

МИНПРОСВЕЩЕНИЯ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования  
"Тульский государственный педагогический университет им. Л.Н. Толстого"  
(ФГБОУ ВО "ТГПУ им. Л.Н. Толстого")

## Объектно-ориентированное программирование

### рабочая программа дисциплины (модуля)

Закреплена за кафедрой	<b>институт передовых информационных технологий</b>
ОПОП	<b>Направление 02.03.02 Фундаментальная информатика и информационные технологии направленность (профиль) Инженерия программного обеспечения</b>
Квалификация	<b>Бакалавр</b>
Год начала подготовки	<b>2022</b>
Форма обучения	<b>очная</b>
Общая трудоемкость	<b>5 з.е.</b>

Виды контроля по семестрам:  
экзамен 2

Семестр(Курс.Номер семестра на курсе)	2(1.2)		Итого	
	УП	РПД	УП	РПД
Лекции	34	34	34	34
Лабораторные	34	34	34	34
Итого ауд.	68	68	68	68
КСР	4	4	4	4
Контактная работа	72	72	72	72
Сам. работа	72	72	72	72
Часы на контроль	36	36	36	36
Практическая подготовка	0	0	0	0
Семинары	0	0	0	0
Консультации	0	0	0	0
Итого трудоемкость в часах	180	180	180	180

Программу составил(и):

*ст. преподаватель, Гладких Илья Юрьевич*

Рабочая программа дисциплины

**Объектно-ориентированное программирование**

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии (приказ Минобрнауки России от 23.08.2017 г. № 808)

составлена на основании учебного плана:

Направление 02.03.02 Фундаментальная информатика и информационные технологии  
направленность (профиль) Инженерия программного обеспечения  
утвержденного Учёным советом вуза от 28.02.2022 протокол № 3.

РПД утверждена Учёным советом университета  
протокол от 28.2.2022 г. № 3

## 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

изучение и практическое освоение общих принципов и современных методов объектно-ориентированного программирования, получение навыков использования конкретных инструментов объектно-ориентированного программирования и средств языков программирования C# и Java.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Цикл (раздел) ООП:	Б1.О
<b>2.1</b>	<b>Требования к предварительной подготовке обучающегося:</b>
1.	Для изучения объектно-ориентированного программирования на достаточном уровне должны быть освоены дисциплины "Программирование", "Алгоритмы и структуры данных" и "Основы проектной деятельности"
<b>2.2</b>	<b>Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:</b>
1.	Параллельное программирование
2.	Разработка кроссплатформенного программного обеспечения
3.	Веб-программирование
4.	Операционные системы
5.	Технологии параллельного программирования
6.	Технологии виртуальной реальности
7.	Машинное обучение
8.	Технология IoT

## 3. СООТНЕСЕНИЕ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ) С ИНДИКАТОРАМИ ДОСТИЖЕНИЯ КОМПЕТЕНЦИЙ

### 3.1 Компетенции обучающегося и индикаторы их достижения:

ОПК-2: Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности	
ОПК-2.1	Знает основные положения и концепции в области программирования, архитектуру языков программирования, теории коммуникации, знает основную терминологию, знаком с содержанием Единого Реестра Российских программ
	правила использования документации; основные принципы объектно-ориентированного программирования; принципы построения классов; методы программирования в парадигме объектно-ориентированного программирования; способы представления числовой, текстовой и структурированной информации в парадигме объектно-ориентированного программирования; методы обработки числовой, текстовой и структурированной информации в парадигме объектно-ориентированного программирования.
ОПК-2.2	Умеет анализировать типовые языки программирования, составлять программы
	создавать программную документацию для разработанного программного продукта; вести разработку согласно этапам и стадиям разработки программных продуктов; реализовывать алгоритмы средствами языка программирования объектно-ориентированной парадигмы; использовать современные методы объектно-ориентированного программирования при кодировании программных систем разного уровня сложности.
ОПК-2.3	Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения, анализа типов коммуникаций
	базовыми принципами объектно-ориентированного моделирования систем и принципы проектирования сложных систем; технологией объектно-ориентированного моделирования; технологией объектно-ориентированной разработки программного обеспечения.

