



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ М.В.ЛОМОНОСОВА

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ  
КАФЕДРА АЛГОРИТМИЧЕСКИХ ЯЗЫКОВ

Отчет о выполнении задания практикума

**Вариант «Система управления  
инвестиционным портфелем»**

Лебедев Андрей Алексеевич  
Группа 424

Москва, 2025

# Содержание

1	Постановка задачи	3
2	Диаграмма классов	6
3	Текстовые спецификации основных классов	9
4	Диаграмма объектов	10
5	Инструментальные средства	11
6	Файловая структура	12
7	Пользовательский интерфейс	13

# 1 Постановка задачи

## Уточнение постановки задачи

- Разработать программную систему, осуществляющую имитационное моделирование процесса или явления и визуализирующую этот процесс или явление.
- Использовать для создания системы один из объектно-ориентированных языков программирования (в данном случае Python).
- Провести с помощью разработанной системы исследование поведения моделируемого процесса, задавая для этого различные значения параметров, от которых зависит этот процесс.

## Основные требования к системе

- Система должна быть спроектирована на основе методологии объектно-ориентированного программирования, т.е. должна быть представлена в виде совокупности взаимодействующих друг с другом объектов, причём каждый объект является экземпляром определённого класса, а классы образуют иерархию. В ходе объектно-ориентированного проектирования необходимо определить и зафиксировать логическую структуру (классы и объекты) и файловую (модульную) структуру системы.
- Система должна предоставлять удобный и понятный пользовательский интерфейс, предусматривающий проведение экспериментов по моделированию и отображение необходимой информации.
- Для проведения экспериментов по моделированию перед началом каждого эксперимента пользователь должен иметь возможность устанавливать нужные значения параметров, от которых зависит этот процесс или явление. Такие параметры называются параметрами моделирования, в их числе – шаг моделирования, т.е. отрезок времени, измеряемый в тех или иных единицах времени (секундах, минутах, днях и пр.), и/или число шагов моделирования.
- Поскольку в большинстве вариантов задания моделируемый процесс или явление зависит от нескольких неопределённых факторов, следует моделировать такие факторы статистически — на основе одного

из законов вероятностного распределения (равномерного, нормального и др.).

## Вариант: Система управления инвестиционным портфелем

- Разрабатываемая система реализует экономическую игру, участник которой — менеджер, управляющий работой инвестиционного фонда. Фонд осуществляет различные вложения собранных денежных средств с целью получения прибыли. Возможны вложения в:
  - срочные депозиты банков (валютные и рублёвые);
  - драгоценные металлы (золотые слитки и др.);
  - государственные облигации;
  - акции предприятий.

Все эти виды вложений различаются доходностью и риском (обычно доход пропорционален риску).

- В начале игры устанавливается общий капитал фонда (например, 560 тыс. у.е.), и определяется его портфель — какая часть капитала куда будет вложена. В портфеле не обязательно присутствуют все виды вложений; допускается несколько вложений одного типа.
- Также в начале игры задаётся внешняя конъюнктура: известны доступные виды вложений и их условия (процент по депозиту, цена акций и т.п.).
- Игра моделирует работу фонда в течение  $M$  месяцев ( $12 \leq M \leq 30$ ). Шаг моделирования — один месяц. В конце каждого месяца выполняются следующие действия:
  1. Расчёт доходности по всем элементам портфеля, определение прибыли и процента доходности.
  2. Выплата налога на прибыль фонда (например, 17% от суммы прибыли).
  3. Учёт новых поступивших денежных средств (например, от продажи паёв).
  4. Учёт расходов (например, возврат паёв держателями).

5. Реструктуризация портфеля с учётом изменённого капитала и новой внешней конъюнктуры.

