Министерство образования Республики Беларусь

УО “Мозырский государственный педагогический университет

имени И. П. Шамякина”

ФИЗИКО-ИНЖЕНЕРНЫЙ ФАКУЛЬТЕТ

Кафедра теоретической физики и прикладной информатики

Курсовая работа

по дисциплине “Объектно-ориентированное программирование”

Реализация принципов ООП при разработке телеграмм-бота

**Выполнил:**

Студент 2 курса 2 группы

Физико-инженерного Факультета

Хомутовский Илья Владимирович

Научный руководитель:

Сафронов Артур Павлович

УО «Мозырский государственный педагогический университет имени И.П.Шамякина»

Физико-инженерный факультет

Кафедра теоретической физики и прикладной информатики

УТВЕРЖДАЮ

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

« \_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г.

## ЗАДАНИЕ

## по курсовому проекту (работе) студента

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## (фамилия, имя, отчество)

## Дисциплина \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тема работы:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Цель работы:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Состав предоставляемых к защите документов:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Календарный график работы над проектом на весь период работы (с указанием сроков выполнения и трудоемкости отдельных этапов):

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ до \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ года;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ до \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ года;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ до \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ года.

Дата предоставления курсовой работы (проекта): «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

Дата выдачи задания: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

Задание принял к исполнению: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

Руководитель работы: \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (И.О. Фамилия)

РЕЗЮМЕ

Выполнил: Хомутовский Илья Владимирович

Тема работы: Реализация принципов ООП при разработке бота в телеграмм.

Ключевые слова: объектно-ориентированное программирование, полиморфизм, наследование, инкапсуляция, класс, экземпляр класса, метод, бот, асинхронная функция, Python.

Объект исследования: ООП, бот в телеграмм

Предмет исследования: реализация принципов ООП в языке программирования Python

Цель работы: изучение принципов объектно-ориентированного программирования в Python при создании телеграмм бота.

(Метод исследования, Полученные результаты, степень использования или рекомендации по использованию, область применения)

Задачи работы:

1. Изучить основные принципы ООП, а также примеры их реализации в языке программирования Python

2. Определить сущность понятий «объектно-ориентированное программирование», «класс», «экземпляр класса», «метод»

3. Рассмотреть средства разработки бота в телеграмм, рассмотреть распространённые библиотеки и их особенности.

4. Построить UML-диаграмму, определить общий вид структуры проекта.

5. Реализовать поставленные задачи, написать код

6. Провести тестирование всех систем бота

7. В случае возникновения ошибок, перейти к пункту 5.

Структура работы: данная курсовая работа включает резюме, введение, N главы заключение, список использованных источников из ~10 наименований приложения.

Объем работы: страниц 37, рисунков 7.

Курсовая работа посвящена изучению ключевых понятий и особенностей возникновения и развития данной парадигмы. Особое внимание уделено основным принципам объектного программирования.

Рассмотрены преимущества и недостатки объектно-ориентированного языка программирования Python.

* оглавление;
* перечень условных обозначений (при необходимости);
* введение;

Создание бота в Telegram с помощью Python и библиотеки aiogram очень полезно и важно по нескольким причинам.

Во-первых, боты позволяют автоматизировать задачи. Например, можно создать бота, который будет отправлять уведомления о новых сообщениях на веб-сайте или предоставлять информацию о погоде в определенном городе. Такой бот может сэкономить время и сделать выполнение повседневных задач более эффективным.

Во-вторых, боты позволяют взаимодействовать с пользователями. Вы можете создать бота, который будет отвечать на вопросы пользователей, предоставлять информацию или выполнять запросы к базе данных. Это упростит коммуникацию с вашей аудиторией и обеспечит быстрый доступ к нужной информации.

В-третьих, библиотека aiogram для Python обладает множеством функций, позволяющих работать с различными типами сообщений и медиафайлов. Она делает процесс разработки ботов более гибким и удобным.

В-четвертых, Python является простым и гибким языком программирования, поэтому создание ботов с его использованием достаточно просто даже для новичков в программировании.

Наконец, есть активное сообщество разработчиков, которое поддерживает и помогает развивать библиотеку aiogram. Вы можете найти много полезных материалов и примеров кода для создания ботов.