### 3.2 Результаты обучения по дисциплине:

**В результате освоения дисциплины обучающийся должен:**

	<b>Знать:</b>
3.1	правила использования документации;
3.2	основные принципы объектно-ориентированного программирования;
3.3	принципы построения классов;
3.4	методы программирования в парадигме объектно-ориентированного программирования;
3.5	способы представления числовой, текстовой и структурированной информации в парадигме объектно-ориентированного программирования;

3.6	методы обработки числовой, текстовой и структурированной информации в парадигме объектно-ориентированного программирования.
	<b>Уметь:</b>
У.1	создавать программную документацию для разработанного программного продукта;
У.2	вести разработку согласно этапам и стадиям разработки программных продуктов;
У.3	реализовывать алгоритмы средствами языка программирования объектно-ориентированной парадигмы;
У.4	использовать современные методы объектно-ориентированного программирования при кодировании программных систем разного уровня сложности.
	<b>Владеть:</b>
В.1	базовыми принципами объектно-ориентированного моделирования систем и принципы проектирования сложных систем;
В.2	технологией объектно-ориентированного моделирования;
В.3	технологией объектно-ориентированной разработки программного обеспечения.

#### 4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Литература	Содержание
	<b>Объектно-ориентированное визуальное программирование</b>				
1.1	Основные принципы объектно-ориентированного программирования /Лек/	2	2	Л1.1 Л1.2 Л1.3Л2.3	Эволюция методологий программирования. Начало начал, или первое поколение языков программирования. Развитие алгоритмических абстракций, или второе поколение языков программирования. Модуль как единица построения программных систем, третье поколение языков программирования. Зарождение объектной модели, четвертое поколение языков программирования. Объектные языки программирования, объектно-ориентированные языки программирования, объектно-ориентированный анализ, дизайн и проектирование. Парадигмы программирования. Составные части объектного подхода. Абстрагирование. Инкапсуляция. Модульность. Иерархия.
1.2	Объектноориентированная модель /Лек/	2	2	Л1.1 Л1.2 Л1.3Л2.3	Понятие объекта.Свойства, присущие объектам. Состояние. Поведение. Идентичность.Отношения между объектами. Типы отношений. Связь (ассоциация). Агрегация.
1.3	Основные алгоритмические отличия C# от C /Лек/	2	2	Л1.1 Л1.2 Л1.3Л2.3	Использование ссылок. Передача аргументов функции по ссылке. Использование констант. Логические тип и перечисления. Операторы управления динамической памятью, инициализация массивов. Структура программы, отдельная компиляция и особенности использования статической памяти. Пространства имен и исключения (краткий обзор). Библиотека ввода вывода (краткий обзор iostream). Функциональный полиморфизм

1.4	Средства объектного программирования языка C# /Лек/	2	6	Л1.1 Л1.2 Л1.3Л2.3	<p>Представление объектов и классов. Реализация поведения объектов на примере добавления функций—членов в структуры. Структура как вырожденный класс. Структура объявления класса. Доступ к членам класса. Поля данных класса как механизм реализации состояния объекта. Функции члены класса как механизм реализации поведения объекта. Спецификаторы доступа для обеспечения инкапсуляции. Средства управления жизнью объекта. Конструкторы и деструкторы. Конструирование и уничтожение объектов и массивов объектов. Особенности использования конструктора копии, конструктора по умолчанию, оператора присваивания. Описание селекторов и модификаторов. Перегрузка операторов C++ как реализация поведения с предопределенным смыслом. Дружественность как механизм нарушения инкапсуляции. Достоинства и недостатки механизма дружественности. Статические поля и методы классов. Инициализация статических полей. Реализация отношений между объектами и классами. Ассоциация и агрегация объектов и классов. Зависимость по времени жизни. Использование и зависимость от интерфейсов. Объекты при передаче параметров и возврате из методов. Варианты реализации отношения клиент-сервер. Внутренние классы.</p>
1.5	Средства объектно-ориентированного программирования C# /Лек/	2	4	Л1.1 Л1.2 Л1.3Л2.3	<p>Наследование как средство организации иерархий классов. Принцип замещения Лисковской. Одиночное наследование. Понятие производного класса. Управление доступом в производных классах. Конструкторы и деструкторы, совмещение имен методов при наследовании, иерархии. Абстрактные классы и виртуальные функции. Виртуальный полиморфизм. Информация о типе на этапе выполнения. RTTI. Множественное наследование. Проблема множественного наследования. Виртуальное наследование как средство разрешения коллизий. Порядок вызовов конструкторов и деструкторов при множественном наследовании. Чистые виртуальные классы, понятие интерфейса. Принципы дизайна иерархий классов. OCP, DIP, ISP. Пространства имен. Пространства имен как средство реализации модульности. Поиск имен и разрешение конфликтов. Объединение пространств имен. Принципы дизайна пакетов. Обработка исключений. Обработка ошибок. Группировка и перехват исключений. Управление ресурсами. Исключения и эффективность. Альтернативные методы обработки ошибок. Стандартные исключения.</p>
1.6	Обобщенное программирование /Лек/	2	4	Л1.1 Л1.2 Л1.3Л2.3	<p>Шаблоны классов. Определение шаблона. Инстанцирование. Параметры шаблонов и проверка типов. Шаблоны функций. Специализация. Наследование и шаблоны</p>

