

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)**
Электростальский институт (филиал)
Московского политехнического университета

УТВЕРЖДАЮ

Директор

Электростальского института (филиала)

Московского политехнического

университета



/О.Д. Филиппова/

03.06.2026

Рабочая программа дисциплины

«Объектно-ориентированное программирование»

Направление подготовки

27.03.04 Управление в технических системах

ОП (образовательная программа)

«Информационные технологии в управлении»

(набор 2026-2027 года)

Квалификация (степень) выпускника

Бакалавр

Форма обучения

очная, очно-заочная

Электросталь 2026

Рабочая программа дисциплины разработана в соответствии с:

1) Приказом Министерства науки и высшего образования Российской Федерации от 31 июля 2020 г. № 871, федеральный государственный образовательный стандарт высшего образования – бакалавриат по направлению подготовки 27.03.04 Управление в технических системах.

2) Профессиональным стандартом 40.178 Специалист в области проектирования АСУ ТП, утвержденный приказом Министерства труда и социальной защиты Российской Федерации от «13» марта 2017 г. №272н.

3) Федеральный закон Российской Федерации от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации» (с изменениями и дополнениями);

4) Приказ Министерства образования и науки Российской Федерации от 05.04.2017 № 301 «Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры».

5) Учебным планом (очной, очно-заочной форм обучения) по направлению подготовки 27.03.04 Управление в технических системах.

Рабочая программа дисциплины включает в себя оценочные материалы для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине (фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины).

Автор: Т.В. Михайлова, доцент, к.т.н. кафедры ПМИИ
(указать ФИО, ученую степень, ученое звание или должность)

Программа одобрена на заседании кафедры ПМИИ (протокол № 7 от 03.06.2026 г.).

Согласовано:

Заведующий кафедрой «Прикладная математика и информатика», к.т.н.



/М.В. Лунева/

Руководитель образовательной программы
доцент, к.э.н.



/Н.В. Зиновьева/

Оглавление

1	Цель и задачи освоения дисциплины.....	4
2	Место дисциплины в структуре ООП бакалавриата.....	4
3	Структура и содержание дисциплины.....	6
3.1	Виды учебной работы и трудоемкость.....	6
3.2	Содержание разделов дисциплины.....	6
3.3	Тематика практических занятий.....	7
4	Учебно-методическое и информационное обеспечение дисциплины.....	8
4.1	Основная литература.....	8
4.2	Дополнительная литература.....	8
4.3	Электронные образовательные ресурсы.....	8
4.4	Лицензионное и свободно распространяемое программное обеспечение.....	8
4.5	Современные профессиональные базы данных и информационные справочные системы.....	9
5	Материально-техническое обеспечение дисциплины.....	9
6	Методические рекомендации.....	9
6.1	Методические рекомендации для обучающихся по освоению дисциплины.....	9
6.2	Методические рекомендации для преподавателя по организации обучения....	10
7	Фонды оценочных средств.....	12
7.1	Методы контроля и оценивания результатов обучения.....	12
7.2	Шкала и критерии оценивания результатов обучения.....	13
7.3	Оценочные средства по дисциплине.....	32

1 Цель и задачи освоения дисциплины

Целью освоения дисциплины «Объектно-ориентированное программирование» является изучение основных концепций и методов объектно-ориентированного программирования, а также изучение языка программирования C++, в котором эти концепции и методы воплощены наиболее полно.

Ставятся следующие задачи:

- 1) изучить основные принципы объектно-ориентированной парадигмы программирования, как наиболее распространенной и востребованной в настоящее время;
- 2) изучить основные возможности объектно-ориентированного языка программирования C++;
- 3) изучить основные методы программирования на языке C++;
- 4) получить навыки практического программирования на языке C++.

2 Место дисциплины в структуре ООП бакалавриата

Данная дисциплина относится к части, формируемой участниками образовательных отношений ООП. Курс «Объектно-ориентированное программирование» опирается на знания и умения, полученные в рамках дисциплины «Программирование и основы алгоритмизации».

Изучение дисциплины направлено, прежде всего, на формирование навыков создания и управления программными системами, которые войдут органичными составными частями в структуру профессиональной деятельности выпускников. Знания по данному предмету помогут формированию целостного представления студента о новых информационных технологиях, необходимых в учебной и профессиональной деятельности.

Связь с другими дисциплинами учебного плана

Перечень действующих и предшествующих дисциплин	Перечень последующих дисциплин, видов работ
Программирование и основы алгоритмизации. Вычислительные машины, системы и сети Математика.	Проектирование информационных систем. Базы данных. Разработка программных приложений. Проектирование ИС Интеллектуальные ИС

Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

В результате освоения дисциплины (модуля) у обучающихся формируются следующие компетенции и должны быть достигнуты следующие результаты обучения как этап формирования соответствующих компетенций (Таблица 1):

Таблица 1

Код и название компетенции	Индикатор достижения компетенции	Перечень планируемых результатов обучения по дисциплине
ПК-2 Способен выполнять расчет и проектирование отдельных элементов и подсистем систем управления	ПК-2.1. Формирует цели и задачи проектирования средств автоматизации ПС ПК-2.2. Использует методы патентных исследований с целью выявления оригинальных решений и аналогов в области систем управления ПК-2.3. Проводит расчеты и проектирование элементов систем управления ПК-2.4. Оформляет и составляет конструкторскую документацию на проектируемые элементы систем управления	Знать: – основные понятия и концепции объектно-ориентированной парадигмы; – понятие системы программирования; – основные элементы процедурного языка программирования, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, кассы памяти; – подпрограммы, составление библиотек программ; – объектно-ориентированную модель программирования, понятие классов и объектов, их свойств и методов. Уметь: – использовать языки программирования, строить логически правильные и эффективные программы – распознавать в конкретных прикладных (технических, экономических, социальных и т.п.) задачах математические модели из соответствующих разделов курса и проводить анализ этих моделей на основе изученных методов. Владеть: - навыками решения практических задач объектно-ориентированного программирования; - методами объектно-ориентированного программирования.
ПК-3 Способен строить модели отдельных элементов и узлов систем управления	ПК-3.1. Проводит вычислительные эксперименты с использованием программных средств для получения математических моделей, процессов и элементов автоматизации и управления ПК-3.2. Разрабатывает модели систем управления, их элементов и узлов ПК-3.3. Анализирует и верифицирует результаты моделирования элементов систем управления ПК-3.4. Определяет возможные варианты реализации отдельных блоков систем управления	Знать: -о информатике как основной естественной науке, предмете и методах информатики, и их особенностях. Уметь: - оперативно находить нужную информацию в управленческих и рекомендательных документах; - грамотно использовать информацию, найденную в управленческих и рекомендательных документах; - с позиций управленческо-правовых норм анализировать конкретные ситуации, возникающие в повседневной практике; -анализировать и оценивать организационно-управленческие решения; -принимать адекватные решения при возникновении критических, спорных ситуаций. Владеть: навыками применения организационно-управленческих решений в текущей профессиональной деятельности.

3 Структура и содержание дисциплины

3.1 Виды учебной работы и трудоемкость

Общая трудоемкость дисциплины «Объектно-ориентированное программирование» составляет 3 зачетные единицы 108 часов (из них 80 часов – самостоятельная работа студентов очно-заочной формы и 54 часа – очной формы обучения).

Разделы дисциплины очной формы изучаются в четвертом семестре: лекции – 18 часов, практические занятия – 36 часов, форма контроля – экзамен.

