

УДК 332.33

DOI: 10.34670/AR.2023.58.87.050

## Методические основы программирования оценки эффективности управления финансовыми активами в строительных проектах

**Губаев Эдуард Александрович**

Магистрант,  
Ижевский государственный технический  
университет им. М.Т. Калашникова,  
426000, Российская Федерация, Ижевск, ул. Студенческая, 7;  
e-mail: i@egubaev.ru

**Алексеева Наталья Анатольевна**

Доктор экономических наук, профессор,  
профессор кафедры экономики и организации;  
Удмуртский государственный аграрный университет;  
профессор,  
Ижевский государственный технический  
университет им. М.Т. Калашникова,  
426000, Российская Федерация, Ижевск, ул. Студенческая, 7;  
e-mail: 497477@mail.ru

**Якушев Николай Михайлович**

Кандидат экономических наук,  
доцент кафедры промышленного и гражданского строительства,  
Ижевский государственный технический  
университет им. М.Т. Калашникова,  
426000, Российская Федерация, Ижевск, ул. Студенческая, 7;  
e-mail: nmy\_planner@mail.ru

### Аннотация

Целью исследования является демонстрация внедрения основных методов вычисления с помощью языка программирования высокого уровня Python с использованием библиотеки NumPy, управления финансовыми активами, возможностей компьютерных технологий в помощи расчетов основных показателей денежного потока, рисков и волатильности. Представлены основные идеи работы исполняемой программы (бэкенд кода) с комментариями для последующей разработки полноценного программного обеспечения с эргономичным и дружелюбным интерфейсом для пользователей. В статье представлены основы финансов и на простых примерах кода Python показаны важнейшие математические объекты и финансовые понятия. Такие фундаментальные понятия из сферы финансов можно представить и объяснить даже в статичной экономике с двумя состояниями. Базовое понимание и уже сформировавшееся в некоторой степени чутье в

финансовых и математических вопросах значительно упрощают переход к более реалистичным финансовым моделям. Рассмотрена лишь единственная библиотека NumPy, которая одна уже предоставляет огромное количество функций для обработки данных. Это лишь верхушка айсберга из всего функционала использования методов программирования для финансовых расчетов. Используя более совершенные Data Science и машинное обучение, а в перспективе и квантовые компьютеры, которые способны решить практически любые задачи, нерешаемые на любом гипотетическом вычислительном устройстве, которые мы когда-либо могли надеяться построить, мы получим возможность наиболее быстро и точно проводить финансовые расчеты.

#### Для цитирования в научных исследованиях

Губаев Э.А., Алексеева Н.А., Якушев Н.М. Методические основы программирования оценки эффективности управления финансовыми активами в строительных проектах // Экономика: вчера, сегодня, завтра. 2023. Том 13. № 9А. С. 864-875. DOI: 10.34670/AR.2023.58.87.050

#### Ключевые слова

Строительство, расчет, затраты, экономическая эффективность, риски, программирование, Python.

## Введение

В современном мире, когда экономика быстро развивается, строительство становится все более динамичным, а методы расчета финансовых активов постоянно совершенствуются, особенно с использованием цифровых технологий и языков программирования. Однако, несмотря на это, существуют недостатки общей методологии применения языков программирования в сфере финансового управления в строительстве. Труд Ива Хилпиша [Хилпиш, 2021] слишком сложен для тех, кто только начинает знакомство с теорией финансов и программированием на Python. Литература по математике для экономистов не затрагивает такие сферы как программирование [Путко, 2021], но в целом предоставляют фундаментальную базу проведения математических операций в экономике. В источниках, непосредственно рассматривающих экономику отрасли строительства, больше затрагивается математика [Павлов, 2021].

На данный момент финансы и программирование являются тесно связанными областями. В течение длительного времени финансовая теория и финансовая инженерия существовали как отдельные дисциплины. Однако в последнее время стало ясно, что навыки программирования, такие как Python и C++, стали неотъемлемой частью магистерских и других университетских программ в области финансовой инженерии. Это свидетельствует о том, что данная область стала требовать все более высоких навыков программирования [Грахов, 2023].