- Операции (1) и (2) выполняются автоматически, операция (5) — игроком, а (3) и (4) могут выполняться как автоматически, так и вручную. Прибыль фонда влияет на количество вложений в него: при высокой доходности — приток капитала, при отрицательной — отток.
- Доходность по вкладам и облигациям известна заранее, доходность по акциям и металлам зависит от внешней конъюнктуры, которая моделируется случайным образом по законам распределения.
- Цель моделирования — выявление стратегий и структуры портфеля, позволяющих устойчиво увеличивать капитал фонда.
- Параметры, задаваемые пользователем: число месяцев  $M$ , исходный капитал, начальная структура портфеля, налоговая ставка, параметры волатильности рыночной конъюнктуры.
- На каждом шаге игроку доступны текущие данные по фонду: суммарный капитал, доходность по каждому активу, обновлённые условия внешней среды. По окончании игры отображается статистика за весь период (например, динамика капитала, структура доходов).

## 2 Диаграмма классов

В данном разделе представлены три диаграммы, отражающие архитектуру разработанной системы. Каждая из них отображает логически обособленную часть модели, что позволяет сделать представление более наглядным.

### 2.1 Архитектура игровой модели

На данной диаграмме показано взаимодействие основных компонентов симуляции: игровых объектов (*Game*, *Player*, *Fund*, *Portfolio*) и структуры инвестиций. Отражена связь между решениями игрока (*PlayerPurchase*, *Purchase*) и реальными инвестициями, входящими в портфель.

