
Краткое содержание

Введение	11
Глава 1. Финансы и Python.....	18
Глава 2. Экономика с двумя состояниями	36
Глава 3. Экономика с тремя состояниями.....	80
Глава 4. Оптимальность и равновесие.....	103
Глава 5. Статическая экономика.....	138
Глава 6. Динамическая экономика.....	169
Глава 7. Что дальше?	192
Об авторе	203
Иллюстрация на обложке	204

Оглавление

Введение	11
Почему именно эта книга?	11
Целевая аудитория	13
Краткое описание книги	14
Условные обозначения	15
Примеры кода, использованные в книге	16
Благодарности	17
От издательства.....	17
Глава 1. Финансы и Python	18
Краткая история финансов	19
Главные тенденции в области финансов	20
Четырехязычная сфера.....	22
Структура книги	22
Вводная информация о Python.....	26
Резюме.....	33
Справочные материалы	34
Глава 2. Экономика с двумя состояниями	36
Экономика	38
Реальные активы.....	38
Агенты	38
Время.....	38
Деньги	39

Денежный поток	40
Доходность	42
Проценты.....	43
Приведенная стоимость.....	44
Чистая приведенная стоимость.....	45
Неопределенность	46
Финансовые активы	48
Риск	49
Вероятностная мера.....	50
Математическое ожидание.....	51
Ожидаемая доходность	52
Волатильность	53
Условные требования.....	55
Репликация.....	57
Арбитражное ценообразование.....	61
Полнота рынка.....	63
Ценные бумаги Эрроу — Дебре	67
Ценообразование по мартингалу	69
Первая фундаментальная теорема ценообразования финансовых активов	70
Ценообразование через математическое ожидание	71
Вторая фундаментальная теорема ценообразования финансовых активов	72
Портфель Марковица	73
Резюме	78
Справочные материалы	78
Глава 3. Экономика с тремя состояниями.....	80
Неопределенность	82
Финансовые активы	82
Достижимые условные требования	82

Ценообразование по мартингалу	86
Мартингальные меры	86
Риск-нейтральное ценообразование	88
Суперрепликация	89
Аппроксимирующая репликация	92
Линия рынка капитала	94
Модель ценообразования капитальных активов	97
Резюме	102
Справочные материалы	102
Глава 4. Оптимальность и равновесие	103
Максимизация полезности	105
Кривые безразличия	107
Условия функции полезности	108
Логарифмическая полезность	109
Аддитивная полезность относительно времени	110
Ожидаемая полезность	113
Оптимальный инвестиционный портфель	115
Аддитивная ожидаемая полезность относительно времени	118
Ценообразование в условиях полного рынка	119
Арбитражное ценообразование	121
Ценообразование по мартингалу	122
Безрисковая процентная ставка	122
Пример в числовом виде (часть 1)	123
Ценообразование в условиях неполного рынка	126
Мартингальные меры	128
Ценообразование в условиях равновесия	130
Пример в числовом виде (часть 2)	132
Резюме	136
Справочные материалы	136

Глава 5. Статическая экономика.....	138
Неопределенность	140
Случайные величины.....	141
Примеры в числовом виде.....	141
Финансовые активы	143
Условные требования.....	146
Полнота рынка.....	147
Фундаментальные теоремы ценообразования финансовых активов.....	151
Модель ценообразования опционов Блэка — Шоулза — Мертона	155
Полнота модельной экономики в системе Блэка — Шоулза — Мертона	159
Модель ценообразования опционов со скачкообразной диффузией Мертона.....	161
Ценообразование через репрезентативного агента.....	165
Резюме	167
Справочные материалы	167
Глава 6. Динамическая экономика.....	169
Биномиальное ценообразование опционов.....	171
Моделирование и оценка посредством циклов Python	173
Моделирование и оценка посредством векторизованного кода	177
Сравнение скорости работы	180
Модель ценообразования опционов Блэка — Шоулза — Мертона	183
Моделирование ценовой траектории акций методом Монте-Карло.....	183
Оценка европейского пут-опциона методом Монте-Карло	187
Оценка американского пут-опциона методом Монте-Карло	188
Резюме	190
Справочные материалы	190

Глава 7. Что дальше?	192
Математика	192
Теория финансов	193
Программирование на языке Python.....	196
Python для финансовых расчетов.....	197
Наука о финансовых данных	197
Алгоритмическая торговля.....	198
Финансовая инженерия	199
Искусственный интеллект	200
Другие ресурсы	201
Послесловие	202
Об авторе	203
Иллюстрация на обложке	204

Введение

Python быстро становится языком науки о данных, машинного обучения и обработки естественного языка. Его использование постоянно открывает источники инноваций. У Python есть многочисленное сообщество разработчиков приложений с открытым исходным кодом, что позволяет быстро внедрять и адаптировать передовые технологии¹.