На фоне перечисленных выше тенденций финансы стали сферой применения четырех языков: естественный язык, финансовый язык, математический язык, язык программирования. Чтобы отлично разбираться в науке о финансах, и теоретик, и практик должны свободно владеть всеми четырьмя языками. Конечно, нельзя утверждать, что в финансовой области нет других языков, кроме английского или, например, языка программирования.

Развитие технологий способствует внедрению инноваций и повышению эффективности

финансового сектора. Как правило, проекты в данной сфере следуют концепции полного перехода на цифровые модели.

В настоящее время все больше финансовых расчетов основывается на алгоритмах, требующих больших вычислительных ресурсов. Постоянно растущая доступность данных и развитие искусственного интеллекта также определяют требования к языку программирования и технологии. Python оказался подходящим языком для удовлетворения этих потребностей и решения новых задач, связанных с развитием финансовой сферы.

Повышение эффективности управления в строительстве может привести к тому, что конкурентное преимущество будет необходимо искать в более сложных решениях, что может повлечь за собой увеличение риска и затрат, необходимых для соблюдения регуляторных требований. В 2007–2008 гг. была продемонстрирована опасность, которую могут принести новые технологии, и эта проблема также касается финансового сектора. Об этом свидетельствует обвал фондового рынка синтетических субстандартных ипотечных свопов в мае 2010 г.

Несмотря на то, что традиционные финансовые модели и теории не идеальны, они все же прошли проверку временем и до сих пор остаются актуальными. Но когда речь идет об обработке данных и принятии решений в микросекундных интервалах, все еще не хватает надежных теорий и математических моделей.

Здесь на помощь приходят современные технологии, такие как высокочастотная торговля, которая использует микроструктурные элементы биржи. Хотя отсутствие надежных финансовых моделей может быть недостатком, современные технологии помогают компенсировать этот недостаток, делая высокочастотную торговлю более эффективной.

В течение последних лет в финансовой области наблюдается повышенный интерес к анализу финансовых данных. Это связано с развитием технологий, позволяющих обрабатывать большие объемы данных и проводить сложные расчеты быстрее. Становится все более актуальным анализ данных в режиме реального времени. Данный анализ подразумевает использование современных технологий и программного обеспечения для анализа финансовых данных и принятия решений в режиме реального времени. Например, оценка влияния изменения стоимости продукта на объем продаж или корректировка оценки кредита для сложных портфелей производных инструментов крупного инвестиционного банка.

Большие объемы данных требуют постоянного увеличения вычислительной мощности и хранилищ данных, а также изменения процессов управления рисками, которые теперь должны быть приняты в течение рабочего дня. Здесь снова прослеживается связь между развитием технологий и практикой ведения бизнеса. С одной стороны, существует постоянная потребность в повышении скорости и точности аналитических решений за счет внедрения современных технологий. А с другой стороны, появление новых технологий делает возможными новые аналитические подходы, которые всего несколько лет назад считались нереалистичными, исходя из бюджетных соображений.

Одной из ключевых тенденций в сфере финансового анализа стало внедрение параллельных архитектур многоядерных вычислений на стороне центрального процессора и массово-параллельных архитектур на основе графических процессоров. Современные графические процессоры содержат тысячи вычислительных ядер, что заставляет переосмысливать алгоритмы параллельных вычислений. Чтобы извлечь преимущества из новых аппаратных платформ, пользователям приходится осваивать новые языки и парадигмы программирования.

Финансовая индустрия постоянно создает различные продукты с показателями риска и

доходности, в том числе это продукты строительства, наиболее стабильные в плане доходности при низкой степени риска. Хотя кризис 2008 г. в США может свидетельствовать об обратном, но он был вызван не спецификой строительной промышленности, а цепной реакцией по невыплатам по ипотечным долговым обязательствам и перегретым рынком сбыта недвижимости.

### Основная часть

В рамках концепции экономики двух состояний рассмотрим концепцию и основные принципы встраивания методов программирования для финансовых расчетов объекта.