Разделы дисциплины очно-заочной формы изучаются в четвертом семестре: лекции – 18 часов, практические занятия – 10 часов, форма контроля – экзамен.

Структура и содержание дисциплины «Объектно-ориентированное программирование» по срокам и видам работы отражены в Приложении А.

3.2 Содержание разделов дисциплины

Тематика лекций

№ раз-дела	Основное содержание
1	Объектно-ориентированное программирование (ООП) – новая технология (парадигма) программирования. Процессно-ориентированный и объектно-ориентированный подходы к программированию. Основные свойства языка, поддерживающего ООП: абстракция, инкапсуляция, наследование, полиморфизм. Объектно-ориентированный анализ и объектно-ориентированное проектирование. Понятие объекта. Выделение используемых объектов, фиксация связей между объектами, фиксация методов обмена сообщениями между объектами
2	Объектно-ориентированный анализ и его роль в процессе создания программного обеспечения. Построение модели предметной области в терминах объектно-ориентированного программирования. Основные понятия и цели ООП. Абстрагирование объектов в ООА. Этапы создания ОО продукта.
3	Язык C++ в сравнении с языком Си. Обзор средств языка C++, реализующих механизмы ООП. Пространства имен, разрешение области видимости. Понятие класса в языке C++. Описание класса: члены-данные и члены-функции. Управление доступом к членам класса – public, private. Специальные функции – конструкторы и деструктор. Перегрузка конструкторов. Конструктор умолчания. Конструктор преобразования. Понятие ссылки на объект. Передача параметров в функции по ссылке. Возврат результата из функции по ссылке Инкапсуляция – базовый принцип ОО методологии. Наследование – базовый принцип ОО методологии. Базовый и производный классы. Наследование реализации, поведения и свойства. Переопределение метода. Типы наследования. Множественное наследование. Правила наследования различных методов. Конструкторы и деструкторы производных классов: порядок вызова, передача параметров. Полиморфизм – базовый принцип ОО методологии. Виртуальные методы. Формы полиморфизма: полиморфизм включения, параметрический полиморфизм, переопределение метода, перегрузка метода. Раннее и позднее связывание Компоновка нескольких файлов в одну программу. Этапы обработки файлов. Использование включаемых файлов. Препроцессор. Абстрактный класс. Абстрактные классы. Создание абстрактного класса. Стандартная библиотека классов C++. Поточные классы. Стандартные потоки ввода-вывода. Форматирование при вводе-выводе. Манипуляторы. Ввод-вывод объектов пользовательских классов. Оператор this . Перегрузка операций. Подписи методов и необязательные аргументы. Обработка ошибок. Обработка исключительных ситуаций. Общая схема обработки исключений: try - throw – catch. Правило выбора обработчика исключения. Передача исключения в объемлющий блок (throw;). Параметризованные функции. Параметры шаблона. Основные свойства. Параметризованные классы. Основные свойства шаблонов классов. Основные концепции STL. Последовательные и ассоциативные контейнеры. Итераторы STL. Общие свойства контейнеров. Алгоритмы. Основные этапы разработки ООП

3.3 Тематика практических занятий

№ раз-дела	План занятия, основное содержание
1	Основные понятия объектно-ориентированного анализа и программирования. Описание классов, данные и компонентные функции. Определение полей и методов класса. Формат описания и область применения
2	Объектно-ориентированный анализ. Класс «Квадрат». Класс «Массив квадратов и его сортировка». Класс «Список квадратов»
3	Основные понятия. Описание классов, данные и компонентные функции, Создание объектов и доступ к данным объекта. Определение методов класса вне класса. Статические данные класса, формат описания и область применения. Назначение конструкторов и деструкторов. Формат конструктора и деструктора, Конструкторы с параметрами и без параметров. Работа со списками. Программирование обработки списочных структур Массивы объектов. Назначение и определение, Создание динамических массивов. Указатели на объекты область применения указателей, Указатели на функции. Указатели на указатели. Ссылки на объект и область применения ссылок, отличие их от указателей. Динамическое выделение и освобождение памяти под массив объектов с помощью операторов new и delete, а также с помощью стандартных функций Область применения перегрузки операций. Перегрузка унарных операций, Перегрузка бинарных операций. Множественная перегрузка. Операции арифметического присваивания. Операции индексации массива. Преобразование типов. Преобразование объектов в основные типы и наоборот. Преобразование объектов классов в объекты других классов. Особенности перегрузки операций и их сложности Определение наследования. Базовые и производные классы. Конструкторы производных классов. Базовые функции класса. Иерархия классов. Наследование и графика. Общее и частное наследование. Уровни наследования. Множественное наследование. Неопределенность в множественном наследовании. Включение: классы в классах. Роль наследования при разработке программ Определение и формат виртуальных функций. Дружественные функции. Статические функции. Инициализация копирования и присвоения. Указатель this. Динамическая информация о типах. Полиморфизм Потоковые классы. Поточковый ввод/вывод. Указатели файлов. Файловый ввод/вывод с 1 помощью методов. Перегрузка операций извлечения и вставки Шаблоны функций. Шаблоны классов. Исключения Контейнеры. Алгоритмы. Итераторы. Специальные итераторы. Последовательные и ассоциативные контейнеры. Хранение пользовательских объектов. Функциональные объекты Основные концепции программирования. Основные этапы разработки ООП (начало, развитие, построение и передача), Принципы разработки ООП по этапам средствами UML.

Самостоятельная работа обучающегося

Кол. час	Темы, разделы, вынесенные на самостоятельную подготовку, вопросы к практическим занятиям; тематика рефератной работы, контрольных работ и др.
54/94	Самостоятельное изучение отдельных тем курса: Параметризованные классы Последовательные и ассоциативные контейнеры Реализация механизма обработки исключений Библиотечные классы исключений. Тематика заданий для реферативной работы 1 Методики объектно-ориентированного проектирования и программирования 2 Обзор ОО языков программирования 3 Классы и их свойства 4 Проектирование и создание класса на заданную тему 5 Инкапсулирование объектов

	6 Простое и множественное наследование 7 Виртуальные функции и полиморфизм 8 Абстрактные классы 9 Обработчик исключений. 10 Файловый ввод-вывод.
	Усвоение текущего учебного материала

4 Учебно-методическое и информационное обеспечение дисциплины

4.1 Основная литература

- 1 Биллиг В.А. Основы объектного программирования на С# : Учебное пособие. – М.: БИНОМ, 2012. – 582с.
- 2 Белоусова С.Н., Бессонова И.А. Основные принципы и концепции программирования на языке VBA в Excel: Учебное пособие. – М.: БИНОМ, 2010. – 200с.

4.2 Дополнительная литература

1. Варфоломеева Т. Н., Ефимова И. Ю. Лабораторный практикум по объектно-ориентированному программированию - Москва: Издательство «Флинта», 2014. – 75с.
https://biblioclub.ru/index.php?page=book_red&id=482219
2. Иванова Г.С. Технология программирования. – М.: КНОРУС, 2003.- 320с.

4.3 Электронные образовательные ресурсы

Учебно-методические материалы в электронном виде, представлены на сайте Электронная библиотека НТБ Московского политехнического университета <http://lib.mospolitech.ru>;
<http://elib-mgup.ru>

ЭБС «Университетская книга онлайн» [http:// biblioclub.ru/](http://biblioclub.ru/);

ЭБС «Лань» www.e.lanbook.com;

Национальная электронная библиотека <https://rusneb.ru/>;

Электронно-библиотечная система и образовательная платформа «Юрайт»
<https://urait.ru/>.