Киндман и Тейлор (2021)

Почему именно эта книга?

Эта книга обучает финансам и языку программирования Python (<http://python.org/>) с нуля. Сейчас финансы и программирование — тесно переплетенные дисциплины, а Python — один из наиболее часто используемых в финансовой отрасли языков программирования. Здесь комплексно изложены основы математики, финансов и программирования в понятном для обычных людей виде. Долгое время теория финансов и финансовая инженерия были отдельными дисциплинами. Однако то, что программирование (например, на Python и C++) стало неотъемлемой частью магистратуры по финансовой инженерии и подобных университетских программ, доказывает, насколько важным стал этот навык в данной области.

¹ Kindman A., Taylor T. Why We Rewrote Our USD30 Billion Asset Management Platform in Python. March 29, 2021. <https://oreil.ly/GghS6>.

Платформы для онлайн-торговли, программное обеспечение с открытым исходным кодом и финансовая информация, находящаяся в свободном доступе, значительно понизили, а то и полностью ликвидировали порог входа на мировые финансовые рынки. Теперь всего за несколько часов обычный человек с ограниченным бюджетом может начать заниматься алгоритмической торговлей. Студенты и преподаватели финансовых дисциплин, немного знающие программирование, могут применить последние инновации в области машинного и глубокого обучения к финансовым данным, пользуясь только ноутбуками, которые у них всегда с собой. Что касается технических средств, то облачные провайдеры с почасовой оплатой и практически неограниченной масштабируемостью за очень небольшую цену предоставляют возможность быстро и качественно вычислять и обрабатывать данные. Сегодня профессиональное финансовое образование лишь частично соответствует этим технологическим тенденциям.

Тем не менее все еще довольно часто *основы математики, теория финансов и основы программирования* преподаются независимо друг от друга и только в самом конце обучения — в комплексе с *финансовой инженерией*. В этой книге используется другой подход: финансовые концепции и методы программирования представлены во взаимосвязи с математическими понятиями (например, из линейной алгебры и теории вероятностей). Таким образом, абстрактные математические понятия объясняются с двух различных точек зрения — финансов и программирования. Вдобавок такой подход позволяет получить новый полезный опыт, поскольку и математические, и финансовые понятия могут быть переведены непосредственно в исполняемый код и исследованы в интерактивном режиме.

Несколько человек, прочитавших одну из моих предыдущих книг, «Python для финансовых расчетов»¹, справедливо отметили, что она не подходит тем, кто только начинает знакомство с теорией финансов и программированием на Python. Действительно, предполагается, что читатель той книги имеет хотя бы небольшой опыт в данных сферах. Книга «Python для финансистов» восполняет этот пробел, поскольку фокусируется на основах и тем самым естественным образом подготавливает к прочтению «Python для финансовых расчетов», что в дальнейшем позволит развиваться и совершенствовать навыки работы с Python применительно к финансовым расчетам. Более подробно об этом рассказано в последней главе.

¹ Хилтиш И. Python для финансовых расчетов. — 2021.

Целевая аудитория

Об использовании Python в финансовой сфере я написал несколько книг, а моя компания, The Python Quants, предлагает соответствующее онлайн-обучение. И книги, и курсы предполагают, что читатель или слушатель уже обладает определенными знаниями в области финансов и программирования на Python или аналогичном ему языке.

Эта же книга знакомит читателя с данными темами с нуля, и ему нужны лишь базовые знания в области математики, в частности математического анализа, линейной алгебры и теории вероятностей. Материал содержит практически полную информацию обо всех описанных в нем математических понятиях. Тем не менее может пригодиться вводный учебник математики, например учебник Пембертона и Рау^{1,2}.

Книга предназначена для студентов, ученых и специалистов, которые хотят получить знания по теории финансов, финансовому моделированию и использованию Python в финансовой инженерии. Она также может послужить систематически выстроенной основой для создания более сложных книг и курсов по этим темам.