Модель экономики объекта имеет два релевантных момента времени и два неопределенных будущих состояния, но, несмотря на свою простоту, позволяет объяснить важные финансовые понятия и концепции. Она также помогает понять некоторые важные достижения, сделанные в области финансовой экономики, например, фундаментальную теорему ценообразования финансовых активов. Более подробно об этой теореме можно узнать из статей Харрисона, Крепса [Harrison, Kreps, 1979] и Харрисона с Плиски [Harrison, Pliska, 1981], опубликованных в 1979 и 1981 гг. соответственно. Финансовая модель с двумя состояниями может помочь формально представить абстрактные финансовые понятия.

Рассмотрим инвестиционный проект, который требует вложения, например, 38,3 денежной единицы сегодня и возвращает 45,4 денежной единицы через 3 года. Инвестиции обычно считаются оттоком денежных средств и часто являются отрицательными числами,  $c \in \mathbb{R}_{<0}$ , или в нашем случае  $c = -38,3$ . Возврат инвестированных средств – это приток и, следовательно, положительное число,  $c \in \mathbb{R}_{\geq 0}$ , или  $c = +45,4$ .

Для обозначения моментов времени, когда происходят отток и приток денежных средств, используются отметки времени. В нашем случае это  $c_{t=0} = -38,3$  и  $c_{t=1} = 45,4$ , или сокращенно  $c_0 = -38,3$  и  $c_1 = 45,4$ .

Денежный поток сейчас и денежный поток через год математически моделируются как упорядоченная пара, или кортеж, объединяющий эту пару в один объект:  $c \in \mathbb{R}^2$ , где  $c = (c_0, c_1)$  и  $c_0, c_1 \in \mathbb{R}$ .

Листинг программы:

```
In [1]: c0 = -38.3 {Определение оттока денежных средств сегодня};
In [2]: c1 = 45.4 {Определение оттока денежных средств ровно через год};
In [3]: c = (c0, c1) {Определение объекта tuple через c};
In [4]: c;
Out [4]: (-38.3, 45.4) {Вывод пары денежных потоков};
In [5]: type (c);
Out [5]: tuple {Поиск и вывод типа объекта c};
In [6]: c [0];
Out [6]: -38.3 {Вывод первого элемента объекта c};
In [7]: c [1];
Out [7]: 45.4 {Вывод второго элемента объекта c}.
```

Стоит пояснить, что в фигурных скобках `{ }` написаны комментарии к коду выполняемой программы в интерпретаторе языка, они не участвуют в выполнении программы и нужны лишь для пояснения действия того или иного алгоритма.

Доходность,  $R \in \mathbb{R}$ , инвестиционного проекта с денежными потоками  $c = (c_0, c_1) = (-38,3,$

45,4) равна сумме этих денежных потоков, или  $R = c_0 + c_1 = (-38,3, 45,4) = 7,1$ . Ставка доходности,  $r \in \mathbb{R}$ , — это доход  $R$ , деленный на инвестиционные затраты сегодня по модулю  $|c_0|$ :

$$r = \frac{R}{|c_0|} = \frac{-38,3+45,4}{38,3} = 0,185. \quad (1)$$

В Python данное вычисление сводится к следующим арифметическим операциям.

Листинг программы:

In [8]:  $c = (-38.3, 45.4)$  {Определение пары денежных потоков через объект tuple.};

In [9]:  $R = \text{sum}(c)$  {Вычисление дохода  $R$  путем сложения всех элементов  $c$  и};

In [10]:  $R$ ;

Out [10]: 7.1 {Вывод результата};

In [11]:  $r = R / \text{abs}(c[0])$  {Вычисление ставки доходности  $r$  через функцию  $\text{abs}(x)$ , возвращающую абсолютное значение  $x$ ...};

In [12]:  $r$ ;

Out [12]: 0.185 {Вывод результата}.

Через год денежный поток может измениться из-за процентов, которые начисляются на заемные средства или должны быть выплачены за использование чужих денег. Проценты представляют собой стоимость управления деньгами другими агентами.

Предположим, сегодня агент размещает в банке депозит  $c_0 = -10$  денежных единиц. Согласно депозитному договору, через год он получит от банка  $c_1 = 11$  денежных единиц. Процент по вкладу,  $I \in \mathbb{R}$ , равен  $I = c_0 + c_1 = -10 + 11 = 1$ .

