**План-конспекты уроков по дисциплине «МДК 01.01 Разработка программных модулей»**

Исполнитель:

Вавилова Д.В.

**План-конспект урока**

**Тема урока:** Объектно-ориентированное программирование (ООП)

**Дата:** 16.02.2023

**Группа:** 214

**Тип урока:** лекция

**Формы организации учебной деятельности:** фронтальная

**Метод обучения :** Словесные, наглядно-демонстрационные.

**Средства обучения :** проектор ,интерактивная доска , презентация , методические материал.

Цель: ознакомить обучающих созданию модулей объектов в объектно-ориентированное программирование (ООП)

**Задачи:**

**Предметные:** формирование представление о создании модулей в программировании.

**Личностные:** готовить и способность к самостоятельной и групповой работе.

**Метапредметные:** владеть навыками познавательной, учебно-исследовательской и проектной деятельности.

**Общие и профессиональные компетенции:**

ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 9 Использовать информационные технологии в профессиональной деятельности.

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием.

***Ход занятия:***

***1. Организационный момент.***

Преподаватель приветствует студентов:

Здравствуйте, уважаемые студенты. Можете присаживаться. Отметим присутствующих.

***Проверка домашнего задания.***

Проверка домашнего задания. На прошлом занятий мы проходили инструментальные средства среды разработки. Вы должны были выписать все инструментальные средства и подробное описание. Проверяю и отмечаю в журнал оценку.

**Тема сегодняшнего занятия Объектно-ориентированное программирование (ООП)**

На занятия мы ознакомимся с основными понятиями и определениями в программирование для создания объектов классов и какие команды и функции используются при создании окон и т.д.

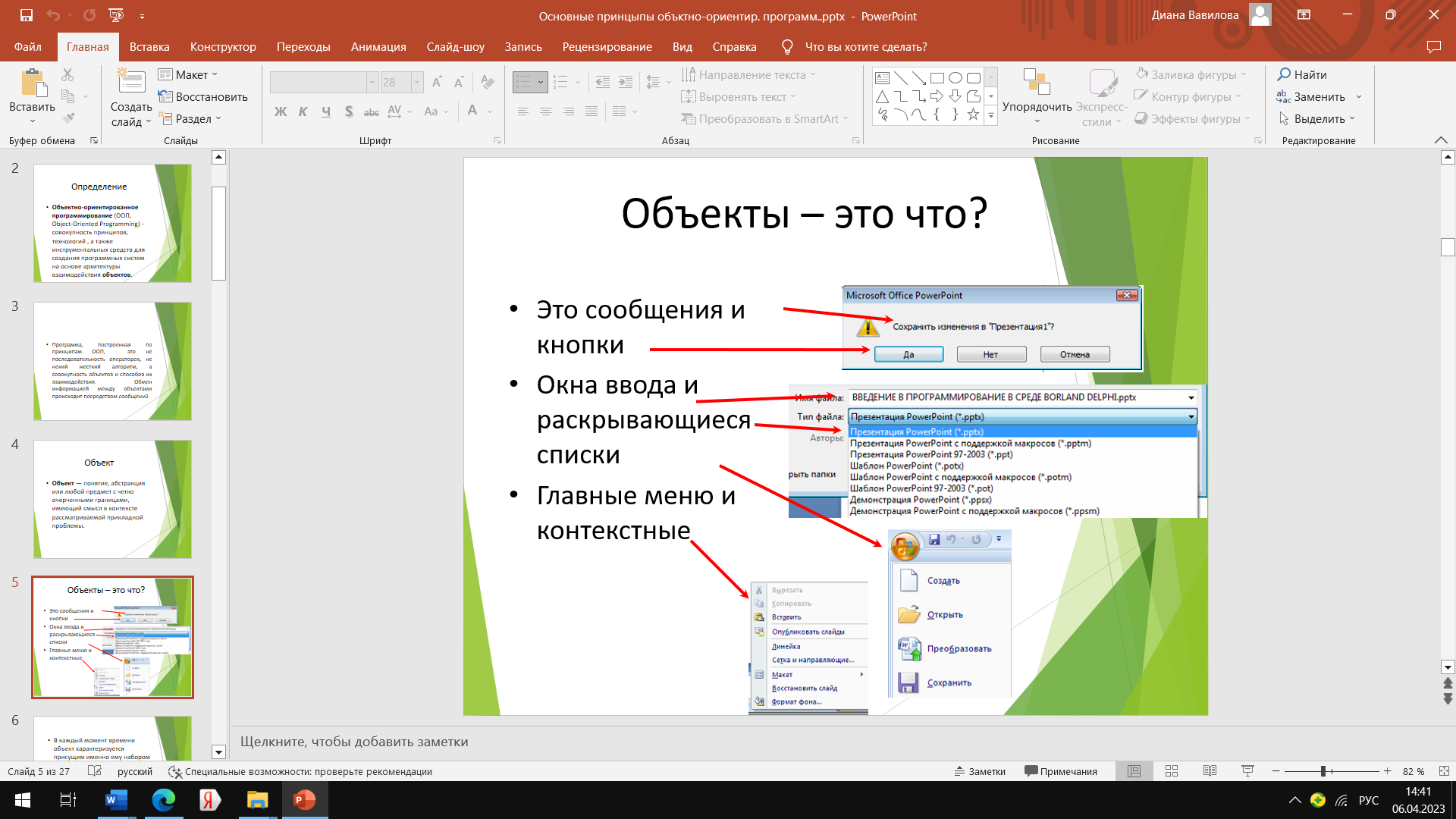
***Определение***

Объектно-ориентированное программирование (ООП, Object-Oriented Programming) - совокупность принципов, технологий , а также инструментальных средств для создания программных систем на основе архитектуры взаимодействия объектов.

