Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение   
высшего образования

«Алтайский государственный технический университет им. И.И. Ползунова»

Университетский технологический колледж

наименование подразделения

Кафедра Информационные системы в экономике

наименование кафедры

Направление Информационные системы и программирование

Отчёт защищен с оценкой\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.В Воробьев

(подпись руководителя от вуза) (инициалы, фамилия)

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025г.

ОТЧЕТ  
по лабораторной работе №4  
 [Некоторые конструкции](https://portal.altstu.ru/nodexx/w95/r/astu/lms/view-object?p7_id_kurs=2b856cba-b43a-90d1-e060-000a0301d4d9&p7_id_object=3633272&session=2291853890243&cs=3KVL3g-watQ6KG_yWWP3hIM78iGjffzqfte33bbjZB5VlO2bVL7Fc_eQaoRHwTqKekTzmMaw6-xQppwIyDi7_vQ)\_\_\_\_\_\_\_\_\_\_\_

тема лабораторной работы

по дисциплине Объектно-ориентированное программирование

ЛР 09.02.07.28.001 ПЗ

обозначение документа

Студент группы 1ИСП-22 Т.С. Луконин

инициалы, фамилия

Руководитель работы преподаватель К.В. Воробьев

должность, ученое звание инициалы, фамилия

Барнаул 2025

**Задание 1: Обработка исключений**

* Используйте try-except для обработки исключений.

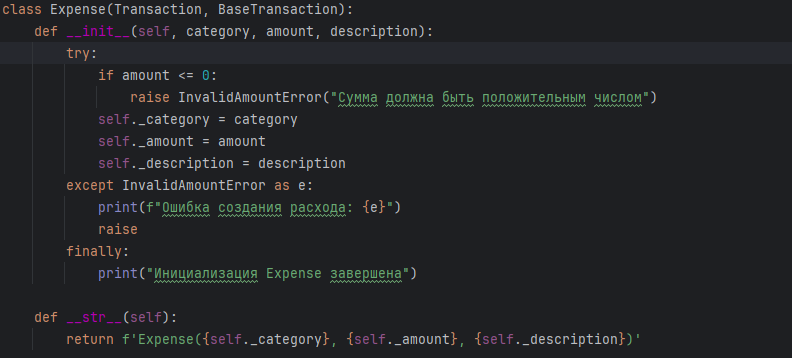


Рисунок 1 - использование конструкции трай ехпект

* Добавьте finally для выполнения обязательного кода.

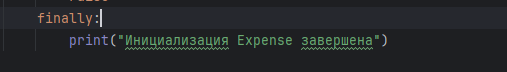


Рисунок 2 - использование finally

* Используйте raise для генерации собственных исключений.



Рисунок 3 - использование raise для генерации своего исключения

* Создайте иерархию пользовательских исключений (например, BaseException -> CustomError -> SpecificError). Обработайте каждое исключение по-разному в зависимости от его типа.

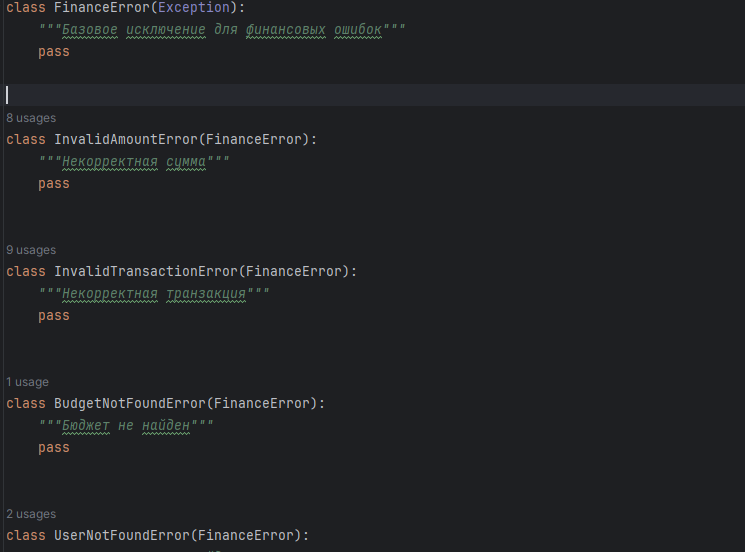


Рисунок 4 - мои исключения

**Задание 2: Работа с массивами объектов**

* Создайте класс и продемонстрируйте работу с одномерным и двумерным списками его объектов.

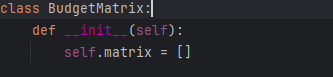


Рисунок 5 - инициализация матрицы

Создаётся класс BudgetMatrix который можно использовать для работы с матрицей бюджета

* + Реализуйте метод, который находит объект с максимальным значением определенного атрибута в двумерном списке. Убедитесь, что метод корректно обрабатывает пустые списки.