Следовательно, процентная ставка,  $I \in \mathbb{R}$ , составляет  $i = \frac{I}{|c_0|} = 0,1$ .

В дальнейших вычислениях можно исходить из того, что процентная ставка как по кредиту, так и по депозиту одинакова и постоянна.

Доступность кредитования и депонирования влекут за собой альтернативные издержки при вложении денег в инвестиционный проект. К примеру, денежный поток  $c_1 = 45,4$  через год нельзя напрямую сравнивать по стоимости с денежным потоком  $c_0 = 45,4$  сегодня, поскольку на деньгах, не вложенных в проект, можно заработать проценты.

Для правильного сравнения этих денежных потоков необходимо рассчитать приведенную стоимость через дисконтирование с использованием фиксированной процентной ставки в экономике. Дисконтирование можно смоделировать как функцию  $D: \mathbb{R} \rightarrow \mathbb{R}$ ,  $c_1 \rightarrow D(c_1)$ , которая сопоставляет одно вещественное число (денежный поток через год) с другим вещественным числом (денежным потоком сегодня). С учетом процентной ставки  $i = 0,1$  приведенная стоимость составит:

$$c_0 = D(c_1) = \frac{c_1}{1+i} = \frac{45,4}{1+0,1} = 41,3. \quad (2)$$

Представление дисконтирования и подобных ему математических функций реализовано в Python довольно просто.

Листинг программы:

In [13]:  $i = 0.1$  {Фиксирование процентной ставки  $i$ };

In [14]:  $\text{def } D(c1):$  {Определение функции через оператор  $\text{def}$ , где  $D$  — название функции,  $c_1$  — имя параметра};

$\text{return } c1 / (1 + i)$  {Вывод приведенной стоимости через оператор  $\text{return}$ .};

In [15]: D (45.4);

Out [15]: 45.4 Расчет приведенной стоимости 45.4;

In [16]: D (41.3);

Out [16]: 41 {Расчет приведенной стоимости}.

Как агенту решить, стоит ли реализовывать инвестиционный проект? Одним из критериев является чистая приведенная стоимость. Чистая приведенная стоимость,  $NPV \in \mathbb{R}$ , представляет собой сумму оттока денежных средств сегодня и приведенной стоимости притока денежных средств через год:

$$NPV(c) = c_0 + D(c_1). \quad (3)$$

Расчет чистой приведенной стоимости представляет собой функцию  $NPV: \mathbb{R} \rightarrow \mathbb{R}^2$ , сопоставляющую кортеж денежного потока с вещественным числом.

Если чистая приведенная стоимость проекта имеет положительное значение, то проект следует реализовать, если отрицательное – нет, поскольку в таком случае альтернатива в виде банковского депозита более выгодна.

Рассмотрим инвестиционный проект с денежными потоками  $c_A = (-38,3, 41,3)$ . Чистая приведенная стоимость  $NPV(c_A) = -10,5 + D(12,1) = -38,3 + 41,3 = 3$ , то есть данный проект выгоден.

Эти примеры легко реализуются в Python через соответствующую функцию. Листинг программы:

In [35]: def NPV(c);

return c [0] + D (c [1]);

In [36]: cA = (-38.3, 41.3);

In [38]: NPV (cA);

Out [38]: 3.

В силу того, что чистая приведенная стоимость проекта имеет положительное значение, считаем, что проект подходит для реализации.

Приток денежных средств от инвестиционного проекта через год в целом имеет неопределенный характер, так как в реальных условиях на него могут повлиять множество факторов: конкуренция, появление новых технологий, рост экономики, погода, проблемы в ходе реализации проекта и т.д. В модельной экономике понятие состояний экономики через год включает в себя влияние всех соответствующих факторов.