Даже если читатель не собирается переходить к более сложным темам финансовой инженерии, вычислительных финансов, алгоритмической торговли или управления активами, знания по Python и финансам, которые он почерпнет из этой книги, можно использовать при выполнении стандартных финансовых задач, например, при составлении инвестиционных портфелей в соответствии с современной портфельной теорией (modern portfolio theory, МРТ). Книга также рассказывает об оценке опционов и других деривативов с помощью стандартных методов, таких как портфельная репликация или риск-нейтральный подход к ценообразованию.

Эта книга подойдет руководителям, которые хотят узнать о применении Python в области финансов. В то же время она будет полезна тем, кто уже владеет Python или другим языком программирования и хочет узнать, как их можно использовать в данной сфере.

¹ *Pemberton M., Rau N. Mathematics for Economists: An Introductory Textbook. 4th ed. — Manchester University Press, 2016.*

² Русскоязычным читателям рекомендуем главную книгу по финансовой математике в России: *Ширяев А. Н. Основы стохастической финансовой математики. В двух томах. Том 1. Факты. Модели. Том 2. Теория. — М.: МЦНМО, 2016. — Примеч. науч. ред.*

Краткое описание книги

Книга состоит из следующих глав.

- **Глава 1. Финансы и Python.** Эта глава формирует основу всей книги. В ней кратко излагается история финансов, освещается подход к Python как к инструменту для финансовых расчетов, показывается, как работать с базовой инфраструктурой Python, на примере кода, представленного в интерактивных блокнотах Jupyter.
- **Глава 2. Экономика с двумя состояниями.** Здесь рассматривается наиболее простая модель экономики, в которой возможен финансовый анализ в условиях неопределенности, когда есть только две релевантные даты и два неопределенных состояния в будущем. Иногда ее называют статической экономикой с двумя состояниями. Несмотря на свою простоту, она позволяет представить такие базовые финансовые понятия, как чистая приведенная стоимость, ожидаемая доходность, волатильность, условные требования, репликация опционов, арбитражное ценообразование, мартингальная мера, полнота рынка, риск-нейтральный подход к ценообразованию и портфели Марковица.
- **Глава 3. Экономика с тремя состояниями.** В этой главе в модель добавляется третье неопределенное состояние в будущем и анализируется статическая экономика с тремя состояниями, что позволяет рассмотреть такие понятия, как неполнота рынка, неопределенность мартингальных мер, суперрепликация условных требований и аппроксимирующая репликация условных требований. Вдобавок здесь представлена модель ценообразования капитальных активов (Capital Asset Pricing Model, CAPM) в качестве подхода к равновесному ценообразованию финансовых активов.
- **Глава 4. Оптимальность и равновесие.** Основной темой здесь являются экономические агенты с их индивидуальными проблемами принятия решений. Анализ здесь опирается на доминирующую парадигму финансов для принятия решений в условиях неопределенности — *максимизацию ожидаемой полезности*. Через так называемого репрезентативного агента вводятся понятия равновесия, демонстрируется связь между оптимальностью и равновесием, с одной стороны, и мартингальными мерами и риск-нейтральным подходом к ценообразованию — с другой. Ко всему прочему, использование концепции репрезентативного агента представляет собой

один из способов преодоления трудностей, возникающих в экономиках с неполными рынками.

- **Глава 5. Статическая экономика.** Обобщает рассмотренные понятия и сводит полученные результаты к конечному (возможно, большому) числу неопределенных будущих состояний. Для анализа такой *обобщенной статической экономики* требуется немного больше математических формул.
- **Глава 6. Динамическая экономика.** Основываясь на анализе обобщенной статической экономики, здесь в финансовое моделирование добавляется динамика, что позволяет рассмотреть два частных случая динамической экономики в дискретном времени. Основная идея заключается в том, что неопределенность относительно будущих состояний экономики в целом разрешается постепенно с течением времени. Это можно смоделировать с помощью стохастических процессов, например биномиального процесса, который можно представить в виде биномиального дерева.
- **Глава 7. Что дальше?** В заключительной главе приводится множество дополнительных материалов для изучения из области математики, теории финансов и программирования на Python, также читателю даются рекомендации о дальнейших действиях.

Условные обозначения

В книге используются следующие условные обозначения.

Курсив

Служит для выделения новых понятий.