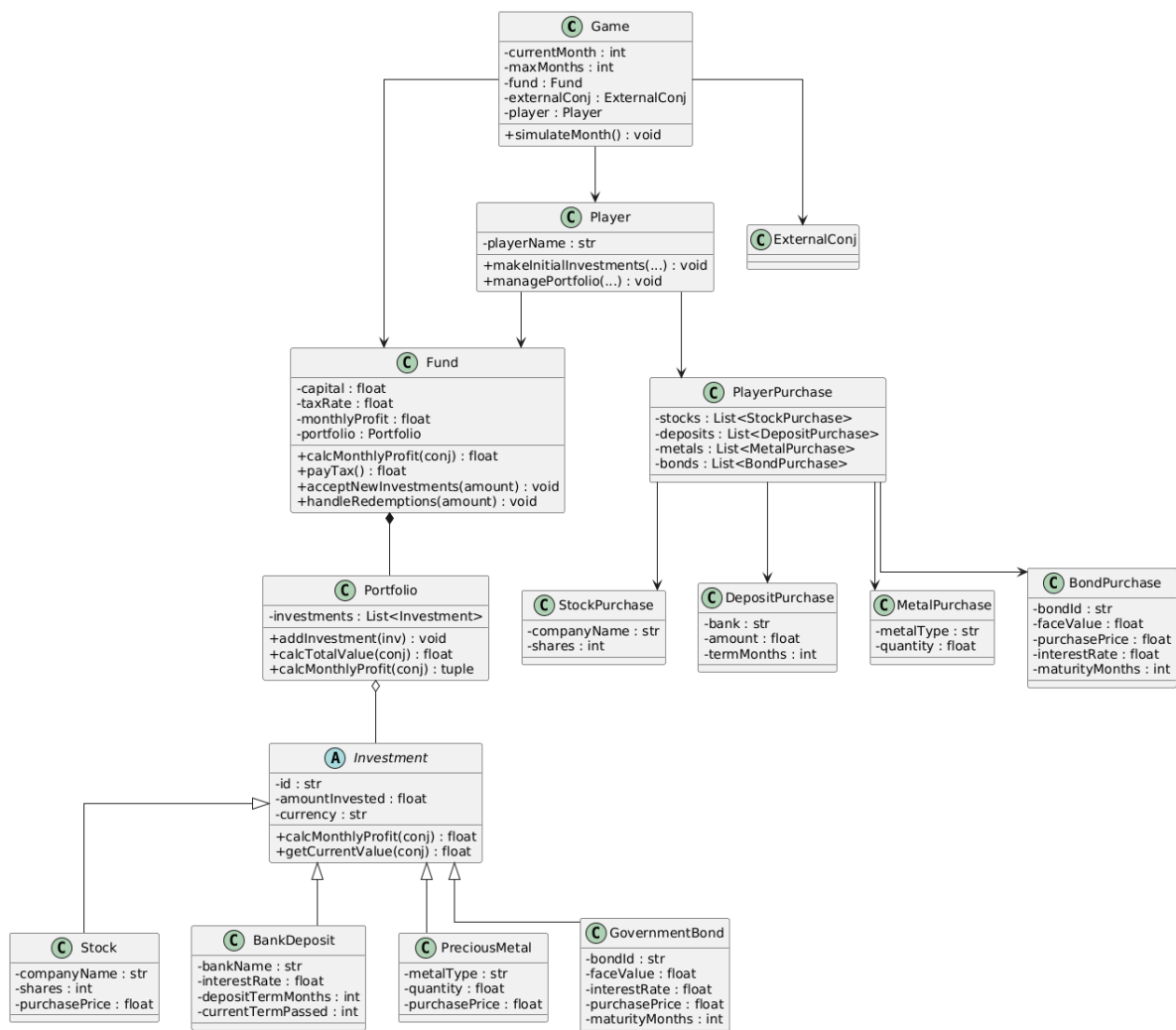


Рис. 1: Диаграмма классов: архитектура игровой модели

## 2.2 Рыночная среда и внешняя конъюнктура

В данной диаграмме представлено, как формируется и обновляется рыночная информация в процессе моделирования. Схема охватывает классы конфигурации (*MarketConfig*, *Config*), структуру рыночных данных и механизм обновления внешней конъюнктуры (*ExternalConj*, *ExternalConjData*).