Предположим, что через год экономика будет находиться в одном из двух состояний,  $u$  и  $d1$ , или, другими словами, в состоянии роста (хорошем) или спада (плохом). Тогда денежный поток проекта через год  $c1$  можно выразить вектором:

$$c_1 \in \mathbb{R}^2, \quad (4)$$

с двумя различными значениями:

$$c_1^u, c_1^d \in \mathbb{R}^2, \quad (5)$$

представляющими соответствующие денежные потоки для каждого состояния экономики. Формально данный денежный поток представлен так называемым вектором-столбцом:

$$c_1 = \begin{pmatrix} c_1^u \\ c_1^d \end{pmatrix}. \quad (6)$$

Математическими операциями над такими векторами являются скалярное умножение и сложение, например:

$$\alpha \cdot c_1 + \beta = \alpha \cdot \begin{pmatrix} c_1^u \\ c_1^d \end{pmatrix} + \beta = \begin{pmatrix} \alpha \cdot c_1^u + \beta \\ \alpha \cdot c_1^d + \beta \end{pmatrix}. \quad (7)$$

Еще одной важной операцией над векторами является создание линейных комбинаций векторов. Рассмотрим два вектора,  $c_1, d_1 \in \mathbb{R}^2$ . Их линейная комбинация составляется следующим образом:

$$\alpha \cdot c_1 + \beta \cdot d_1 = \begin{pmatrix} \alpha \cdot c_1^u + \beta d_1^u \\ \alpha \cdot c_1^d + \beta d_1^d \end{pmatrix}. \quad (8)$$

Следует отметить, что здесь и ранее предполагается  $\alpha, \beta \in \mathbb{R}$ .

Наиболее распространенным способом моделирования векторов и матриц в Python является библиотека NumPy. Для следующего примера кода рассмотрим инвестиционный проект с денежными потоками  $c_0 = -38,3$  и  $c_1 = (45,4)^T$ , где T обозначает транспонирование вектора – преобразование горизонтального вектора (вектора-строки) в вертикальный вектор (вектор-столбец).

Основной класс, используемый для моделирования векторов, – ndarray (n-мерный массив).

Листинг программы:

```
In [39]: import numpy as np {подключение библиотеки numpy};
In [40]: c0 = -38.3 {Отток денежных средств сегодня};
In [41]: c1 = np.array ((45.4, 5)) {Приток денежных средств неопределенного размера через
год: одно- мерные объекты ndarray не различают строки (горизонталь) и столбцы (вертикаль)};
In [42]: type (c1) {Поиск и вывод типа c1};
Out [42]: numpy.ndarray;
In [43]: c1 {Вывод вектора денежного потока};
Out [43]: array ([45.4, 5]);
In [44]: c = (c0, c1) {Вывод вектора денежного потока};
In [45]: c {Объект tuple, как и объект list, может содержать другие сложные структуры
данных};
Out [45]: (-38.3, array ([45.4, 5. ]));
In [46]: 1.5 * c1 + 2 {Линейное преобразование вектора путем скалярного умножения и
сложения (векторная числовая операция и трансляция)};
Out [46]: array ([70.1, 9.5]);
In [47]: c1 + 1.5 * np.array ((10, 4)) {Линейная комбинация двух объектов ndarray (векторов)};
Out [47]: array ([60.4, 11]).
```

Несмотря на то, что в Python имеется большое количество базовых структур данных, пользователям предлагаются еще более широкие возможности, реализуемые в дополнительных библиотеках. Познакомимся с библиотекой NumPy, в которой содержится класс многомерных массивов, позволяющий обрабатывать однородные и неоднородные структуры данных. Другое преимущество библиотеки – поддержка векторизации кода.

Часто неявно предполагается равная вероятность двух состояний экономики, то есть при бесконечном многократном повторении рассматриваемого эксперимента в экономике наблюдается, что в одной половине случаев реализуется состояние  $u$ , а в другой –  $d$ .

При определении вероятности на основе частоты вероятность реализации состояния рассчитывается как соотношение частоты наблюдения состояния к общему числу испытаний. Если состояние  $u$  наблюдается в 30 экспериментах из 50, то вероятность  $p \in [0, 1]$  при  $0 \leq p \leq 1$  составляет  $p = 30 / 50 = 0,6$ , или 60 %.

В моделировании предполагается, что вероятность реализации всех возможных состояний задана априори. Ее называют также объективной вероятностью или физической вероятностью.