4.4 Лицензионное и свободно распространяемое программное обеспечение

- Операционная система Windows 7 DreamSpark № 9d0e9d49-31d1-494a-b303-612508131616
- Офисные приложения, Microsoft Office 2013 (или ниже) – Microsoft Open License. Лицензия № 61984042
- Microsoft Project 2013 Standart 32- bit/x64 Russian.
- Антивирусное ПО Avast (бесплатная версия)
- Turbo C++ (свободная лицензия)
- TurboPascal 7.1 (свободная лицензия)
- VBA 7.0 (свободная лицензия)
- Delphi 7.0 (бесплатно для образовательных целей)
- LinuxUbuntu (свободная лицензия)
- Arduino 1.6.5 (свободная лицензия)
- 1С: Предприятие 8.2 (версия для обучения)
- AnyLogic (версия пакета имитационного моделирования бесплатно для образовательных целей)
- ForexOptimizer, LiteUpdateDevelop – программное обеспечение для работы на учебном сегменте рынка Форекс (свободная лицензия)
- XAMPP (свободная лицензия)
- MySQL (свободная лицензия).

4.5 Современные профессиональные базы данных и информационные справочные системы

Каждый студент обеспечен индивидуальным неограниченным доступом к электронным библиотекам университета (elib.mgup; lib.mami.ru/lib/content/elektronyy-katalog), к электронным каталогам вузовских библиотек и крупнейших библиотек Москвы (<http://window.edu.ru>), к электронно-библиотечным системам (электронным библиотекам).

Полезные учебно-методические и информационные материалы представлены на сайтах:

<http://window.edu.ru/>

http://www.arcotel.ru/bibl/res_inet.php

<http://www.ict.edu.ru/lib/index.php>

<http://elibrary.rsl.ru/?menu=s410/elibrary/elibrary4454/science/&lang=ru>

<http://dispace.edu.nstu.ru/didesk/index>

Изучение дисциплины «Объектно-ориентированное программирование» предполагает использование мультимедийных учебных аудиторий или аудиторий, оснащенных видеопроектором и компьютером.

5 Материально-техническое обеспечение дисциплины

Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы
Учебная аудитория лекционного типа № 508, учебно-лабораторный корпус, 144000, Московская область, г. Электросталь, ул. Первомайская, д.7	Комплект мебели, переносной мультимедийный комплекс (проекционный экран, проектор, ноутбук)
Учебная аудитория для занятий семинарского типа № 303, учебно-лабораторный корпус, 144000, Московская область, г. Электросталь, ул. Первомайская, д.7	Комплект мебели, компьютеры, проектор.
Компьютерные классы № 305, 306, учебно-лабораторный корпус, 144000, Московская область, г. Электросталь, ул. Первомайская, д.7	Комплект мебели, компьютеры, проектор.

6 Методические рекомендации

6.1 Методические рекомендации для обучающихся по освоению дисциплины

Время, отводимое на самостоятельную работу должно затрачиваться студентами для изучения лекционного материала, выполнение практических задач и подготовку к лабораторным работам (при их наличии). Самостоятельная работа студентов в ходе семестра является важной составной частью учебного процесса и необходима для закрепления и углубления знаний, полученных в период сессии на лекциях, практических и интерактивных занятиях, а также для индивидуального изучения дисциплины в соответствии с программой и рекомендованной литературой.

Лекции и частично практические занятия базируются на литературных источниках, указанных в основном и дополнительном списках литературы, приведенных в рабочей программе. Более детальные и подробные рекомендации по использованию в самостоятельной работе литературных источников, а также программного обеспечения, даются на занятиях преподавателем. На этих же занятиях преподаватель передает студентам интернет-ссылки или на флэшке видеоматериалы по лабораторным работам.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты.

В конце рабочей программы есть контрольные вопросы, по которым студент имеет возможность самоконтроля выполненной работы.

В ряде дисциплин предусмотрены домашние задания, которые выполняются студентами в указанные преподавателем периоды времени (семестра). При этом студентом используются

возможности представления выполненной работы в виде реферата, презентации или эссе.

При подготовке к контрольным мероприятиям, в том числе, защите курсовых проектов (работ), экзаменам и зачетам студент пользуется конспектами лекций, примерами выполнения практических расчетов, видеоматериалами и заполненными на лабораторных работах бланками по их выполнению. Преподавателем контроль качества самостоятельной работы может осуществляться с помощью устного опроса на лекциях или практических занятиях, тестирования, проведения коллоквиума, защиты презентации, эссе или рефератов, проверки письменных контрольных работ и реферативных обзоров.

Перед контрольными мероприятиями преподаватель выдает примерные вопросы, основная доля которых представлена в рабочей программе.

Методические указания для обучающихся по организации самостоятельной работы

Самостоятельная работа обучающихся направлена на самостоятельное изучение отдельной темы учебной дисциплины. Самостоятельная работа является обязательной для каждого обучающегося, ее объем определяется учебным планом. При самостоятельной работе студент взаимодействует с рекомендованными материалами при участии преподавателя в виде консультаций. Для выполнения самостоятельной работы предусмотрено методическое обеспечение. Электронно-библиотечной система (электронная библиотека) обеспечивает возможность индивидуального доступа каждого обучающегося из любой точки, в которой имеется доступ к сети Интернет.

6.2 Методические рекомендации для преподавателя по организации обучения

1. Изучив глубоко содержание учебной дисциплины, целесообразно разработать матрицу наиболее предпочтительных методов обучения и форм самостоятельной работы студентов, адекватных видам лекционных и семинарских занятий.
2. Необходимо предусмотреть развитие форм самостоятельной работы, выводя студентов к завершению изучения учебной дисциплины на её высший уровень.
3. Организуя самостоятельную работу, необходимо постоянно обучать студентов методам такой работы.
4. Вузовская лекция - главное звено дидактического цикла обучения. Её цель – формирование у студентов ориентировочной основы для последующего усвоения материала методом самостоятельной работы. Содержание лекции должно отвечать следующим дидактическим требованиям:
 - изложение материала от простого к сложному, от известного к неизвестному;
 - логичность, четкость и ясность в изложении материала;
 - возможность проблемного изложения, дискуссии, диалога с целью активизации деятельности студентов;
 - опора смысловой части лекции на подлинные факты, события, явления, статистические данные;
 - тесная связь теоретических положений и выводов с практикой и будущей профессиональной деятельностью студентов.

Преподаватель, читающий лекционные курсы в вузе, должен знать существующие в педагогической науке и используемые на практике варианты лекций, их дидактические и воспитывающие возможности, а также их методическое место в структуре процесса обучения.

5. При изложении материала важно помнить, что почти половина информации на лекции передается через интонацию. В профессиональном общении исходить из того, что восприятие лекций студентами очно-заочной формы обучения существенно отличается по готовности и умению от восприятия студентами очной формы.

6. При проведении аттестации студентов важно всегда помнить, что систематичность, объективность, аргументированность – главные принципы, на которых основаны контроль и оценка знаний студентов. Проверка, контроль и оценка знаний студента, требуют учета его индивидуального стиля в осуществлении учебной деятельности. Знание критериев оценки знаний

обязательно для преподавателя и студента.

Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья

Обучение по дисциплине «Объектно-ориентированное программирование» инвалидов и лиц с ограниченными возможностями здоровья (далее ОВЗ) осуществляется преподавателем с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

Обучающиеся с ограниченными возможностями здоровья обеспечены электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

По дисциплине «Объектно-ориентированное программирование» обучение инвалидов и лиц с ограниченными возможностями здоровья может осуществляться как в аудитории, так и дистанционно с использованием возможностей электронной образовательной среды (образовательного портала) и электронной почты.

7 Фонды оценочных средств

7.1 Методы контроля и оценивания результатов обучения

Основаны на использовании современных достижений науки и информационных технологий. Направлены на повышение качества подготовки путём развития у студентов творческих способностей и самостоятельности (методы проблемного обучения, исследовательские методы и др.). Нацелены на активизацию творческого потенциала и самостоятельности студентов и могут реализовываться на базе инновационных структур (научных лабораторий, центров, предприятий и организаций и др.).

При проведении всех видов занятий используются активные и интерактивные методы и технологии обучения. При проведении занятий в дистанционном формате используются информационные технологии, реализуемые через сеть Интернет (ЭИОС, ZOOM-конференция и др.).

В целях реализации индивидуального подхода к обучению студентов, осуществляющих учебный процесс по собственной траектории в рамках индивидуального рабочего плана, изучение данной дисциплины базируется на следующих возможностях: обеспечение внеаудиторной работы со студентами, в том числе в электронной образовательной среде с использованием соответствующего программного оборудования, дистанционных форм обучения, возможностей интернет-ресурсов, индивидуальных консультаций и т.д.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

- _вопросы для устного опроса,
- _вопросы к экзамену,
- _контрольная работа,
- _фонд тестовых заданий.

Образцы контрольных вопросов и заданий для проведения текущего и промежуточного кон**Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы**

В результате освоения дисциплины (модуля) формируются следующие компетенции:

Код компетенции	Содержание компетенции
ПК-2	Способен выполнять расчет и проектирование отдельных элементов и подсистем систем управления
ПК-3	Способен строить модели отдельных элементов и узлов систем управления

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплин (модулей), практик в соответствии с учебным планом и календарным графиком учебного процесса.

7.2 Шкала и критерии оценивания результатов обучения

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине (Таблица 2).

Таблица 2

Показатель	Критерии оценивания			
	2	3	4	5
ПК-2 Способен выполнять расчет и проектирование отдельных элементов и подсистем систем управления				
Знать: — основные понятия и концепции объектно-ориентированной парадигмы; понятие системы программирования; — основные элементы процедурного языка программирования, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, кассы памяти; — подпрограммы, составление библиотек программ; объектно-ориентированную модель программирования, понятие классов и объектов, их свойств и методов.	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие знаний, основные понятия и концепции объектно-ориентированной парадигмы; понятие системы программирования.	Обучающийся демонстрирует неполное соответствие знаний, основные элементы процедурного языка программирования, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, кассы памяти. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие знаний, подпрограммы, составление библиотек программ; объектно-ориентированную модель программирования, понятие классов и объектов, их свойств и методов, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие необходимых знаний, подпрограммы, составление библиотек программ; объектно-ориентированную модель программирования, понятие классов и объектов, их свойств и методов, свободно оперирует приобретенными знаниями.
Уметь: — использовать языки программирования, строить логически правильные и эффективные программы — распознавать в конкретных прикладных (технических, экономических, социальных и т.п.) задачах математические модели из соответствующих разделов курса и проводить анализ этих моделей на основе изученных методов.	Обучающийся не умеет или в недостаточной степени умеет использовать языки программирования, строить логически правильные и эффективные программы.	Обучающийся демонстрирует неполное соответствие умений, использовать языки программирования, строить логически правильные и эффективные программы. Допускаются значительные ошибки, проявляется недостаточность умений, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании умениями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие умений, распознавать в конкретных прикладных (технических, экономических, социальных и т.п.) задачах математические модели из соответствующих разделов курса и проводить анализ этих моделей на основе изученных методов. Умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	Обучающийся демонстрирует полное соответствие умений, распознавать в конкретных прикладных (технических, экономических, социальных и т.п.) задачах математические модели из соответствующих разделов курса и проводить анализ этих моделей на основе изученных методов. Свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
Владеть: - навыками решения практических задач объектно-ориентированного программирования; - методами объектно-ориентированного	Обучающийся не владеет или в недостаточной степени владеет навыками решения	Обучающийся владеет навыками решения практических задач объектно-ориентированного программирования в неполном объеме, допускаются значительные ошибки, проявляется	Обучающийся частично владеет методами объектно-ориентированного программирования, навыки освоены, но допускаются	Обучающийся в полном объеме владеет методами объектно-ориентированного программирования, свободно

программирования.	практических задач объектно-ориентированного программирования.	недостаточность владения навыками по ряду показателей, Обучающийся испытывает значительные затруднения при применении навыков в новых ситуациях.	незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	применяет полученные навыки в ситуациях повышенной сложности.
ПК-3 Способен строить модели отдельных элементов и узлов систем управления				
Знать: - об информатике как основной естественной науке, предмете и методах информатики и их особенностях.	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие знаний об информатике как основной естественной науке, предмете и методах информатики и их особенностях.	Обучающийся демонстрирует неполное соответствие знаний об информатике как основной естественной науке, предмете и методах информатики, и их особенностях. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие знаний об информатике как основной естественной науке, предмете и методах информатики и их особенностях, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие необходимых знаний об информатике как основной естественной науке, предмете и методах информатики, и их особенностях, свободно оперирует приобретенными знаниями.
Уметь: - оперативно находить нужную информацию в управленческих и рекомендательных документах; - грамотно использовать информацию, найденную в управленческих и рекомендательных документах; с позиций управленческо-правовых норм анализировать конкретные ситуации, возникающие в повседневной практике; - анализировать и оценивать организационно-управленческие решения; принимать адекватные решения при возникновении критических, спорных ситуаций.	Обучающийся не умеет или в недостаточной степени умеет оперативно находить нужную информацию в управленческих и рекомендательных документах.	Обучающийся демонстрирует неполное соответствие умений, грамотно использовать информацию, найденную в управленческих и рекомендательных документах; с позиций управленческо-правовых норм анализировать конкретные ситуации, возникающие в повседневной практике. Допускаются значительные ошибки, проявляется недостаточность умений, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании умениями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие умений, анализировать и оценивать организационно-управленческие решения; принимать адекватные решения при возникновении критических, спорных ситуаций. Умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	Обучающийся демонстрирует полное соответствие умений, анализировать и оценивать организационно-управленческие решения; принимать адекватные решения при возникновении критических, спорных ситуаций. Свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
Владеть: навыками применения организационно-управленческих решений в текущей профессиональной деятельности.	Обучающийся не владеет или в недостаточной степени владеет навыками применения организационно-управленческих решений в	Обучающийся владеет навыками применения организационно-управленческих решений в текущей профессиональной деятельности в неполном объеме, допускаются значительные ошибки, проявляется недостаточность владения навыками по ряду	Обучающийся частично владеет навыками применения организационно-управленческих решений в текущей профессиональной деятельности, навыки освоены, но допускаются	Обучающийся в полном объеме владеет навыками применения организационно-управленческих решений в текущей профессиональной деятельности свободно

	текущей профессиональной деятельности.	показателей, Обучающийся испытывает значительные затруднения при применении навыков в новых ситуациях.	незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	применяет полученные навыки в ситуациях повышенной сложности.
--	--	--	---	---

Шкалы оценивания результатов промежуточной аттестации и их описание:

Форма промежуточной аттестации: экзамен

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине «Объектно-ориентированное программирование» (прошли промежуточный контроль, выполнили практические работы).