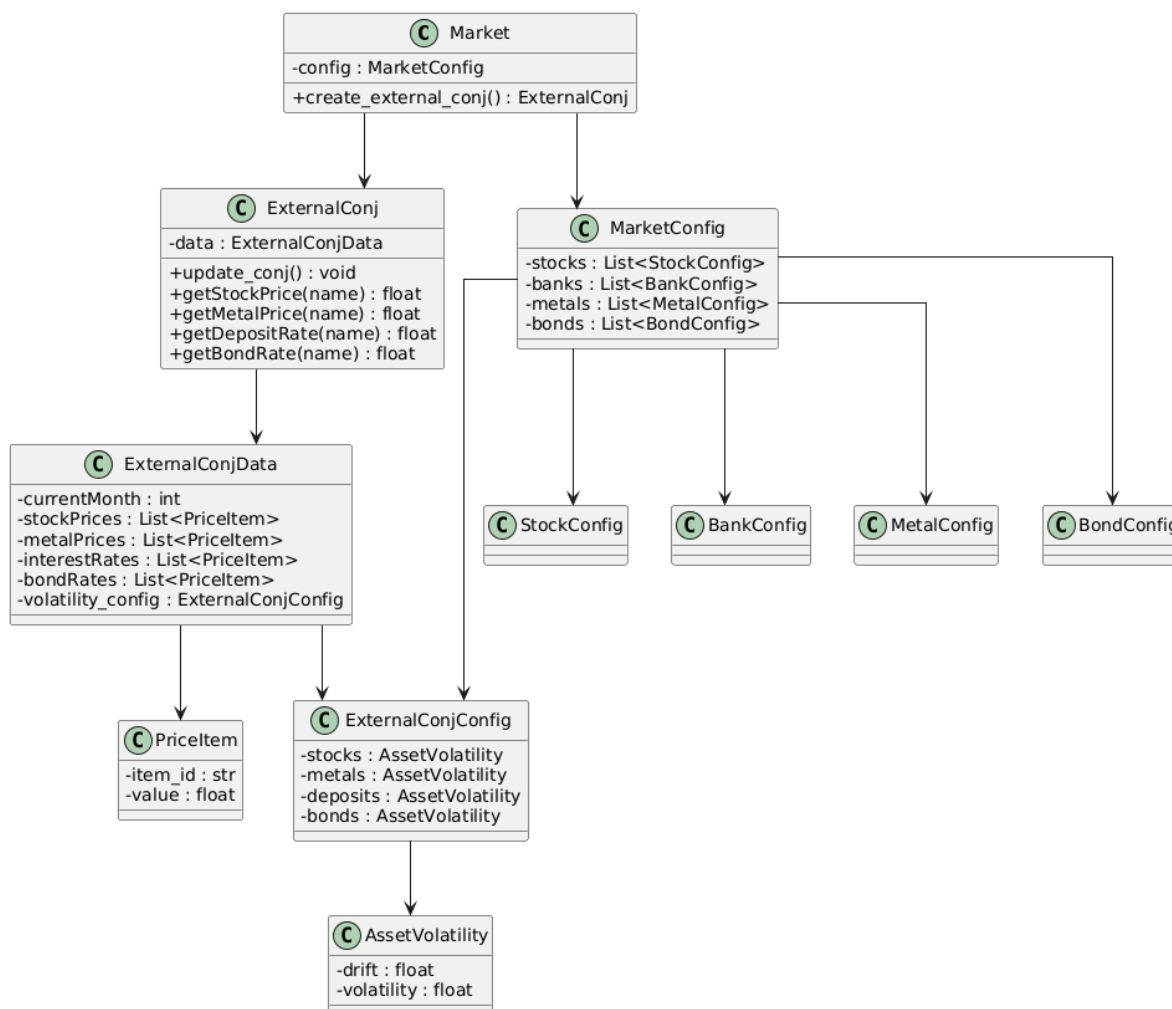


Рис. 2: Диаграмма классов: рыночная среда и внешняя конъюнктура

## 2.3 Графический интерфейс пользователя

Диаграмма иллюстрирует архитектуру пользовательского интерфейса. Показано, как главное окно (*PlayWindow*) управляет вкладками, отображающими состояние портфеля и меню покупки, а также как интерфейс взаимодействует с бизнес-логикой (*Fund*, *Market*, *ExternalConj*).

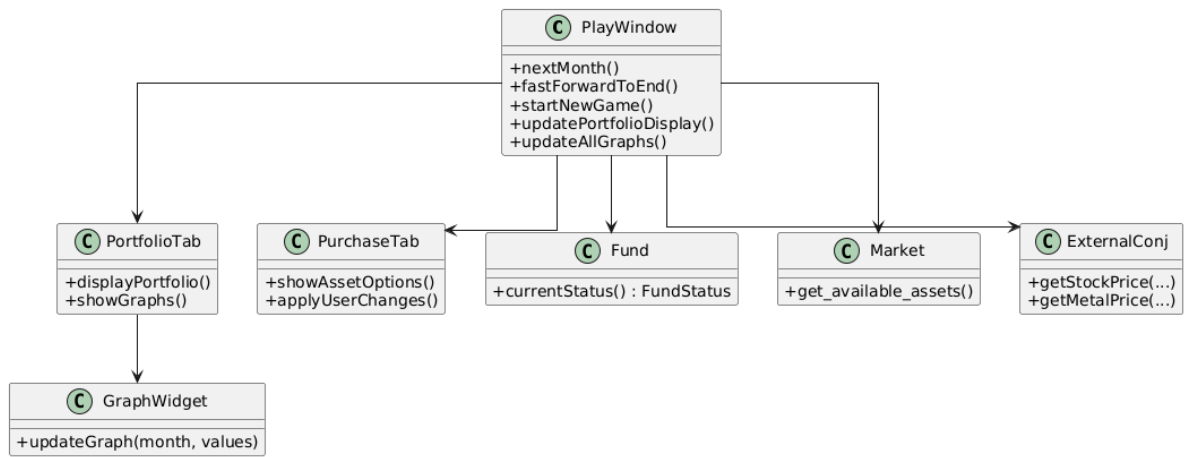


Рис. 3: Диаграмма классов: структура интерфейса пользователя



### 3 Текстовые спецификации основных классов

Ниже представлены фрагменты конструкторов ключевых классов, реализующих основную игровую логику.