Совокупность вероятностей физически возможных событий образует вероятностную меру с функцией  $P: \wp(\{u, d\}) \rightarrow \mathbb{R}_{\geq 0}$ , отображающей на единичном интервале все элементы булеана (показательного множества)  $\{u, d\}$  при  $\wp(\emptyset\{u\}, \{d\}, \{u, d\})$ . В данном случае булеан представляет собой все физически возможные события (множество всех подмножеств).

В нашем случае множество  $\{u, d\}$  является пространством состояний и обозначается символом  $\Omega$ . Тройка  $(\Omega, \wp(\Omega), P)$  называется вероятностным пространством.

Функция  $P$ , представляющая вероятностную меру, должна удовлетворять трем условиям:

$$P(\emptyset) = 0;$$

$$0 \leq P(\omega), \omega \in \Omega \leq 1;$$

$$P(\Omega) = P(u) + P(d) = 1.$$

Первое условие предполагает обязательную реализацию по крайней мере одного из состояний. Второе означает, что вероятность реализации того или иного состояния должна находиться в диапазоне от 0 до 1. Под третьим условием подразумевается, что все вероятности составляют в сумме 1.

В простой модельной экономике с двумя состояниями вероятность можно выразить как  $p \equiv P(u)$  и на основе третьего условия посчитать  $P(d) = 1 - p$ . Таким образом, вероятностная мера  $P$  определяется через значение вероятности  $p$ .

При наличии полностью определенной вероятностной меры модельная экономика обычно называется экономикой в условиях риска, а при ее отсутствии – экономикой в условиях неопределенности.

Неопределенность в финансовой экономике может принимать различные формы. Риск в целом относится к ситуации, когда известно (предполагается, что известно) полное распределение вероятностей по будущим состояниям экономики. Неопределенность относится к ситуациям, в которых такое распределение неизвестно. Практически всегда финансы опираются на модельную экономику в условиях риска, однако существует ряд исследований, посвященных финансовым проблемам именно в условиях неопределенности.

Как правило, вероятностная мера моделируется через вектор и объект ndarray. По крайней мере это возможно в условиях дискретного пространства состояний с конечным числом элементов. Листинг программы:

```
In [56]: p = 0.4;
```

```
In [57]: 1 - p;
```

```
Out [57]: 0.6;
```

```
In [58]: P = np.array((p, 1-p));
```

```
In [59]: P;
```

```
Out [59]: array ([0.4, 0.6]).
```

В финансах главная пара понятий – это риск и ожидаемая доходность. Риск можно измерить

множеством способов, а волатильность, измеряемая стандартным отклонением ставок доходности, вероятно, является наиболее распространенным финансовым показателем. В нашем случае дисперсия ставок доходности финансового актива определяется как:

$$\sigma^2(r) = E^P((r - \mu)^2) = \left( \left( \begin{matrix} p \\ 1-p \end{matrix} \right), \left( \begin{matrix} (r^u - \mu)^2 \\ (r^d - \mu)^2 \end{matrix} \right) \right), \quad (9)$$

где  $r^\omega \equiv (S_1^\omega - S_0)/S_0$ ,  $\omega \in \Omega$ .

Поскольку волатильность определяется как стандартное (среднеквадратическое) отклонение ставок доходности, она представляет собой квадратный корень из дисперсии:

$$\sigma(r) = \sqrt{\sigma^2(r)}. \quad (10)$$

Далее приведены функции Python, моделирующие эти два показателя риска, а также вспомогательная функция для расчета вектора ставок доходности.

Листинг программы:

```
In [68]: def r(x0, x1);
return (x1 - x0) / x0 {Векторное вычисление ставки доходности};
In [69]: r(S0, S1) {Применение функции к ранее определенному финансовому активу};
Out [69]: array ([ 1., -0.5]);
In [70]: mu = np.dot (P, r (S0, S1)) {Ожидаемая ставка доходности через скалярное
произведение и...};
In [71]: mu {... Вывод результата};
Out [71]: 0.10000000000000003;
In [72]: def sigma2(P, r, mu);
return np.dot (P, (r - mu) ** 2) {Определение дисперсии ставок доходности};
In [73]: sigma2(P, r (S0, S1), mu) {Применение функции к вектору ставок доходности};
Out [73]: 0.54;
In [74]: def sigma (P, r, mu);
return np.sqrt (np.dot(P, (r - mu) ** 2)) {Определение волатильности};
In [75]: sigma (P, r (S0, S1), mu) {И ее применение к вектору ставок доходности};
Out [75]: 0.7348469228349535.
```