Шкала оценивания	Описание
Отлично	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.
Удовлетворительно	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
Неудовлетворительно	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

Фонды оценочных средств представлены в Приложении Б к рабочей программе.

**Структура и содержание дисциплины «Объектно-ориентированное программирование»
по направлению подготовки 27.03.04 Управление в технических системах (бакалавр)**

Очно-заочная форма обучения

n/ n	Раздел	Виды учебной работы, включающая самостоятельную работу студентов, и трудоемкость в часах					Виды самостоятельной работы студентов *						Формы аттестации	
		Л	П/С	Лаб	СРС	КСР	ДС	УО	РЗЗ	Р	К/р	Т	Э	З
	Четвёртый семестр													
1.1	Основные понятия объектно-ориентированного анализа и программирования	2	2	–	24			+			+	+		
1.2	Объектно-ориентированный анализ	2	4		26			+			+	+		
1.3	Объектно-ориентированное программирование	14	4	–	30			+			+	+		
	<i>Форма аттестации</i>							1			1	1	Э	
	Всего часов по дисциплине в четвёртом семестре	18	10		80									

Очная форма обучения

n/ n	Раздел	Виды учебной работы, включающая самостоятельную работу студентов, и трудоемкость в часах					Виды самостоятельной работы студентов *						Формы аттестации	
		Л	П/С	Лаб	СРС	КСР	ДС	УО	РЗЗ	Р	К/р	Т	Э	З
	Четвёртый семестр													
1.1	Основные понятия объектно-ориентированного анализа и программирования	4	6		12			+			+	+		
1.2	Объектно-ориентированный анализ	6	10		18			+			+	+		
1.3	Объектно-ориентированное программирование	8	20		24			+			+	+		
	<i>Форма аттестации</i>							1			1	1	Э	
	Всего часов по дисциплине в четвёртом семестре	18	36		54									

* – Сокращения форм оценочных средств см. в Приложении В к РП.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)
Электростальский институт (филиал)
Московского политехнического университета

Направление подготовки **27.03.04 Управление в технических системах**

ОП (образовательная программа) **«Информационные технологии в управлении»**

Форма обучения: **очная, очно-заочная**

Виды профессиональной деятельности:

проектно-конструкторская;

организационно-управленческая деятельность

Кафедра Прикладной математики и информатики

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО ДИСЦИПЛИНЕ

«Объектно-ориентированное программирование»

(набор 2026-2027 года)

Состав: 1) Паспорт фонда оценочных средств

2) Описание оценочных средств:

вопросы для устного опроса,

вопросы к экзамену,

контрольная работа,

фонд тестовых заданий.

Составители:

Т.В. Михайлова

Электросталь 2026

**Паспорт
фонда оценочных средств по дисциплине
«Объектно-ориентированное программирование»**

Направление подготовки
27.03.04 Управление в технических системах

ОП (образовательная программа)

«Информационные технологии в управлении»

Уровень

бакалавриат

Форма обучения

очная, очно-заочная

Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1 Основные понятия объектно-ориентированного анализа и программирования	ПК-2, ПК-3	УО, Т, К/Р
2 Объектно-ориентированный анализ	ПК-2, ПК-3	УО, Т, К/Р
3 Объектно-ориентированное программирование	ПК-2, ПК-3	УО, Т, К/Р
Промежуточная аттестация		Экзамен

Показатель уровня сформированности компетенций

Объектно-ориентированное программирование				
ФГОС ВО 27.03.04 Управление в технических системах				
КОМПЕТЕНЦИИ ИНДЕКС ФОРМУЛИРОВКА	Перечень компонентов	Технология формирования компетенций	Форма оценочного средства	Степени уровней освоения компетенций
ПК-2 Способен выполнять расчет и проектирование отдельных элементов и подсистем систем управления	Знать: – основные понятия и концепции объектно-ориентированной парадигмы; – понятие системы программирования; – основные элементы процедурного языка программирования, структуры программы, операторы и операции, управляющие структуры, структуры данных, файлы, кассы памяти; – подпрограммы, составление библиотек программ; – объектно-ориентированную модель программирования, понятие классов и объектов, их свойств и методов. Уметь: использовать языки программирования, строить логически правильные и эффективные программы – распознавать в конкретных прикладных (технических, экономических, социальных и т.п.) задачах математические модели из соответствующих разделов курса и проводить анализ этих моделей на основе изученных методов. Владеть: навыками решения практических задач объектно-ориентированного программирования; - методами объектно-ориентированного программирования.	лекции, самостоятельная работа, практические занятия	УО, К/Р, Т; Э	Базовый уровень: воспроизводство полученных знаний в ходе текущего контроля Повышенный уровень: практическое применение полученных знаний в процессе подготовки к практическим работам.
ПК-3 Способен строить модели отдельных элементов и узлов систем управления	Знать: об информатике как основной естественной науке, предмете и методах информатики, и их особенностях. Уметь: оперативно находить нужную информацию в управленческих и рекомендательных документах; - грамотно использовать информацию, найденную в управленческих и рекомендательных документах; - с позиций управленческо-правовых норм анализировать конкретные ситуации, возникающие в повседневной практике; - анализировать и оценивать организационно-управленческие решения; - принимать адекватные решения при возникновении критических, спорных ситуаций. Владеть: навыками применения организационно-управленческих решений в текущей профессиональной деятельности.	лекции, самостоятельная работа, практические занятия	УО, К/Р, Т; Э	Базовый уровень: способен осуществлять поиск, хранение, обработку и анализ информации, компьютерных и сетевых технологий. Повышенный уровень: способен осуществлять поиск, хранение, обработку и анализ информации, компьютерных и сетевых технологий в нестандартных ситуациях с их последующим анализом

Формы текущего контроля успеваемости студентов: компьютерное тестирование, защита практических работ (УО), контрольная работа.

Виды и формы аттестации: промежуточная аттестация выставляется на основании выполнения графика учебных работ и результатов контроля текущей успеваемости.

Итоговая аттестация: экзамен.

Текущий контроль успеваемости (тестирование)

формирование компетенций ПК-2, ПК-3

Тема 1 «Основные понятия ООА и программирования»

1. Модификатор `private` следует использовать для тех членов класса, которые:

- a) Желательно скрыть от общего использования
- b) Составляют внешний интерфейс класса
- c) Определяют внутреннюю реализацию данного конкретного класса
- d) Определяют некоторые характерные особенности данной ветви иерархии

2. Чисто виртуальные методы класса предоставляют потомкам класса:

- a) Детали реализации
- b) Интерфейсы
- c) Внутреннюю структуру
- d) Базовую функциональность
- e) Ничего из вышеперечисленного

3. Выберите верный вариант объявления метода `M()` класса `A` другом класса `B`:

- a) `friend void A::B();`
`class B {`
`// объявление членов класса`
`}`
- b) `class B {`
`friend void A::M();`
`// объявление членов класса`
`}`
- c) `class B : friend A::M() {`
`// объявление членов класса`
`}`
- d) `class A {`
`friend class B void M();`
`}`

4. Должна ли функция, реализующая перегруженный оператор для некоторого класса, быть дружественной этому классу?