Листинг 1: Класс Game

```
1 class Game:
2     def __init__(self, max_months: int, fund: Fund, external_conj:
      ExternalConj, player: Player) -> None:
3         self.current_month = 1
4         self.max_months = max_months
5         self.fund = fund
6         self.external_conj = external_conj
7         self.player = player
8         self.statistics = []
```

Листинг 2: Класс Fund

```
1 class Fund:
2     def __init__(self, capital: float, tax_rate: float, portfolio: Portfolio)
      -> None:
3         self.capital = capital
4         self.tax_rate = tax_rate
5         self.portfolio = portfolio
6         self.monthly_profit = 0.0
```

Листинг 3: Класс Player

```
1 class Player:
2     def __init__(self, player_name: str) -> None:
3         self.player_name = player_name
```

Листинг 4: Класс Portfolio

```
1 class Portfolio:
2     def __init__(self) -> None:
3         self.investments = [] # Список объектов Investment
```

Листинг 5: Класс Market

```
1 class Market:
2     def __init__(self, config_path: str):
3         self.config = self.load_market_config(config_path)
4         self.stocks = self.config.stocks
5         self.banks = self.config.banks
6         self.metals = self.config.metals
7         self.bonds = self.config.bonds
```

## 4 Диаграмма объектов

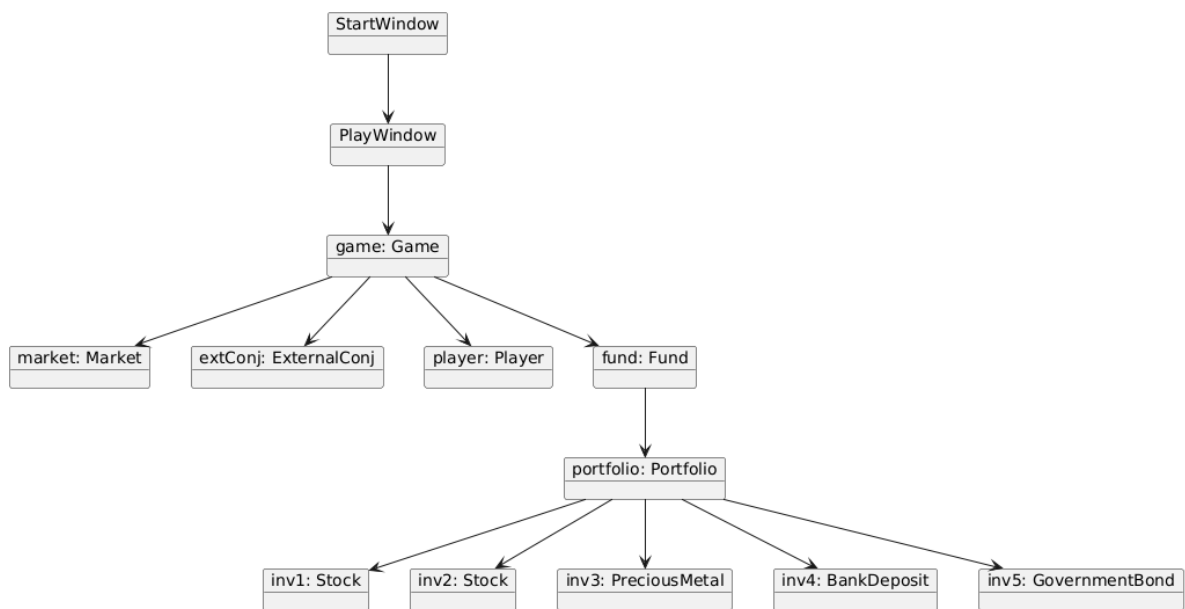


Рис. 4: Пример взаимодействия объектов в процессе моделирования

## 5 Инструментальные средства

- Язык программирования: Python 3.12
- GUI-библиотека: PyQt5
- Доп. библиотеки: matplotlib, pydantic, yaml, random
- IDE: VS Code
- Система контроля версий: Git