1.7	Стандартная библиотека C# /Лек/	2	4	Л1.1 Л1.2 Л1.3Л2.3	Библиотека стандартных шаблонов. Общие сведения (понятия контейнеров, итераторов и объектов функций). Контейнеры (виды контейнеров, последовательные и ассоциативные контейнеры, адаптеры). Итераторы (итератор как обобщение указателя, классы итераторов). Алгоритмы (примеры алгоритмов с использованием итераторов: алгоритмы сортировки, алгоритмы, не изменяющие содержание контейнера, алгоритмы, изменяющие содержание контейнера). Библиотека ввода-вывода. Поток вывода. Вывод типов определяемых пользователем. Поток ввода. Ввод типов определяемых пользователем. Форматирование в потоках ввода-вывода. Буферизация.
1.8	Теоретические основы разработки классов в среде объектного программирования /Ср/	2	6	Л1.1 Л1.2 Л1.3Л2.3	Объектно-ориентированное и процедурное программирование. Действие и объект. Программы, сообщения, объекты. Объекты, свойства, методы, события. Классы и наследование, инкапсуляция, полиморфизм. Конструкторы и деструкторы. Основы программирования в среде визуального интерфейса. Интегрированные среды C# и Java. Общий вид окон сред. Главное меню. Различные способы задания свойств (значение, список, диалог). Окно дизайнера. Окно редактора кода. Палитра компонентов. Быстрые кнопки. Управление конфигурациями окон среды. Структура кода модуля и пакета.
1.9	Проекты C# /Ср/	2	6	Л1.1 Л1.2 Л1.3Л2.3	Виды проектов в C#. Организация проекта в C#, основные файлы проектов. Основные проектные операции. Создание и сохранение проекта. Менеджер проекта. Оформление завершенного проекта. Включение в проект новых приложения, окна или страницы. Создание отдельной копии окна. Просмотр приложения и модулей без включения их в проект. Размещение компонентов в приложении. Работа с группой компонентов. Инструментальные средства поддержки разработки кода. Применение Code Insight—Знатока Кода. Исследователь кода Code Explorer. Просмотр иерархии классов, модулей и глобальных символов — Object Browser. Получение информации о классах, свойствах, методах из окна Редактора Кода. Навигация в коде, закладки и дополнительные окна редактирования. Отладка приложений. Настройка инструментальной панели. Настройка палитры компонентов. Настройка Редактора Кода. Настройка Code Insight. Настройка Code Explorer и Object Browser. Настройка отладчика.

1.10	Компоненты WPF и C# /Ср/	2	6	Л1.1 Л1.2 Л1.3Л2.3	<p>Организация библиотеки компонентов. Страницы библиотеки компонентов. Программный доступ к свойствам и методам объектов. Настройка палитры компонентов. Создание и запись в библиотеку шаблонов компонентов и групп компонентов. Компоненты ввода и отображения текстовой, цифровой и иерархической информации. Компоненты выбора из списков. Таблица строк—компонент StringGrid. Ввод и отображение чисел, дат и времени. Секционированное отображение текстов. Компоненты отображения графической информации. Компонент Shape. Построение графиков и диаграмм. Мультимедиа и анимация. Универсальный проигрыватель Media Player Воспроизведение видеоклипов. Кнопки, индикаторы, управляющие элементы. Компоненты — меню. Главное меню — компонент MainMenu. Контекстное всплывающее меню - компонент PopupMenu. Панели и компоненты внешнего оформления. Инструментальные панели. Перестраиваемые панели. Системные диалоги. Диалоги открытия и сохранения файлов. Диалог выбора шрифта. Диалог выбора цвета. Диалоги печати и установки принтера. Диалоги поиска и замены текста. Компоненты организации управления приложением.</p>
1.11	Язык C# /Ср/	2	6	Л1.1 Л1.2 Л1.3Л2.3	<p>Синтаксис языка. Компилятор. Файлы проекта C#. Области видимости и время жизни. Константы, переменные, типизированные константы. Переменные. Процедуры и функции. Операции. Операторы. Условные операторы выбора if. Условный оператор множественного выбора case . Операторы цикла for, repeat, while. Прерывание цикла: оператор break, процедуры continue, exit и abort. Исключения. Типы данных в языке C#. Классы. Свойства. Методы и их наследование, полиморфизм. Конструкторы и деструкторы. События. Процедуры и функции C#. Строка описания формата и функция Format. Математические преобразования дат и времени. Процедуры и функции файлового ввода/вывода и управления файлами. Процедуры и функции вызова диалоговых окон, воспроизведения звуков.</p>
1.12	Введение в Java. Виртуальная машина Java. /Лек/	2	4	Л1.1 Л1.2 Л1.3Л2.1 Л2.2	<p>Знакомство с Java. Переменные и константы, поля объектов и классов. Область видимости. Примитивные типы данных. Ссылочные типы данных. Выражения и операторы. Преобразование типов. Оболочечные классы. Класс Math, его методы и константы. Работа с датой и временем в Java Строки в Java. Работа со строками, преобразование типов. Объекты и классы в Java. Права доступа. Наследование. Ключевое слово super. Полиморфизм в Java. Виртуальная машина Java. Байт-код. Принципы работы Java приложений.</p>