Таким образом, создание телеграм-бота с использованием принципов ООП и библиотеки aiogram является интересной и полезной темой для курсовой работы, которая позволяет применить знания по программированию и разработке, а также создать полезный инструмент для автоматизации задач и взаимодействия с пользователями.

Глава 1.1 Общие сведения.

Объектно-ориентированное программирование (ООП) - это парадигма программирования, которая фокусируется на создании объектов, которые могут содержать как данные, так и поведение. Python - это высокоуровневый язык программирования, который поддерживает ООП. Вот основные принципы ООП в Python:

Инкапсуляция: Этот принцип связан с скрытием деталей реализации объекта от пользователя. Это достигается путем определения атрибутов и методов как публичных, защищенных или приватных. Публичные атрибуты и методы могут быть доступными для любого другого объекта, защищенные атрибуты и методы могут быть доступными для объектов того же класса или его подклассов, а приватные атрибуты и методы могут быть доступными только для самого объекта.

Наследование: Этот принцип позволяет создавать новый класс на основе существующего класса, наследуя все его атрибуты и методы. Затем новый класс может добавлять или переопределять любые из этих унаследованных атрибутов и методов. Наследование позволяет повторное использование кода и помогает создавать иерархию классов.

Полиморфизм: Этот принцип относится к способности объектов принимать множество форм. В Python полиморфизм может быть достигнут путем переопределения методов и перегрузки методов. Переопределение методов происходит, когда подкласс предоставляет другую реализацию метода, который уже определен в его суперклассе. Перегрузка методов происходит, когда класс определяет несколько методов с одним и тем же именем, но с разными параметрами.

Абстракция: Этот принцип связан с моделированием сложных систем путем скрытия ненужных деталей от пользователя. В Python абстракцию можно достичь с использованием абстрактных классов и интерфейсов. Абстрактный класс - это класс, который не может быть создан экземпляром и предназначен для наследования. Интерфейс - это набор методов, которые должны быть реализованы любым классом, который реализует интерфейс.

В итоге, вот четыре основных принципа ООП в Python: инкапсуляция, наследование, полиморфизм и абстракция. Понимание и применение этих принципов могут привести к более модульному, повторно используемому и поддерживаемому коду.

Глава 1.2 Полиморфизм

Полиморфизм в языке Python относится к способности объектов разных типов проявлять разное поведение при вызове одноименных методов. Это позволяет работать с объектами различных классов, используя общий интерфейс.

В Python полиморфизм может быть достигнут через два механизма: методное переопределение (method overriding) и методную перегрузку (method overloading).

Методное переопределение (method overriding) позволяет классу-потомку предоставить свою собственную реализацию метода, который уже определен в его классе-родителе. При вызове этого метода у объекта класса-потомка будет использована его собственная реализация, а не реализация из класса-родителя. Для переопределения метода необходимо в классе-потомке создать метод с тем же именем.

Пример переопределения встроенного метода str для объекта класса User:

(Вставить картинку)

* заключение;

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Павловская, Т.А. С/С++. Структурное и объектно-ориентированное программирование: Практикум [Текст] / Т.А.Павловская, Ю.А.Щупак –СПб.: Питер, 2011. –352 с.
2. Лутц, М. (2013). Изучаем Python: Мощное объектно-ориентированное программирование. Издательство O'Reilly Media.
3. Россум, Г.ван (1995). The Python Language Reference. Python Software Foundation.
4. Лаптев, В.В. С++. Объектно-ориентированное программирование: Учебное пособие [Текст] / В.В.Лаптев. – СПб.: Питер. 2008. – 464 с.
5. Принципы ООП в программировании
6. Преимущества применения ООП при разработке оконных приложений.
7. Смит, Дж. (2019). Объектно-ориентированное программирование на Python 3: Создание надежного и поддерживаемого программного обеспечения с использованием объектно-ориентированных паттернов проектирования на Python 3. Издательство Packt Publishing.
8. Петрова, Ю.А. Дифференцированный подход при обучении объектно-ориентированному программированию в старшей школе: дис. … канд. пед. наук. – Санкт-Петербург, 2002. ­–169 с.
9. Бертран, М. Основы объектно-ориентированного программирования [Электронный ресурс] / Бертран М. – Электронное издательства «Интернет-университет информационных технологий – ИНТУИТ.ру», 2005 г.
10. Гвидо ван Россум о Python и ООП.