- a) Да, всегда должна
- b) Должна в том случае, если для реализации оператора необходимо иметь доступ к скрытым членам класса
- c) Должна в том случае, если результат оператора – объект этого класса
- d) Не должна во всех случаях

5. Наследование – это:

- a) Включение в один объект экземпляра другого объекта
- b) Включение в один объект ссылки на другой объект
- c) Включение функциональности одного класса в другой
- d) Переопределение некоторых функций одного класса в другом

6. Корректен ли следующий вызов `delete`:

```
classA *ptrA = new classA[5];  
...  
delete ptrA;
```

- a) Да
- b) Нет, т.к. будет удален только один объект
- c) Нет, т.к. объекты удалять вручную нет необходимости
- d) Нет, т.к. будет удален массив, а не объекты

7. Может ли абстрактный метод не быть виртуальным?

- a) Может

- b) Может, только если он закрытый (private)
- c) Не может, любой абстрактный метод автоматически является виртуальным
- d) Не может, если не указать спецификатор virtual в объявлении абстрактного метода, то компилятор выдаст ошибку
8. Какие библиотеки входят в комплект поставки MS Visual C++ 6.0:
- a) MFC
- b) ATL
- c) OWL
- d) STL
9. В языке C++ функция может быть вызвана:
- a) перед ее объявлением или определением;
- b) только после ее объявления и определения;
- c) зависит от конкретной ситуации;
10. С точки зрения C++ правилен ли приведенный код:
- ```
int A;
int *PInt;
void *PVoid = &A;
PInt = PVoid;
```
- a) последнее присваивание вызовет ошибку;
- b) код пройдет без ошибок;
- c) зависит от настроек компилятора;
- d) это фрагмент не на языке C/C++;
11. Какими преимуществами обладает dynamic\_cast<> ?
- a) Возвращает NULL, если операция не удалась
- b) Позволяет преобразовать int\* в char\*
- c) Позволяет удалить const атрибут, т.е. преобразовать const char\* в char\*
12. Чему равна переменная x?
- ```
int p = 5;
int x = ++p++;
```
- a) 7
- b) 6
- c) 5
- d) Результат не определен
13. Какие из следующих операторов следует перегружать нестатическими методами?
- a) операторы преобразования типов
- b) + (плюс)
- c) (звездочка)
- d) = (присваивание)
- e) -> (минус больше)
- f) ->* (минус больше и звездочка)
- g) () (круглые скобки)
- h) [] (квадратные скобки)
14. Должен ли метод, объявленный как виртуальный в базовом классе, быть объявленным как виртуальный в производных классах?
- a) Не должен
- b) Должен, но разрешается не указывать ключевое слово virtual в классе, который является последним в иерархии наследования
- c) Должен, но только в непосредственном потомке
- d) Всегда должен
15. В шаблонных классах неявные типы могут использоваться в качестве:
- a) Типов обычных полей класса
- b) Типов статических полей класса
- c) Типов параметров для функций, перегружающих операторы
- d) Для объявления указателей на члены этого класса
16. После генерирования исключительной ситуации (exception), выполнение функции, которая ее сгенерировала:
- a) Продолжается
- b) Прерывается
- c) Продолжается после выполнения обработчика исключительной ситуации

d) Прерывается, если обработчик этой исключительной ситуации существует, и продолжается в противном случае

17. Способен ли компилятор MS Visual C++ генерировать:

- a) только 32-ти разрядный код;
- b) только 16-ти разрядный код;
- c) 16-ти и 32 разрядный код;
- d) зависит от поставки (Professional, Standard);

18. С точки зрения C++ правилен ли приведенный код:

```
structTypeA
{
    int I;
    char C;
    Double D;
};
structTypeA X;
TypeA Y;
```

- a) код ошибочен;
- b) это код не на C/C++;
- c) код корректен, будут созданы 2 структуры TypeA;
- d) код корректен, но будет создана только 1 структура TypeA;

19. С точки зрения C++ правилен ли приведенный код:

```
for (inti = 0; i < MAXLINES; ++i)
{
    int Length = GetLineLength (i);
    // другие операторы
}
```

- a) Это код не на C\C++;
- b) Код некорректен;
- c) Код корректен;
- d) Код корректен, на значение Length далее можно ссылаться;

20. Являются ли следующие записи корректными и идентичными?

```
class cA {
    int value;
    void method();
}
structsA {
    int value;
    void method();
}
```

- a) Корректны и идентичны
- b) cA некорректен, sA корректна
- c) cA корректен, sA некорректна
- d) Корректны, но не идентичны

21. Что обозначает такая запись?

```
class A: protected B
{
}
```

- a) Все члены класса B будут защищенными (protected) в классе A
- b) Закрытые (private) члены класса B будут защищенными (protected) в классе A, остальные не поменяют модификаторов доступа
- c) Все члены класса B, кроме закрытых (private), будут унаследованы классом A с модификатором доступа protected
- d) Класс A унаследует только защищенные (protected) члены класса B

22. Что такое "отладочная версия" программы в MS Visual C++ 6.0:

- a) такой версии в MS Visual C++ не существует;
- b) эта версия для работы со встроенным в Visual Studio отладчиком;
- c) версия, в которой встроена проверка типов и значений;
- d) это тот код, с которым работают в среде разработки;

23. С точки зрения C++ правилен ли приведенный код:

```
int a = 0;
```

```
switch (Code)
{
    case 1:
        a=1;
        break;
    case 2:
        int Count=0;
}
```

- a) компиляция вызовет ошибку;
- b) ошибка возможна при выполнении, компиляция не вызовет ошибку;
- c) это код не на C/C++;
- d) зависит от установок компилятора;

Тема 2 «Объектно-ориентированное проектирование программ»

24. С точки зрения C++ правилен ли приведенный код:

```
double A;
void mail ()
{
    int A;
    A=5;
    ::A = 2.5;
}
```

- a) код содержит синтаксическую ошибку;
- b) код правилен;
- c) это код не на C/C++;

25. Каким будет результат выполнения следующей программы?

```
#include <iostream>
#define max(a,b) ((a) > (b) ? (a) : (b))
int main()
{
    int x = 5;
    int y = 3;

    std::cout << max(++x, y) << endl;
    std::cout << max(--x, y) << endl;
    return 0;
}
```

- a) 6
- 5
- b) 5
- 4
- c) 7
- 5
- d) 6
- 4
- e) 7
- 7
- f) Результат не определен

26. Каким будет результат выполнения следующей программы?

```
#include <iostream>

int main()
{
    float a = 1 / 5;
    std::cout << "Result: " << a << std::endl;
    return 0;
}
```

- a) 0
- b) 1
- c) 5

d) 0.2

e) Результат не определен

27. Ключевое слово 'explicit' используется для того, чтобы:

a) Избежать неявного преобразования функций во встраиваемые (inline)

b) Избежать неявного преобразования типов

c) Избежать использования параметров в конструкторах по умолчанию

d) Ни одно из утверждений не верно

28. Если некий метод класса А является дружественным классу В, то в каком порядке должны быть объявлены эти классы?

a) Класс А должен быть полностью объявлен до объявления класса В

b) Класс В должен быть полностью объявлен до объявления класса А

c) До объявления класса А должен быть объявлен прототип класса В

d) До объявления класса В должен быть объявлен прототип класса А

e) Порядок объявлений значения не имеет

29. Наследуются ли закрытые (private) члены класса?