Финансы как прикладная математическая дисциплина в значительной степени опираются на линейную алгебру и теорию вероятностей. Для дискретной модели экономики оба эти раздела прекрасно обрабатываются с помощью библиотеки NumPy (в частности, посредством объекта ndarray). Помимо моделирования, NumPy подходит для обработки информации, расчетов, оптимизации, визуализации и др.

## Заключение

Таким образом, представлены основы финансов и на простых примерах кода Python показаны важнейшие математические объекты и финансовые понятия. Такие фундаментальные понятия из сферы финансов можно представить и объяснить даже в статичной экономике с двумя состояниями. Базовое понимание и уже сформировавшееся в некоторой степени чутье в финансовых и математических вопросах значительно упрощают переход к более реалистичным

финансовым моделям. Рассмотрена лишь единственная библиотека NumPy, которая одна уже предоставляет огромное количество функций для обработки данных. Это лишь верхушка айсберга из всего функционала использования методов программирования для финансовых расчетов. Используя более совершенные Data Science и машинное обучение, а в перспективе и квантовые компьютеры, которые способны решить практически любые задачи, нерешаемые на любом гипотетическом вычислительном устройстве, которые мы когда-либо могли надеяться построить, мы получим возможность наиболее быстро и точно проводить финансовые расчеты.

Благодаря применению языков программирования на порядки возрастает скорость обработки входящих данных и предоставления результатов финансовых расчетов, исключается ручной расчет, снижается вероятность возникновения ошибки, а также минимизируется человеческий фактор. Существует возможность на основе алгоритмов разработать программное обеспечение в виде приложения с дружелюбным интерфейсом (эту задачу выполняет frontend разработчик совместно с отделом дизайна) для максимального упрощения пользователям уровня user взаимодействия с ПО. Это позволяет, не нагружая его громоздкими и не дружелюбными таблицами Excel, рассчитывать данные буквально «на лету» для любых объектов, и не только строительства.

## Библиография

1. Грахов В.П. Разработка инновационной модели финансирования инвестиционных проектов в строительстве за счет средств частных инвесторов // Жилищные стратегии. 2023. Т. 10. № 1. С. 11-32.
2. Кеннеди Б. Основы Python для Data Science. СПб.: Питер, 2023. 272 с.
3. Павлов А.С. Экономика строительства. В 2 частях. Ч. 1: учебник и практикум для среднего профессионального образования. М.: Юрайт, 2021. 337 с.
4. Путко Б.А. Математика для экономистов: от арифметики до эконометрики. М.: Юрайт, 2021. 760 с.
5. Хилпиш И. Python для финансовых расчетов. Киев: Диалектика, 2021. 800 с.
6. Harrison M., Kreps D. Martingales and Arbitrage in Multiperiod Securities Markets // Journal of Economic Theory. 1979. № 20. P. 381-408.
7. Harrison M., Pliska S. Martingales and Stochastic Integrals in the Theory of Continuous Trading // Stochastic Processes and their Applications. 1981. № 11. P. 215-260.
8. Kelly J., Male S., Graham D. Value management of construction projects. – John Wiley & Sons, 2014.
9. Love P. E. D., Matthews J. The ‘how’ of benefits management for digital technology: From engineering to asset management // Automation in Construction. – 2019. – Т. 107. – С. 102930.
10. Vanier D. J. D. Why industry needs asset management tools // Journal of computing in civil engineering. – 2001. – Т. 15. – № 1. – С. 35-43.