1.13	Язык Java /Ср/	2	6	Л1.1 Л1.2 Л1.3Л2.1 Л2.2	Синтаксис языка. Компилятор. Файлы проекта Java. Области видимости и время жизни. Константы, переменные, типизированные константы. Переменные. Процедуры и функции. Операции. Операторы. Условные операторы выбора. Условный оператор множественного выбора. Операторы цикла. Прерывание цикла/ Исключения. Типы данных в языке Java. Классы. Свойства. Методы и их наследование, полиморфизм. Конструкторы и деструкторы. События. Процедуры и функции Java. Процедуры и функции файлового ввода/вывода и управления файлами. Процедуры и функции вызова диалоговых окон, воспроизведения звуков.
1.14	Многопоточная модель Multi-threading programming. /Лек/	2	4	Л1.1 Л1.2 Л1.3Л2.3 Л2.4	Понятие потока (thread) и многопоточное выполнение (multi-threading); модели многопоточности; пользовательские потоки и потоки ядра.
1.15	Многопоточная модель Multi-threading programming. /Ср/	2	8	Л1.1 Л1.2 Л1.3Л2.3	Понятие потока (thread) и многопоточное выполнение (multi-threading); модели многопоточности; пользовательские потоки и потоки ядра
1.16	Безопасность приложений. Защита данных /Лек/	2	2	Л1.1 Л1.2 Л1.3Л2.3	Модель безопасности. Принципы организации и эволюция модели безопасности. Security Manager. Инициализация и функции. Права доступа. Управление и проверка прав доступа. Исключительные ситуации. Алгоритмы шифрования. Ключи и цифровые подписи
1.17	Объектно-ориентированное визуальное программирование /КСР/	2	2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Л2.3 Л2.4	
	<b>Разработка приложений для Windows</b>				
2.1	Основы создания WPF-приложений с использованием XAML /Лаб/	2	4	Л1.1 Л1.2 Л1.3Л2.3	Создание простейшего WPF приложения
2.2	Создание классов для WPF-приложений /Лаб/	2	2	Л1.1 Л1.2 Л1.3Л2.3	Понятие классов в объектно-ориентированном программировании. Создание собственного класса средствами MS Visual Studio. Создание приложения для работы с треугольником
2.3	Управление расположением элементов WPF-приложения /Лаб/	2	2	Л1.1 Л1.2 Л1.3Л2.3	Управление расположением элементов WPF-приложения
2.4	Создание пользовательского компонента WPF /Лаб/	2	2	Л1.1 Л1.2 Л1.3Л2.3	Создание тестового приложения
2.5	Работа с окнами приложения /Лаб/	2	2	Л1.1 Л1.2 Л1.3Л2.3	Создание дочерних окон в приложении WPF. Манипуляции с данными и объектами из разных окон. Чтение данных из текстового файла. Чтение из файла текстовой информации.
2.6	Разработка WPF-приложения пользовательским интерфейсом и модульной организацией. /Лаб/	2	4	Л1.1 Л1.2 Л1.3Л2.3	Создание WPF-приложения. Создание класса ядра приложения. Обработчики событий.