## 6 Файловая структура

- **main.py** — точка входа в программу.
- **core/** — реализация логики фонда, игрока, игры, моделей.
- **investments/** — классы для всех типов инвестиций.
- **trading\_market/** — управление рынком и доступными активами.
- **configs/** — YAML-файлы с настройками рынка и волатильности.
- **gui/** — интерфейс пользователя: окна, графики, вкладки.

## 7 Пользовательский интерфейс

Разработанная система обладает графическим пользовательским интерфейсом, реализованным с использованием библиотеки PyQt5. Интерфейс разделён на три основных окна: окно настройки игры, окно с меню покупки и вкладкой визуализации портфеля.

### 7.1 Окно настройки игры

На стартовом экране пользователь задаёт ключевые параметры игры: стартовый капитал, налоговую ставку, длительность, а также формирует начальный инвестиционный портфель, распределяя средства между активами.

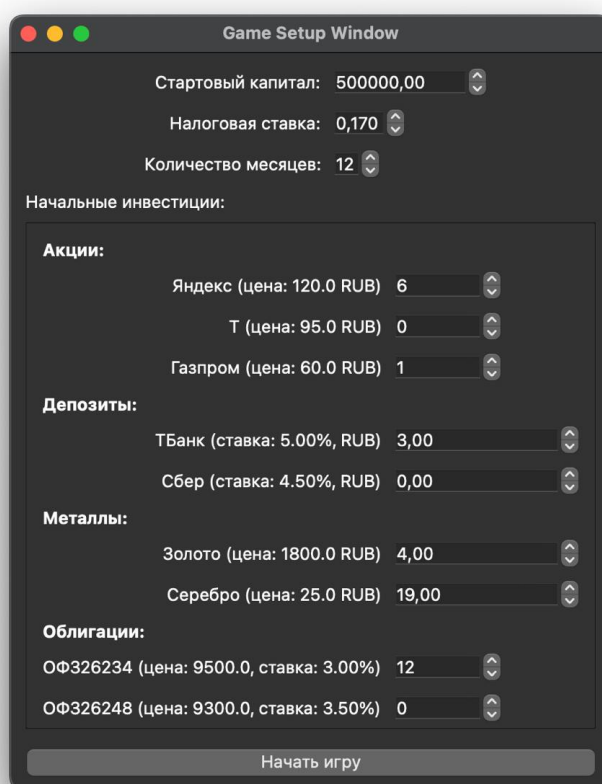


Рис. 5: Интерфейс: окно настройки игры

### 7.2 Меню управления портфелем

После запуска игры открывается главное окно управления инвестициями. Пользователь может перейти на вкладку *Меню покупки*, где доступно

изменение структуры портфеля: покупка или продажа активов по текущим рыночным условиям.