a) Наследуются, но доступны только из методов этого класса

b) Наследуются, но не доступны из методов этого класса

c) Не наследуются вообще

d) Наследуются и доступны извне

30. При каком способе передачи параметра в функцию исходный объект может изменить свое состояние?

a) По значению

b) По ссылке

c) По указателю

31. Почему конструкторы не могут быть виртуальными?

a) При вызове конструктора объекта еще не существует

b) Необходимо явно указать, какому классу принадлежит создаваемый объект

c) Конструкторы всегда вызываются неявно

d) Конструкторы не наследуются

32. После генерирования исключительной ситуации (exception), выполнение функции, которая ее сгенерировала:

a) Продолжается

b) Прерывается

c) Продолжается после выполнения обработчика исключительной ситуации

d) Прерывается, если обработчик этой исключительной ситуации существует, и продолжается в противном случае

33. Каков будет результат двукратного вызова следующего макроса ?

```
#define ABS (N) ((N)<0 ? -N : N;)
```

a) Abs (N);

b) Abs (- N);

c) Abs(- - N);

d) не определен;

34. Модификатор public следует использовать для тех членов класса, которые:

a) Составляют внешний интерфейс класса

b) Определяют внутреннюю реализацию данного конкретного класса

c) Являются абстрактными

d) Определяют некоторые характерные особенности данной ветви иерархии

35. Какой модификатор доступа менее всего подходит для абстрактных методов класса?

a) public

b) protected

c) private

36. С точки зрения C++ правилен ли приведенный код:

```
const double CD = 2.5;
```

```
double D;
```

```
D=CD;
```

```
CD = 5.0;
```

a) код ошибочен;

b) код верен;

с) зависит от установок компилятора;
 d) это код не на C/C++;

37. С точки зрения C++ правильно ли утверждение: "... если константная переменная инициализирована выражением, содержащим другие переменные, то ее нельзя использовать для определения размерности массива ...".

a) совершенно правильное утверждение;
 b) совершенно неверное утверждение;
 c) зависит от типа переменных;
 d) зависит от настроек компилятора;

38. Какие операторы можно переопределить в C++?

a) ()
 b) +
 c) ->
 d) ::
 e) *
 f) #

39. Вычислите `fn(7)`:

```
int fn(int v)
{
    if(v==1 || v==0)
        return 1;
    if(v%2==0)
        return fn(v/2) + 2;
    else
        return fn(v-1) + 3;
}
```

a) 10
 b) 11
 c) 1
 d) 0
 e) Результат не определен

Критерии оценки:

отлично – от 90% до 100% правильных ответов;
 хорошо – от 75% до 90% правильных ответов;
 удовлетворительно – от 55% до 75% правильных ответов;
 неудовлетворительно – менее 55% правильных ответов.

Примерный перечень заданий для контрольных работ формирование компетенций ПК-2, ПК-3

1. Создать инструмент построения графика функции прямой с задаваемыми коэффициентами.
2. Создать инструмент построения графика функции параболы с задаваемыми коэффициентами.
3. Создать инструмент построения графика функции гиперболы с задаваемыми коэффициентами.
4. Создать усовершенствованный ползунок (базовый компонент `TTrackBar` с вкладки `Win32` Палитры компонентов), в котором прорезь ползунка можно изобразить в виде тонкой линии. Добавить свойство, которое позволяет выводить ползунок как в привычной форме, так и с прорезью в виде тонкой линии.
5. Дана следующая иерархия объектов: родительский объект "точка" с прямым потомком "вектор", который имеет потомком "комплексное число". Поля объектов: "точка" – две вещественные координаты, потомки, наследуя эти поля, других полей не имеют. Создайте приложение, графически иллюстрирующее методы этих объектов: "точка" – нарисовать/стереть точку; "вектор" – нарисовать вектор, операции умножения вектора на число и сложения векторов; "комплексное число" – изобразить комплексное число, операции: сложение, вычитание,

умножение и деление.

6. Блуждание с поглощающими экранами. Частица движется вдоль оси OX внутри отрезка $[a; b]$ под действием случайных сил. В начальный момент времени она находится в точке x_0 . В моменты времени $t = 1, 2, 3, \dots$ она перемещается на величину Δx влево с вероятностью p , либо вправо с вероятностью $1 - p$. Если координата частицы окажется меньше a , либо больше b , то процесс завершается. Требуется:

- отразить на экране процесс блуждания точки;
- построить графики количеств находжений точки в левой и в правой половинах отрезка в зависимости от времени.

7. Блуждание точки с поглощающими экранами внутри прямоугольника. Точка движется внутри прямоугольника $[a; b] \times [c; d]$ под действием случайных сил. В начальный момент времени она находится в середине прямоугольника. В моменты времени $t = 1, 2, 3, \dots$ точка перемещается на величину Δx влево с вероятностью p_1 , либо вправо с вероятностью p_2 , либо вверх с вероятностью p_3 , либо вниз с вероятностью $1 - p_1 - p_2 - p_3$. Если абсцисса точки выходит за пределы отрезка $[a; b]$, либо ордината выходит за пределы отрезка $[c; d]$, то процесс завершается. Требуется:

- отразить на экране процесс блуждания точки;
- построить графики количеств находжений точки в четырёх четвертях прямоугольника в зависимости от времени.

8. Блуждание точки с отражающими экранами внутри прямоугольника. Условия задачи те же, что и в п.3, но точка “отражается” от границ прямоугольника, например, если ордината точки равна f и ей предстоит двигаться вниз, и $f + \Delta x > d$, то ордината точки не меняется, то есть она остаётся на месте.

9. Распространение вируса. Дан прямоугольник, разбитый на $k \times m$ ячеек. В центральную ячейку участка помещается здоровая клетка, которая начинает размножаться. Процесс размножения заключается в том, что в каждый момент времени с вероятностью q в любой из четырёх соседних (клетки, имеющие общую сторону, являются соседними) свободных ячеек появляется копия здоровой клетки. После того как количество клеток достигнет больше половины ячеек, центральная клетка заражается вирусом. В любой момент времени после заражения клетка может с вероятностью p заразить любую здоровую из четырех соседних клеток. По прошествии шести единиц времени зараженная клетка перестает болеть и становится невосприимчивой к вирусу. Иммуитет действует в течение последующих четырех единиц времени, после чего клетка становится обычной (здоровой) клеткой. Заразившись в четвёртый раз, клетка погибает, то есть освобождает ячейку. Создайте приложение, моделирующее размножение клеток и распространение вируса. В основу создания приложения положить объект “клетка” с наследником “вирус”. Основным свойством объекта-родителя “клетка” является размножение, а размножение объекта-наследника “вирус” заключается в том, что копия вируса помещается только в ту соседнюю ячейку, которая занята здоровой клеткой. Приложение должно в виде диаграммы выводить количество здоровых и больных клеток в каждый момент времени.