2.7	Работа с графической составляющей в WPF-приложениях /Лаб/	2	2	Л1.1 Л1.2 Л1.3Л2.3	Проект «Редактор градиентов». Файл, содержащий набор системных цветов. Создание C# кода приложения «Редактор градиентов».
2.8	Стили, триггеры и анимация в WPF-приложениях /Лаб/	2	4	Л1.1 Л1.2 Л1.3Л2.3	Использование Стилией и триггерах в WPF-приложениях с определением в XAML и CS составляющих.
2.9	Основы привязки данных в WPF-приложениях /Лаб/	2	4	Л1.1 Л1.2 Л1.3Л2.3	Привязка к свойству элемента. Ошибки привязки. Создание привязки в коде. Множественные привязки. Направление привязки. Обновления привязки. Привязка объектов, не являющихся элементами
2.10	Разработка технического задания на программу /Лаб/	2	4	Л1.1 Л1.2 Л1.3Л2.3	Программная документация. Техническое задание: порядок разработки и основные разделы.
2.11	Требования к интерфейсу пользователя приложений для Windows /Ср/	2	8	Л1.1 Л1.2 Л1.3Л2.3	Общие рекомендации по разработке графического интерфейса. Многооконные приложения. Меню. Методика проектирования меню и инструментальной панели. Компоновка. Печать из приложения текстов и изображений. Проектирование окон с изменяемыми размерами. Масштабирование компонентов. Обработка событий клавиатуры и мыши. Перетаскивание объектов. Буксировка компонентов в окне приложения. Управление окнами и страницами. Графика и мультимедиа Построение графических изображений. Установка и настройка приложения: работа с системным реестром. Автономные приложения и пакеты.
2.12	Моделирование реальных природных процессов /Ср/	2	8	Л1.1 Л1.2 Л1.3Л2.3	Молекула газа в закрытом сосуде Броуновское движение. «Равновесие» (второе начало термодинамики). «Равновесие» (второе начало термодинамики). Графические эффекты. Моделирование спрайтов. Организация скроллинга изображения, реализация эффектов затухания, составления изображения из отдельных точек, красивой смены фонового изображения.
2.13	Введение в Java. Виртуальная машина Java. /Лаб/	2	4	Л1.1 Л1.2 Л1.3Л2.1 Л2.2	Консольный ввод вывод в Java. Работа со строками в Java. Преобразование типов. Работа с файлами в Java. Классы в Java. Создание класса и объектов этого класса.
2.14	Язык программирования Java. Виртуальная машина Java. /Ср/	2	6	Л1.1 Л1.2 Л1.3Л2.1 Л2.2	Консольный ввод вывод в Java. Работа со строками в Java. Преобразование типов. Работа с файлами в Java. Классы в Java. Создание класса и объектов этого класса.
2.15	Работа над индивидуальным проектом /Ср/	2	6	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Л2.3 Л2.4	
2.16	Разработка приложений для Windows /КСР/	2	2	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Л2.3 Л2.4	
2.17	Подготовка к экзамену /Ср/	2	6	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Л2.3 Л2.4	Вопросы к экзамену

## 5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

### 5.1. Типовые задания для проведения текущего контроля

Типовое тестовое задание

Вопрос 1

Чем не может являться операция: << ?

Выберите один ответ:

- a. меньше или равно;
- b. сдвиг влево;
- c. операция вывода;

Вопрос 2

Как называется выделение существенных характеристик объекта, отличающих его от других видов объектов?

Выберите один ответ:

- a. абстракция;
- b. полиморфизм;
- c. инкапсуляция;

Вопрос 3

Какие из записей соответствуют правильным обозначениям отрицательных констант?

1.- 0x3C

2.- 022

3.- 4.8

4.- 0.1e4

Выберите один ответ:

- a. нет верных утверждений;
- b. верно 1 и 3;
- c. верно 2 и 3;
- d. верно 1 и 2;
- e. все верно;

Вопрос 4

Что обозначает данный код?

```
int someFunc(double, int);
int main(int argc, char* pArgs[])
{
    someFunc(1, 2);
}
int someFunc(double arg1, int arg2)
{ //...выполнение каких-то действий }
```

Выберите один ответ:

- a. создание итератора с указателем на;
- b. определение функции;
- c. обнуление;

Вопрос 5

Как называется язык, если в нем реализованы классы, но не реализовано наследование?

Выберите один ответ:

- a. функциональный;
- b. объектно-ориентированный;
- c. объектный;

Вопрос 6

Что подразумевается под проектированием программного продукта?