## Methodological basis for evaluation programming efficiency of financial asset management in construction projects

**Eduard A. Gubaev**

Master's Student,  
Izhevsk State Technical University,  
426000, 7, Studencheskaya str., Izhevsk, Russian Federation;  
e-mail: i@gubaev.ru

**Natal'ya A. Alekseeva**

Doctor of Economics,  
Professor,  
Professor of the Department of Economics  
and Organization of the Agro-industrial complex,  
Udmurt State Agrarian University;  
Professor of Izhevsk State Technical University,  
426000, 7, Studencheskaya str., Izhevsk, Russian Federation;  
e-mail: 497477@mail.ru

**Nikolai M. Yakushev**

PhD in Economics,  
Associate Professor of the Department of Industrial and Civil Construction,  
Izhevsk State Technical University,  
426000, 7, Studencheskaya str., Izhevsk, Russian Federation;  
e-mail: nmy\_planner@mail.ru

**Abstract**

The purpose of the study is to demonstrate the implementation of basic calculation methods using the Python high-level programming language using the NumPy library, financial asset management, and the capabilities of computer technology to help calculate basic indicators of cash flow, risk and volatility. The basic ideas of the executable program (backend code) are presented with comments for the subsequent development of full-fledged software with an ergonomic and user-friendly interface. The article introduces the basics of finance and demonstrates the most important mathematical objects and financial concepts using simple Python code examples. Such fundamental concepts from finance can be represented and explained even in a static two-state economy. Having a basic understanding and some sense of financial and mathematical understanding makes the transition to more realistic financial models much easier. Only a single library, NumPy, is considered, which alone already provides a huge number of functions for data processing. This is just the tip of the iceberg of all the functionality of using programming methods for financial calculations. Using more advanced Data Science and machine learning, and eventually quantum computers that can solve virtually any problem that would be impossible to solve on any hypothetical computing device we could ever hope to build, we will be able to make financial calculations more quickly and accurately.

**For citation**

Gubaev E.A., Alekseeva N.A., Yakushev N.M. (2023) Metodicheskie osnovy programmirovaniya otsenki effektivnosti upravleniya finansovymi aktivami v stroitel'nykh proektakh [Methodological basis for evaluation programming efficiency of financial asset management in construction projects]. *Ekonomika: vchera, segodnya, zavtra* [Economics: Yesterday, Today and Tomorrow], 13 (9A), pp. 864-875. DOI: 10.34670/AR.2023.58.87.050

**Keywords**

Construction, calculation, costs, economic efficiency, risks, programming, Python.

---

---

## References

1. Grakhov V.P. (2023) Razrabotka innovatsionnoi modeli finansirovaniya investitsionnykh proektov v stroitel'stve za schet sredstv chastnykh investorov [Development of an innovative model for financing investment projects in construction at the expense of private investors]. *Zhilishchnye strategii* [Housing strategies], 10, 1, pp. 11-32.
2. Harrison M., Kreps D. (1979) Martingales and Arbitrage in Multiperiod Securities Markets. *Journal of Economic Theory*, 20, pp. 381-408.
3. Harrison M., Pliska S. (1981) Martingales and Stochastic Integrals in the Theory of Continuous Trading. *Stochastic Processes and their Applications*, 11, pp. 215-260.
4. Hilpisch Y. (2021) *Python dlya finansovykh raschetov* [Financial Theory with Python. A Gentle Introduction]. Kiev: Dialektika Publ.
5. Kennedy B. (2023) *Osnovy Python dlya Data Science* [Foundational Python for Data Science]. St. Petersburg: Piter Publ.
6. Pavlov A.S. (2021) *Ekonomika stroitel'stva. V 2 chastyakh. Ch. 1: uchebnik i praktikum dlya srednego professional'nogo obrazovaniya* [Economics of construction. In 2 parts. Part 1: textbook and workshop for secondary vocational education]. Moscow: Yurait Publ.
7. Putko B.A. (2021) *Matematika dlya ekonomistov: ot arifmetiki do ekonometriki* [Mathematics for economists: from arithmetic to econometrics]. Moscow: Yurait Publ.
8. Kelly, J., Male, S., & Graham, D. (2014). Value management of construction projects. John Wiley & Sons.
9. Love, P. E., & Matthews, J. (2019). The 'how' of benefits management for digital technology: From engineering to asset management. *Automation in Construction*, 107, 102930.
10. Vanier, D. D. (2001). Why industry needs asset management tools. *Journal of computing in civil engineering*, 15(1), 35-43.