Программа, построенная по принципам ООП -это не последовательность операторов, не некий жесткий алгоритм, а совокупность объектов и способов их взаимодействия. Обмен информацией между объектами происходит посредством *сообщений.*

*Что такое объект?*

* **Объект** — понятие, абстракция или любой предмет с четко очерченными границами, имеющий смысл в контексте рассматриваемой прикладной проблемы.



* В каждый момент времени объект характеризуется присущим именно ему набором **свойств** (properties) и **методов** (methods) – операций, совершаемых над другими объектами или данным объектом, а также реагирует на **события** (events).

*Что такое Свойства?*

* **Свойства** – перечень параметров объекта, которые определяют внешний вид и поведение объекта, выделяют уникальные особенности каждого экземпляра.
* К свойствам относятся: имя, тип, значение, цвет, размер и др.
* **Состояние**– совокупность всех свойств данного объекта.

**Методы**

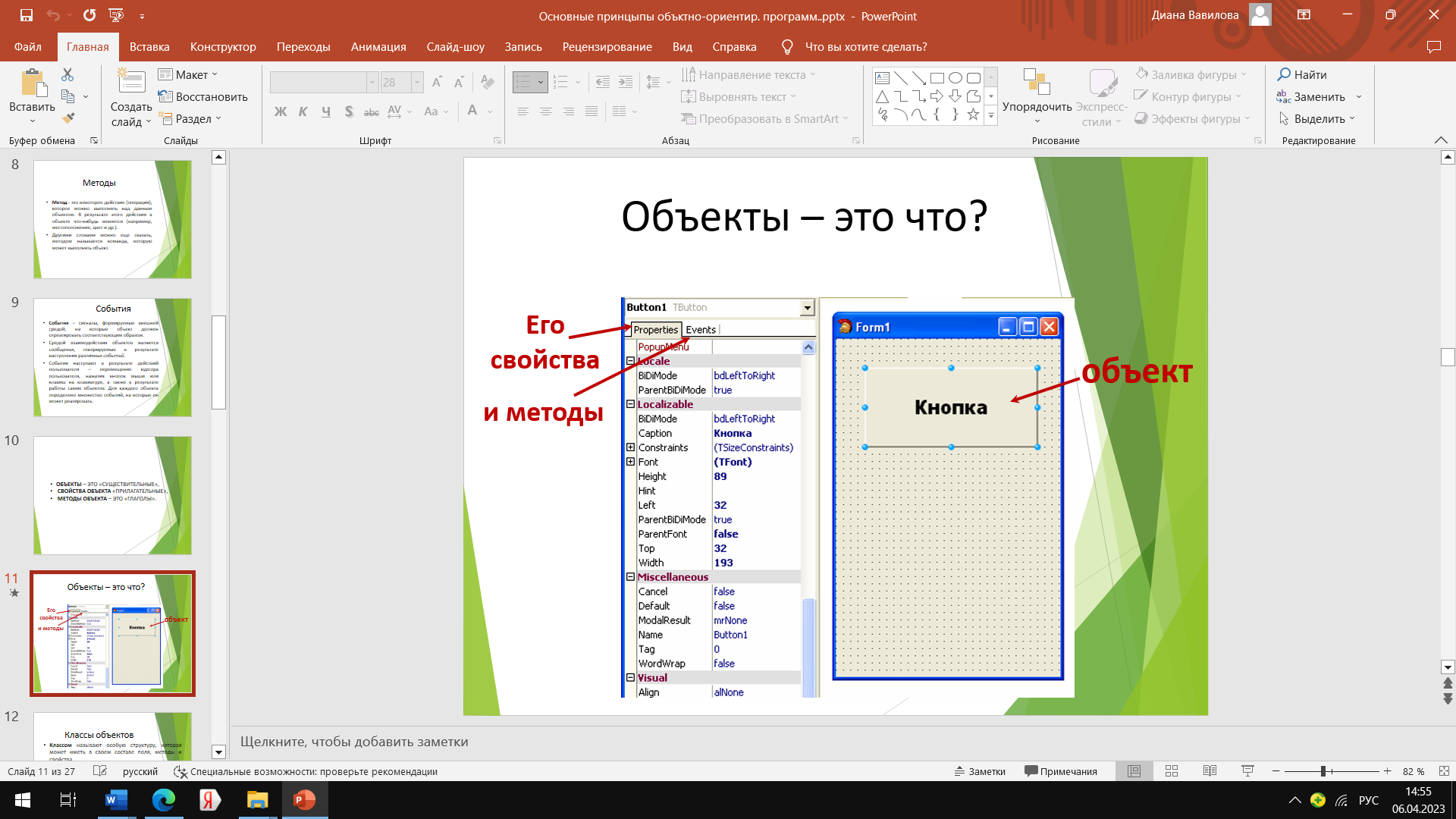
* **Метод****-** это некоторое действие (операция), которое можно выполнять над данным объектом. В результате этого действия в объекте что-нибудь меняется (например, местоположение, цвет и др.).
* Другими словами можно еще сказать, методом называется команда, которую может выполнять объект.