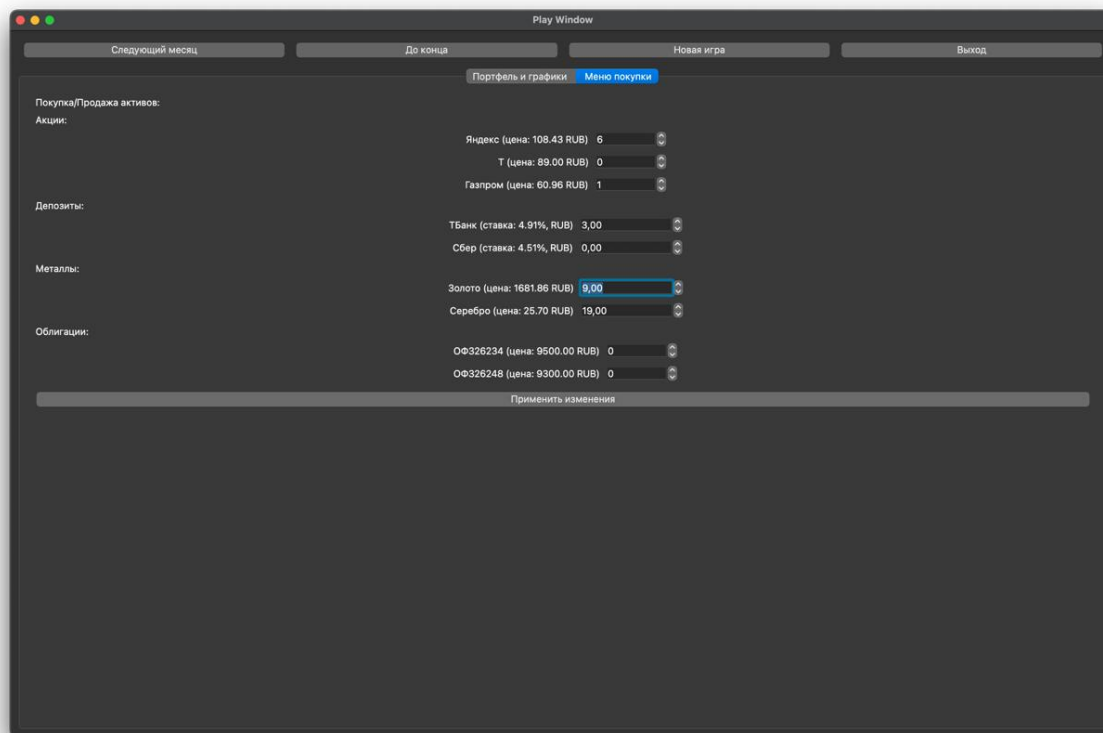


Рис. 6: Интерфейс: меню для реализации инвестиций

### 7.3 Вкладка «Портфель и графики»

На отдельной вкладке отображается текущее состояние инвестиционного портфеля, а также графики изменения стоимости портфеля и рыночных условий. Это позволяет пользователю отслеживать динамику доходности и корректировать стратегию.