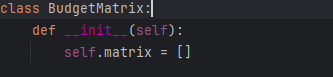


Рисунок 6 - создание матрицы

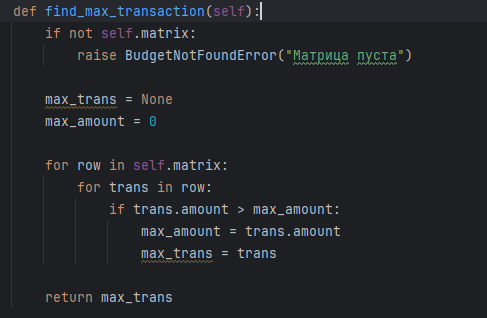


Рисунок 7 - поиск максимального элемента в матрице

**Задание 3: Наследование**

* Создайте базовый и производный классы.
* Переопределите метод базового класса в производном.
* Продемонстрируйте вызов метода базового класса из производного.
* Добавьте в производный класс метод, который использует как переопределенный метод, так и метод базового класса, но в разной последовательности в зависимости от условия.

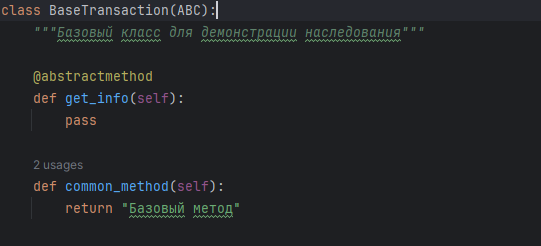


Рисунок 8 - Создание базового и производного класса.



Рисунок 9 - Переопределение метода базового класса в производствен.

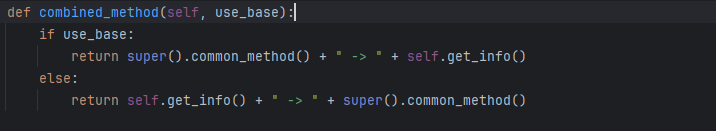


Рисунок 10 - демонстрация вызова метода базового класса из производного.

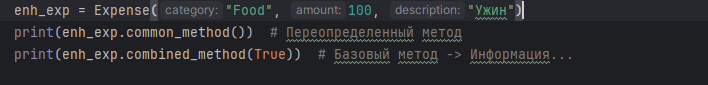


Рисунок 11 - Добавление в производный класс метод, который использует как переопределенный метод, так и метод базового класса, но в разной последовательности в зависимости от условия.

**Задание 4: Защищенные атрибуты**

* Используйте \_ для обозначения защищенных атрибутов.
* Продемонстрируйте доступ к ним в производном классе.

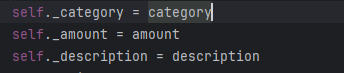


Рисунок 12 - Использование \_ для обозначения защищённых атрибутов.

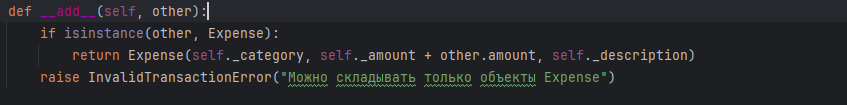


Рисунок 13 - Продемонстрируйте доступ к ним в производном классе.

**Задание 5: Конструкторы и наследование**

* Создайте конструктор в производном классе, вызывающий конструктор базового через super().

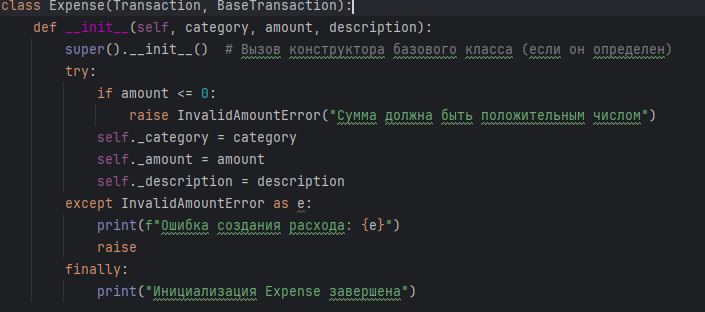


Рисунок 14 - создание конструктора с использования super

Работа с абстрактным классом, была продемонстрирована в лр ранее

**Задание 6: Строковое представление**

* Переопределите \_\_str\_\_ для вывода информации об объекте.
* Реализуйте метод \_\_repr\_\_ так, чтобы он возвращал строку, которую можно использовать для повторного создания объекта (например, с помощью eval).

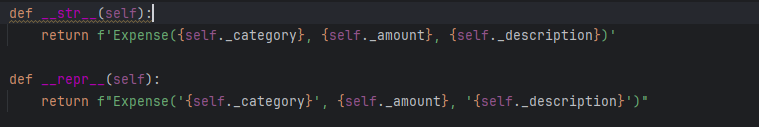


Рисунок 15 - перегрузка операторов

Оператор str создаёт удобночитаемое строковое представление объекта, которое показывает:

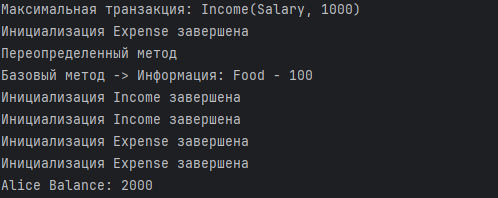


Рисунок 16 - демонстрация работы программы

https://github.com/Lukonin-eng/Project