**События**

*События* – сигналы, формируемые внешней средой, на которые объект должен отреагировать соответствующим образом.

Средой взаимодействия объектов являются *сообщения*, генерируемые в результате наступления различных *событий*.

*События наступают в результате действий пользователя* – перемещение курсора пользователя, нажатия кнопок мыши или клавиш на клавиатуре, а также в результате работы самих объектов. Для каждого объекта определено множество событий, на которые он может реагировать.



* **Классом** называют особую структуру, которая может иметь в своем составе поля, методы и свойства.
* **Класс** представляет собой множество **объектов**
  + имеющих общую структуру
  + обладающих одинаковым поведением.
* Класс выступает в качестве объектного типа данных, а объект – это конкретный *экземпляр* класса.

**Пример**

Каждый из классов обладает специфическим набором свойств, методов и событий.

Например, в приложении Word существует класс объектов «Документ»(Document), который обладает определенными наборами:

* **Свойств**: имя (Name), полное имя (FullName) и так далее;
* **Методов**: открыть документ (Open), напечатать документ (PrintOut), сохранить документ (Save);
* **Событий**: создание документа (Document\_New), закрытие документа (Document\_Close) и т.д.

Объявление класса

Туре

<имя класса> = Class(<имя класса - родителя>)

public // т.е. доступно всем

<поля, методы, свойства, события>

published // т.е. видны в Инспекторе Объекта и изменяемы

<поля, свойства>

protected // доступно только потомкам

<поля, методы, свойства, события>

private// доступно только в этом модуле

<поля, методы, свойства, события>

end;

*Создание класса*

MyClass = class – создаем класс Myclass.

Name: string;

Color: string; – описываем параметры объекта, его имя и цвет. Практически задаем переменные.

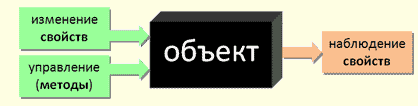
function MyNewFunction(o: MyClass): string ;

– создаем метод используя один параметр передавая в него объект.

сonstructor Create (NewName: string; NewColor: string); – формируем конструктор и задаем начальные значения параметров объекта.

destructor Destroy; – разрушаем объект.

* Мы ничего не знаем о том, как устроен объект внутри. Для нас это, как говорят кибернетики, «черный ящик».
* Для того, чтобы работать с объектом, не нужно знать его внутреннее устройство! Достаточно, что можно узнать и изменить его свойства, а также применять доступные методы управления.



Свойства и методы представляют собой **интерфейс** объекта, то есть способ его общения с внешним миром.

**Инкапсуляция**

* Пример: объект - телевизор.
* Внутри этого объекта находятся множество других объектов: микросхемы, провода, электронно-лучевая трубка и так далее. Но при взаимодействии с телевизором мы об этом даже и не задумываемся. В этом заключается первый принцип ООП - *инкапсуляция*.

**Инкапсуляция (encapsulation)** — это сокрытие реализации класса и отделение его внутреннего представления от внешнего (интерфейса).

***Наследование***

* Пример: цветной телевизор произошел от черно-белого, а телевизор с плазменным экраном - от обыкновенного.
* При этом каждый потомок наследовал свойства и функции предшественника, дополняя их своими, качественно новыми.
* **Наследование позволяет расширить возможности объекта, не создавая при этом новый объект с нуля.**

**Наследование (inheritance)** — это отношение между классами, при котором класс использует структуру или поведение другого класса (одиночное наследование), или других (множественное наследование) классов.

Наследование вводит иерархию «общее/частное», в которой подкласс наследует от одного или нескольких более общих суперклассов.

Подклассы обычно дополняют или переопределяют унаследованную структуру и поведение*.*

**Преимущества ООП:**

(при создании больших программ):

* использование при программировании понятий, более близких к предметной области;
* локализация свойств и поведения объекта о одном месте, позволяющая лучше структурировать и, следовательно, отлаживать программу;
* возможность создания библиотеки объектов и создания программы из готовых частей;
* исключение избыточного кода за счет того, что можно многократно не описывать повторяющиеся действия;
* сравнительно простая возможность внесения изменений в программу без изменения уже написанных частей, а в ряде случаев и без их перекомпиляции.

**Подведение итогов**

Подведение итогов занятия, анализ усвоения учебного материала методом рефлексии и устного опроса*.*

-Что такое объект?

-Что такое класс, и что в него входит?

-какие действия выполняет объект?

**Домашняя работа:** повторение пройдённого материала.