Выберите один ответ:

- a. программирование;
- b. определение структуры и общих принципов;
- c. постановка задачи;

## Вопрос 7

Какое свойство объекта называется инвариантом?

Выберите один ответ:

- a. проверяющее состояние объекта;
- b. полностью определяющее состояние объекта;
- c. позволяющее изменить состояние объекта;

## Вопрос 8

Сколько вызовов системных функций всегда будет в программе, независимо от ее реализации?

Выберите один ответ:

- a. по крайней мере один;
- b. не менее двух;
- c. всегда 2;

## Вопрос 9

Что должно стать результатом проектирования создаваемой системы?

Выберите один или несколько ответов:

- a. прототипы системы;
- b. выработка общих тактических приемов;
- c. описание назначения системы;
- d. описание архитектуры;

## Вопрос 10

Что должно стать результатом анализа создаваемой системы?

Выберите один ответ:

- a. прототипы системы;
- b. выработка общих тактических приемов;
- c. описание архитектуры;
- d. описание назначения системы;

## Вопрос 11

Функции, которые вызывают сами себя, называются

Выберите один ответ:

- a. рекурсивными;
- b. динамическими;
- c. априорными;

## Вопрос 12

Что должно стать результатом концептуализации создаваемой системы?

Выберите один ответ:

- a. описание назначения системы;
- b. прототипы системы;
- c. описание архитектуры;
- d. выработка общих тактических приемов;

## Вопрос 13

Что не верно по отношению к тестированию?

Выберите один ответ:

- a. средства тестирования желательно прямо встраивать в систему;
- b. следует начинать только после полного создания программы;
- c. следует начинать, как только появляется работающая система;

## Вопрос 14

Укажите шаги для улучшения читаемости программы?

Выберите один или несколько ответов:

- a. создание корректных комментариев;
- b. уход от вложенных конструкций;
- c. использование осмысленных имен переменных;

Вопрос 15

Как называется процесс отделения друг от друга элементов объекта, определяющих его устройство и поведение?

Выберите один ответ:

- a. инкапсуляция;
- b. полиморфизм;
- c. абстракция;

Вопрос 16

Что называется иерархичностью?

Выберите один ответ:

- a. ограничение доступа ко внутренней структуре объекта;
- b. возможность наследования;
- c. возможность описания классов в отдельных модулях;

Вопрос 17

Как можно определить бинарную операцию?

Выберите один ответ:

- a. как функцию, принимающую два параметра;
- b. как функцию, принимающую один параметр;
- c. обоими способами;

Вопрос 18

Неверно высказывание:

Выберите один ответ:

- a. Контекст пространства имен может содержать другие контексты;
- b. Контекст пространства имен может содержать объявления и определения переменных;
- c. Контекст пространства имен может содержать только объявления переменных;

Вопрос 19

Как можно определить унарную операцию?

Выберите один ответ:

- a. только как функцию, принимающую один параметр;
- b. как функцию, без параметров или как функцию, принимающую один параметр;
- c. только как функцию, без параметров;

Вопрос 20

Что называется инкапсуляцией?

Выберите один ответ:

- a. возможность описания классов в отдельных модулях;
- b. ограничение доступа ко внутренней структуре объекта;
- c. возможность наследования;Пока нет ответа

## 5.2. Типовые задания для проведения промежуточной аттестации

### ВОПРОСЫ К ЭКЗАМЕНУ

1. Описание класса. Экземпляр класса.
2. Реализация класса в ООП.
3. Поля, методы, свойства класса.
4. Объектно-ориентированная модель системы.
5. Наследование инкапсуляция, полиморфизм.

6.	Перечислите и охарактеризуйте основные принципы ООП.
7.	Основные структурные части ИСР C#.
8.	Основные проектные операции C#.
9.	Назначение основных элементов кода C#.
10.	Основные структурные части ИСР Java.
11.	Основные проектные операции Java.
12.	Назначение основных элементов кода Java.
13.	Перегрузка операторов.
14.	Классы. Объекты. Члены, поля, методы.
15.	Классы. Объекты. Доступ к членам класса.
16.	Конструкторы и деструкторы класса.
17.	Статические и константные методы и данные класса.
18.	Постройте иерархию базовых классов VCL?
19.	Перечислите и опишите компоненты отображения информации.
20.	Опишите особенности работы с канвой. Укажите компоненты, обладающие канвой. Перечислите основные методы канвы.
21.	Перечислите компоненты VCL, реализующие работу системных диалогов различного назначения. Укажите их основные свойства и методы.
22.	Операции с классами as и is. Приведите примеры их использования.
23.	Укажите основные принципы работы с приложениями, окнами, страницами. Перечислите основные события страниц.
24.	Приложение стандарта Windows. MDI и SDI.
25.	Приложение стандарта Windows. Установка и настройка приложения.

