоненты файла *f* в обратном порядке.

Пример 5. Структуры.

1. Опишите, используя структуру, телефонную книгу.

### *Примеры заданий для рубежных контролей*

#### *Рубежный контроль 1*

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;printf("v++%8d%8d%8d\n",a,b,c);
v=k;a=v;b=v--;c=v;printf("v--%8d%8d%8d\n",a,b,c);
v=k;a=v;b=++v;c=v;printf("++v%8d%8d%8d\n",a,b,c);
v=k;a=v;b=--v;c=v;printf("--v%8d%8d%8d\n",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 "a" 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 максимальному из соответствующих значений двух других массивов.

Пример 9. 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.

Пример 10. Создать шаблон семейства функций для обмена значений двух передаваемых им параметров.

Пример 11. Создать шаблон, который определяет семейство функций, каждая из которых подсчитывает количество нулевых элементов одномерного массива параметризованного тип

#### *Примеры заданий для рубежного контроля № 2*

1. В чем преимущества и недостатки организации структур в виде стека?
2. В чем преимущества и недостатки организации структур в виде очереди?
3. Для моделирования каких реальных задач удобно использовать стек? А для каких очередь?
4. Какое значение хранит указатель на стек?
5. Какое значение хранит указатель на очередь?
6. Какие существуют ограничения на тип информационного поля стеки и очереди?
7. С какой целью в программах выполняется проверка на пустоту стека и очереди?
8. При работе со стеком или очередью доступны позиции ограниченного числа элементов. Возможна ли ситуация записи новых элементов стека



- или очереди на уже занятые собственными элементами участки памяти (запись себя поверх себя)? Ответ обоснуйте.
9. С какой целью в программах выполняется удаление стека и очереди по окончании работы с ними? Как изменится работа программы, если операцию удаления не выполнять?
  10. Создать класс Tiles (кафель), который будет содержать поля с открытым доступом: brand, size\_h, size\_w, price и метод класса getData(). В главной функции объявить пару объектов класса и внести данные в поля. Затем отобразить их, вызвав метод getData().
  11. Определить класс Children, который содержит такие поля (члены класса): закрытые — имя ребенка, фамилию и возраст, публичные — методы ввода данных и отображения их на экран. Объявить два объекта класса, внести данные и показать их.
  12. Создать иерархию классов: линия - прямоугольник - пирамида. Все классы должны содержать методы для фиксации и получения значений всех координат, а производные классы методы вычисления площади (прямоугольник), площади поверхности и объема (пирамида).
  13. Описать базовый класс СТРОКА Обязательные поля класса:
    - # Указатель на char - хранит адрес динамически выделенной памяти для размещения символов строки;
    - # Значение типа int - хранит длину строки в байтах;
    - Обязательные методы:
      - # Конструктор без параметров;
      - # Конструктор, принимающий в качестве параметра C-строку (заканчивается нулевым байтом);
      - # Конструктор, принимающий в качестве параметра символ;
      - # Конструктор копирования;
      - # Получение длины строки;
      - # Очистка строки (сделать строку пустой);
      - # Деструктор.

### 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. Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. — М. : ИД «ФОРУМ» : ИНФРА-М, 2018. — 512 с. [Электронный ресурс] / режим доступа: <http://znanium.com/bookread2.php?book=918098>.

### **7.2. Дополнительная учебная литература**

1. Программирование на С++ с погружением: практические задания и примеры кода - М.:НИЦ ИНФРА-М, 2016. - 80 с. [Электронный ресурс] / режим доступа: <http://znanium.com/bookread2.php?book=563294>.

## **8. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**

1. Методические рекомендации к выполнению лабораторных работ для студентов очной формы обучения:
2. Адаменко Ю.В. Объектно-ориентированное программирование. Методические рекомендации для студентов специальности 01.05.01. (на правах рукописи).
3. Методические рекомендации для изучения основ объектно-ориентированного программирования с использованием языка С++ [Электронный ресурс]: методические рекомендации для студентов направлений 01.03.01, 010100.62 «Математика», 050100.62 «Педагогическое образование» (профиль «Математика»), 44.03.01 «Педагогическое образование» (профиль «Информатика») / Министерство образования и науки Российской Федерации, Курганский государственный университет, Кафедра информационных технологий и методики преподавания информатики ; [сост.: Ю.В. Адаменко, А.А. Медведев]. - Электрон. текстовые дан. (тип файла: pdf ; размер: 609 Кб). - Курган : Издательство Курганского государственного университета, 2015. - 47 с.: рис., табл. - Библиогр.: с.47.
4. Медведев А.А. Основы языка программирования Си++. Методические рекомендации для проведения лабораторных работ - Курган: Изд-во Курганского гос. ун-та, 1999. – 62 с.

## **9. РЕСУРСЫ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫЕ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

1. [it.kgsu.ru](http://it.kgsu.ru) - Сайт кафедры ИТ и МПИ «Шаг за шагом»
2. <http://www.edu.ru/> - Федеральный портал «Российское образование»
3. <http://dspace.kgsu.ru/xmlui/> - Электронная библиотека КГУ
4. <http://znanium.com/> - Электронно-библиотечная система «Знаниум»

## **10. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ**

1. ЭБС «Лань»
2. ЭБС «Консультант студента»
3. ЭБС «Znanium.com»
4. «Гарант» - справочно-правовая система

## **11. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

Материально-техническое обеспечение по реализации дисциплины осуществляется в соответствии с требованиями ФГОС ВО по данной образовательной программе.

## **12. ДЛЯ СТУДЕНТОВ, ОБУЧАЮЩИХСЯ С ИСПОЛЬЗОВАНИЕМ ДИСТАНЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ**

При использовании электронного обучения и дистанционных образовательных технологий (далее ЭО и ДОТ) занятия полностью или частично проводятся в режиме он-лайн. Объем дисциплины и распределение нагрузки по видам работ соответствует п. 4.1. Распределение баллов соответствует п. 6.2 либо может быть изменено в соответствии с решением кафедры, в случае перехода на ЭО и ДОТ в процессе обучения. Решение кафедры об используемых технологиях и системе оценивания достижений обучающихся принимается с учетом мнения ведущего преподавателя и доводится до сведения обучающихся.