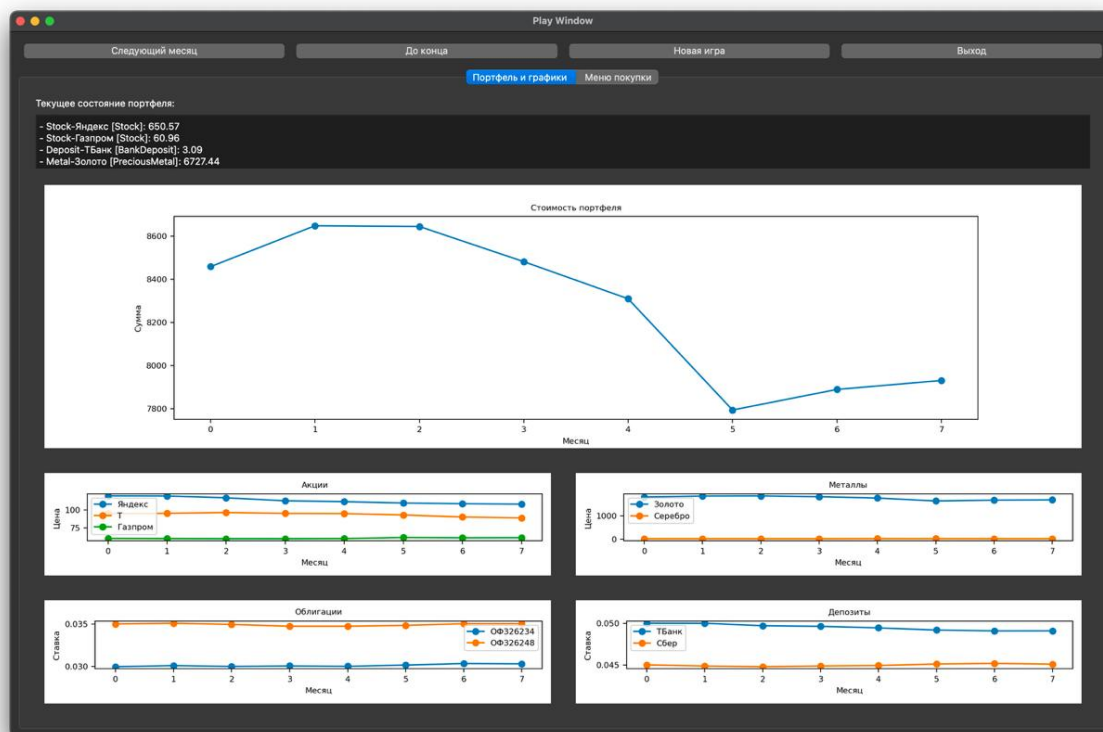


Рис. 7: Интерфейс: графическая визуализация портфеля и рыночных данных

## 7.4 Панель управления симуляцией

В верхней части главного окна располагается панель с кнопками управления ходом моделирования:

- **Следующий месяц** — выполняет симуляцию следующего временного шага и обновляет состояние портфеля и графиков.
- **До конца** — запускает симуляцию всех оставшихся месяцев.
- **Новая игра** — возвращает пользователя в окно начальной настройки и позволяет запустить симуляцию с новыми параметрами.
- **Выход** — завершает работу приложения.

Эти кнопки доступны независимо от выбранной вкладки и обеспечивают удобное управление процессом моделирования.

## 7.5 Финальная статистика

По завершении симуляции пользователь может перейти на вкладку *Финальная статистика*, где доступна подробная таблица с результатами моделирования. Статистика может быть представлена в двух режимах: *сводка по активам* и *помесячно*. Таблица содержит данные по каждому активу: его доходность, стоимость и изменения за каждый месяц.

ID актива	Месяц	Прибыль за месяц	Стоимость
1 Deposit-ТБанк	1	0.01	3.01
2 Deposit-ТБанк	2	0.01	3.02
3 Deposit-ТБанк	3	0.01	3.04
4 Deposit-ТБанк	4	0.01	3.05
5 Deposit-ТБанк	5	0.01	3.06
6 Deposit-ТБанк	6	0.01	3.08
7 Deposit-ТБанк	7	0.01	3.09
8 Deposit-ТБанк	8	0.01	3.10
9 Deposit-ТБанк	9	0.01	3.11
10 Deposit-ТБанк	10	0.01	3.12
11 Deposit-ТБанк	11	0.01	3.14
12 Deposit-ТБанк	12	0.01	3.15
13 Metal-Золото	1	178.03	7378.03
14 Metal-Золото	2	20.22	7398.25
15 Metal-Золото	3	-137.71	7260.54
16 Metal-Золото	4	-225.05	7035.50
17 Metal-Золото	5	-500.03	6535.47
18 Metal-Золото	6	147.52	6682.99
19 Metal-Золото	7	44.46	6727.44
20 Metal-Золото	8	49.94	6777.38
21 Metal-Золото	9	-226.21	6551.18
22 Metal-Золото	10	486.85	7038.03
23 Metal-Золото	11	421.35	7459.38
24 Metal-Золото	12	5.57	7464.95
25 Metal-Серебро	1	12.13	487.13
26 Metal-Серебро	2	-8.08	479.05
27 Metal-Паладий	3	0.20	479.25

Рис. 8: Интерфейс: финальная статистика по результатам моделирования