### 5.3. Перечень видов оценочных средств

Задания для самостоятельного выполнения  
 Задания для лабораторных работ  
 Проектные работы  
 Контрольные вопросы и задания

### 5.4. Процедура применения оценочных материалов

Промежуточная аттестация может проводиться с применением электронного обучения и (или) дистанционных образовательных технологий в соответствии с "Порядком проведения промежуточной аттестации с применением электронного обучения и /или дистанционных образовательных технологий".

Описание балльно-рейтинговой системы по дисциплине.

Итоговая рейтинговая оценка по дисциплине складывается из следующих составляющих:

- 1) В течение семестра за выполнение заданий по курсу студент может максимально получить 60 баллов.;
- 2) Обязательной формой текущей аттестации знаний является выполнение индивидуального проектного задания 20 баллов.
- 3) На экзамене ответ студента может быть максимально оценен в 30 баллов.

При этом, для получения положительной итоговой оценки по дисциплине необходимо получить не менее 60% по каждой составляющей и выполнить все задания для лабораторных работ. Шкала перевода баллов в оценку:

86-100	отлично
70-85	хорошо
50-69	удовлетворительно
менее 50	неудовлетворительно

Выполнение заданий:	70
Лабораторные работы	50
Индивидуальное проектное задание	20
Экзамен	30
<b>ИТОГО:</b>	<b>100</b>

## 6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

### 6.1. Рекомендуемая литература

#### 6.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л1.1	Зыков С. В.	Введение в теорию программирования. Объектно-ориентированный подход	, 2016	<a href="http://biblioclub.ru/index.php?page=book&amp;id=42907">http://biblioclub.ru/index.php?page=book&amp;id=42907</a> <u>3</u>

	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л1.2	Мейер, Б.	Объектно-ориентированное программирование и программная инженерия	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019	<a href="http://www.iprbookshop.ru/79706.html">http://www.iprbookshop.ru/79706.html</a>
Л1.3	Куренкова, Т. В., Светозарова, Г. И.	Основы алгоритмизации и объектно-ориентированного программирования: учебное пособие	Москва: Издательский Дом МИСиС, 2011	<a href="http://www.iprbookshop.ru/98879.html">http://www.iprbookshop.ru/98879.html</a>

#### 6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л2.1	Васюткина, И. А.	Технология разработки объектно-ориентированных программ на JAVA: учебно-методическое пособие	Новосибирск: Новосибирский государственный технический университет, 2012	<a href="http://www.iprbookshop.ru/45047.html">http://www.iprbookshop.ru/45047.html</a>
Л2.2	Мухаметзянов, Р. Р.	Основы программирования на Java: учебное пособие	Набережные Челны: Набережночелнинский государственный педагогический университет, 2017	<a href="http://www.iprbookshop.ru/66812.html">http://www.iprbookshop.ru/66812.html</a>
Л2.3	Биллиг, В. А.	Основы программирования на C#	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016	<a href="http://www.iprbookshop.ru/73695.html">http://www.iprbookshop.ru/73695.html</a>
Л2.4	Туральчук, К. А.	Параллельное программирование с помощью языка C#	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019	<a href="http://www.iprbookshop.ru/79714.html">http://www.iprbookshop.ru/79714.html</a>

#### 6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	Среда электронного обучения LMS Moodle [Электронный ресурс] URL: <a href="http://moodle.tsput.ru">http://moodle.tsput.ru</a>
Э2	Microsoft Docs "Техническая документация, материалы по API и примеры кода"[Электронный ресурс] URL: <a href="http://docs.microsoft.com/">http://docs.microsoft.com/</a>

#### 6.3. Информационные технологии

##### 6.3.1 Перечень лицензионного и свободно распространяемого программного обеспечения

1.	Операционная система ROSA Enterprise Linux Desktop № RL00450-1-110518-01. RL00450-1-110518-17 от 11 мая 2018 г.
2.	Операционная система Microsoft Windows Professional 7 Russian. Лицензия №48497058 от 13.05.2011 г., договор № Пр/16/6 от 05 апреля 2016 г.
3.	Операционная система Microsoft Windows 10 Professional Russian. Контракт № ПР/ФЕН/15/18 от 23.10.2015 г., договор № Пр/16/6 от 05 апреля 2016 г.
4.	Текстовый редактор NotePad++. Свободно распространяемое ПО
5.	Оболочка программирования Code: Blocks 17.12. Свободно распространяемое ПО
6.	Среда программирования и набор инструментов для программирования. MinGW 0.6.3 Свободно распространяемое ПО
7.	Редактор диаграмм, схем, блок-схем, UML-схем Dia 0.97.2. Свободно распространяемое ПО