10. Биосистема «кролики и волки». Остров в $k \times k$ клеток заселен кроликами и волками. Кролики и волки в каждый момент времени с вероятностью $1/9$ передвигаются в одну из восьми соседних клеток (за исключением участков, ограниченных береговой линией) или просто сидят неподвижно. Каждый кролик с вероятностью $1/5$ в любую из девяти свободных соседних клеток “воспроизводит” свою копию. Если волк и кролик оказываются в одной клетке, то волк съедает кролика и получает одно очко; если волк оказывается в клетке, в которой нет кролика, то он теряет $0,1$ очка (первоначально у каждого волка на счету $m > 0$ очков). Волк с нулевой суммой погибает. Волк, сумма очков которого превышает 75% от начальной суммы m , в любую свободную из 9 соседних клеток с вероятностью $1/7$ «воспроизводит» свою копию с суммой очков m . После воспроизводства волк теряет 25% своей суммы. Если в одной клетке оказываются кроликов меньше чем волков, то кролика съедает более сильный волк, если в клетке остались еще кролики, то их съедают остальные хищники по порядку убывания суммы очков. Создайте приложение, моделирующее развитие данной биосистемы. В основу приложения положите объект “кролик”, который может передвигаться и размножаться, и его потомка “волк”, который,

наследуя от родителя возможности передвигаться и размножаться, дополнительно получает поле "сумма очков" и метод "съесть кролика". Приложение должно в виде графиков выводить количество кроликов и волков на острове в каждый момент времени.

11. Создать программу для поиска и замены словосочетаний: 1) жи, ши 2) ча, ща 3) чу, шу 4) Бер, бир 5) Пер, пир 6) Кас, кос 7) Зар, зор 8) Лаг, лож Осуществить подсчет количества замен и вывод. Вывести все строки, в которых производились замены.

12. Разбить введенный текст на слова, разместив каждое в новой строке.

13. Определить длину самого длинного и самого короткого слов и вывести их на экран (слово - его длина).

14. Построить систему классов для описания плоских геометрических фигур: круга, квадрата, прямоугольника. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и поворота на заданный угол.

15. Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность раздельного изменения составных частей адреса, создания и уничтожения объектов этого класса.

16. Составить описание класса для представления комплексных чисел с возможностью задания вещественной и мнимой частей как числами типов double, так и целыми числами. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел.

17. Составить описание класса для работы с цепными списками строк (строки произвольной длины) с операциями включения в список, удаления из списка элемента с заданным значением данного, удаления всего списка или конца списка, начиная с заданного элемента.

18. Составить описание класса для объектов-векторов, задаваемых координатами концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами.

19. оставить описание класса прямоугольников со сторонами, параллельными осям координат. Предусмотреть возможность перемещения прямоугольников на плоскости, изменения размеров, построения наименьшего прямоугольника, содержащего два заданных прямоугольника, и прямоугольника, являющегося общей частью (пересечением) двух прямоугольников.

20. Составить описание класса для определения одномерных массивов целых чисел (векторов). Предусмотреть возможность обращения к отдельному элементу массива с контролем выхода за пределы индексов, возможность задания произвольных границ индексов при создании объекта и выполнения операций поэлементного сложения и вычитания массивов с одинаковыми границами индексов, умножения и деления всех элементов массива на скаляр, печати (вывода на экран) элементов массива по индексам и всего массива.

21. Составить описание класса для определения одномерных массивов строк фиксированной длины.

Критерии оценки контрольной работы

Оценка	Критерий оценки
Отлично	полное, правильное выполнение заданий с отдельными недочётами; выполнение от 91% и более.
Хорошо	правильное выполнение заданий с незначительным количеством ошибок; выполнение более 75% менее 95 %.
Удовлетворительно	выполнение основной части заданий с ошибками; выполнение более 50% менее 75 %.
Неудовлетворительно	частичное выполнение заданий (менее половины); допущение значительного количества ошибок; выполнение менее 50%.

Контрольные вопросы для проведения текущего контроля (УО) формирования компетенций ПК-2, ПК-36

1 Основные идеи и компоненты .Net технологии.

2. Последовательность создания и выполнения программ на платформе .Net.

3. Сборка (assembly) и ее структура в .Net технологии. Основные части и их назначение.
4. Структура программы на языке C#.
5. Описания классов на языке C#.
6. Встроенные типы языка C#. Неявное и явное приведение (преобразование) встроенных типов.
7. Описание и использование методов класса. Перегрузка методов класса.
8. Базовый класс System.Object. Назначение, методы.
9. Операции языка C#. Перегрузка операций в создаваемых классах.
10. Программирование явного и неявного преобразования пользовательских типов (собственных классов).

Критерии оценки устного опроса (собеседования)

Оценка «отлично» выставляется студенту, если студент ориентируется в теоретическом материале; имеет представление об основных подходах к излагаемому материалу; знает определения основных теоретических понятий излагаемой темы, умеет применять теоретические сведения для анализа практического материала, в основном демонстрирует готовность применять теоретические знания в практической деятельности и освоение большинства показателей формируемых компетенций.

Оценка «хорошо» выставляется студенту, если студент ориентируется в теоретическом материале; имеет представление об основных подходах к излагаемому материалу, но затрудняется в ответах на некоторые вопросы; знает определения основных теоретических понятий излагаемой темы, но не в полной мере отражает суть рассматриваемой проблемы, в основном умеет применять теоретические сведения для анализа практического материала, в основном демонстрирует готовность применять теоретические знания в практической деятельности и освоение большинства показателей формируемых компетенций.

Оценка «удовлетворительно» выставляется студенту, если показаны недостаточные знания теоретического материала, основных понятий излагаемой темы, не всегда с правильным и необходимым применением специальных терминов, понятий и категорий; анализ практического материала был нечёткий.

Оценка «неудовлетворительно» выставляется в случаях, когда не выполнены условия, позволяющие выставить оценку «удовлетворительно».

Промежуточная аттестация (экзамен)

формирование компетенций ПК-2, ПК-3

1. Объектно-ориентированный анализ и его роль в процессе создания программного обеспечения. Построение модели предметной области в терминах объектно-ориентированного программирования.
2. Основные понятия и цели ООП. Абстрагирование объектов в ООА. Этапы создания ОО продукта: анализ-проектирование- эволюция-модификация. Атрибуты, их типы и представление при ОМ.
3. Понятия: объектов, классов, инкапсуляции, наследования, полиморфизма. Пример.
4. Понятие класса. Полное описание базового класса. Пример
5. Понятие класса. Дружественные классы. Пример
6. Конструкторы, деструкторы и другие методы класса. Пример.
7. Скрытие информации. Инкапсуляция. Пример.
8. Понятие класса. Простое наследование. Пример.
9. Понятие класса. Множественное наследование. Пример.
10. Виртуальные функции и полиморфизм. Пример.
11. Понятие класса. Абстрактные и конкретные классы. Пример.
12. Конструкторы и деструкторы производного класса. Пример.
13. Область видимости классов. Пример.
14. Компоновка нескольких файлов в одну программу. Пример.
15. Использование включаемых файлов. Препроцессор. Пример.
16. Оператор this. Перегрузка операторов. Пример.

17. Понятие класса. Шаблоны класса. Пример.
18. Обработка исключительных ситуаций. Пример.
19. Создание базы данных. Компоненты доступа к данным. Пример.
20. Выбор информации из базы данных. SQL – запрос. Пример.

7.3 Оценочные средства по дисциплине

«Объектно-ориентированное программирование»

№ ОС	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в ФОС
1	Контрольная работа (К/Р)	Средство проверки умений применять полученные знания для решения задач определенного типа по теме или разделу.	Комплект контрольных заданий по вариантам
2	Устный опрос собеседование (УО)	Средство контроля, организованное как специальная беседа педагогического работника с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п.	Вопросы по темам/разделам дисциплины
3	Тест (Т)	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Фонд тестовых заданий
4	Экзамен (Э)	Итоговая форма оценки знаний. В высших учебных заведениях проводится во время сессии.	Вопросы к экзамену