##### 6.3.2 Перечень информационных справочных систем и профессиональных баз данных

1.	Базы данных издательства Springer ( <a href="https://link.springer.com">https://link.springer.com</a> )
2.	Web of Science Core Collection – политематическая реферативно-библиографическая и наукометрическая (библиометрическая) база данных ( <a href="http://webofscience.com">http://webofscience.com</a> )
3.	Компьютерная информационно-правовая система «Гарант»
4.	Портал Федеральных государственных образовательных стандартов высшего образования ( <a href="http://fgosvo.ru">http://fgosvo.ru</a> )

**7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)**

Ауд.	Назначение	Оборудование и технические средства обучения	Вид
4-303	Помещение для самостоятельной работы	аудиоколонки, кондиционер, маркерная доска, столы компьютерные, столы учебные, компьютерная техника с возможностью подключения сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду университета	
4-305	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, аудиоколонки учебные, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	
4-306	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	

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

Приступая к изучению новой учебной дисциплины, студенты должны ознакомиться с учебной программой, учебной, научной и методической литературой, имеющейся в библиотеке университета, встретиться с преподавателем, ведущим дисциплину, получить в библиотеке рекомендованные учебники и учебно-методические пособия, осуществить запись на соответствующий курс в среде электронного обучения университета.

Глубина усвоения дисциплины зависит от активной и систематической работы студента на лекциях и лабораторных работ, а также в ходе самостоятельной работы, по изучению рекомендованной литературы.

На лекциях важно сосредоточить внимание на ее содержании. Это поможет лучше воспринимать учебный материал и уяснить взаимосвязь проблем по всей дисциплине. Основное содержание лекции целесообразнее записывать в тетради в виде ключевых фраз, понятий, тезисов, обобщений, схем, опорных выводов. Необходимо обращать внимание на термины, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Желательно оставлять в конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющей материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. С целью уяснения теоретических положений, разрешения спорных ситуаций необходимо задавать преподавателю уточняющие вопросы. Для закрепления содержания лекции в памяти, необходимо во время самостоятельной работы внимательно прочесть свой конспект и дополнить его записями из учебников и рекомендованной литературы. Конспектирование читаемых лекций и их последующая доработка способствует более глубокому усвоению знаний, и поэтому являются важной формой учебной деятельности студентов.

Прочное усвоение и долговременное закрепление учебного материала невозможно без продуманной самостоятельной работы. Такая работа требует от студента значительных усилий, творчества и высокой организованности. В ходе самостоятельной работы студенты выполняют следующие задачи: дорабатывают лекции, изучают рекомендованную литературу, готовятся к лабораторным работам, к коллоквиуму, контрольным работам по отдельным темам дисциплины. При этом эффективность учебной деятельности студента во многом зависит от того, как он распорядился выделенным для самостоятельной работы бюджетом времени.

Результатом самостоятельной работы является прочное усвоение материалов по предмету согласно программе дисциплины. В итоге этой работы формируются профессиональные умения и компетенции, развивается творческий подход к решению возникших в ходе учебной деятельности проблемных задач, появляется самостоятельности мышления.

Целью лабораторных работ по данной дисциплине является закрепление теоретических знаний, полученных при изучении дисциплины.

При подготовке к лабораторной работе целесообразно выполнить следующие рекомендации: изучить основную литературу; ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях: журналах, газетах и т. д.; при необходимости доработать конспект лекций. При этом учесть рекомендации преподавателя и требования учебной программы.

При выполнении лабораторных работ основным методом обучения является самостоятельная работа студента под управлением преподавателя. На них пополняются теоретические знания студентов, их умение творчески мыслить, анализировать, обобщать изученный материал, проверяется отношение студентов к будущей профессиональной деятельности.

Оценка выполненной работы осуществляется преподавателем комплексно: по результатам выполнения заданий, устному сообщению и оформлению работы. После подведения итогов занятия студент обязан устранить недостатки, отмеченные преподавателем при оценке его работы.