Интерфейс

Применяется для выделения URL-адресов, электронных адресов.

Моноширинный шрифт

Служит для выделения в тексте элементов программ, таких как переменные или названия функций, баз данных, типов данных, переменных окружения, операторов и ключевых слов, имен и расширений файлов. Кроме того, используется для оформления листингов.



Этот рисунок обозначает обычное примечание.



А этот — предупреждение или предостережение.



Этим рисунком обозначена важная информация.

Примеры кода, использованные в книге

Все вспомогательные материалы — примеры кода, упражнения и т. д. — можно скачать здесь: <https://finpy.pqp.io>.

По техническим вопросам или проблемам, связанным с примерами кода, можно обращаться по адресу bookquestions@oreilly.com.

В общем случае все примеры кода из книги вы можете использовать в своих программах и в документации. Вам не нужно обращаться в издательство за разрешением, если вы не собираетесь воспроизводить существенные части программного кода. Если вы разрабатываете программу и используете в ней несколько фрагментов кода из книги, вам не нужно обращаться за разрешением. Но для продажи или распространения примеров из книги вам потребуется разрешение от издательства O'Reilly. Вы можете отвечать на вопросы, цитируя данную книгу или примеры из нее, но для включения существенных объемов программного кода из книги в документацию вашего продукта потребуется разрешение.

Мы рекомендуем, но не требуем добавлять ссылку на первоисточник при цитировании. Под ссылкой на первоисточник мы подразумеваем указание авторов, издательства и ISBN.

За получением разрешения на использование значительных объемов программного кода из книги обращайтесь по адресу permissions@oreilly.com.

Благодарности

В книге приведены ценные отзывы участников наших сертификационных программ по Python для финансовых расчетов.

Благодарю научных редакторов за их полезные комментарии. Они помогли внести многочисленные улучшения.

Я также признателен всей команде O'Reilly за помощь и поддержку.

Эта книга посвящена моей жене Сандре. Ты — любовь всей моей жизни.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете подробную информацию о наших книгах.

ГЛАВА 1

Финансы и Python

История теории финансов — интересный пример взаимосвязи между абстрактным теоретизированием и практическим применением.

Франк Мильне (1995)

Благодаря современным технологиям в последние годы хедж-фонды поглотили десятки миллиардов долларов инвестиций. Эти же технологии приносят прибыль людям, принимающим финансовые решения в таких организациях.

Лоуренс Флетчер (2020)

Цель главы 1 состоит в том, чтобы заложить основы для последующих глав, поэтому в ней дается краткий обзор освещаемых тем. Она начинается с раздела «Краткая история финансов», в котором в общих чертах рассказывается об истории развития и современном состоянии финансов. В разделе «Главные тенденции в области финансов» рассматриваются основные тенденции, касающиеся математики, технологий, данных, искусственного интеллекта и определяющие развитие финансов в долгосрочной перспективе. В связи с этим раздел «Четырехязычная сфера» говорит о том, что финансы в наше время — это дисциплина, состоящая из четырех тесно связанных между собой типов языков: естественного (в большей мере английского), языка финансов, математического и языка программирования. Далее объясняется, как устроена книга (раздел «Структура книги»). А в последнем разделе, «Вводная информа-

ция о Python», рассказывается об установке окружения Python на компьютер и об альтернативном варианте работы с ним: весь код может использоваться и исполняться через обычный браузер на Quant Platform.

Краткая история финансов

Чтобы лучше понять текущее состояние финансов и финансовой отрасли, нужно взглянуть, как они развивались. В 2006 году Рубинштейн условно разделил историю финансов как науки на три периода.

- *Древний период (до 1950 года)*. Период, характеризующийся в основном неформальными рассуждениями, эмпирическими правилами и опытом субъектов финансового рынка.
- *Классический период (1950–1980 годы)*. В это время произошло внедрение в финансы более строгих обоснований и математики. Были разработаны финансовые модели, например модель ценообразования опционов Блэка — Шоулза (1973), а также заложены некоторые основы, например риск-нейтральный подход к ценообразованию Харрисона и Крепса (1979).
- *Современный период (1980–2000 годы)*. Это период прогресса в отдельных областях финансов, например в финансовой инженерии, и решения проблем, связанных с важными эмпирическими явлениями на финансовых рынках, такими как стохастическая модель процентных ставок (Кокс, Ингерсолл и Росс, 1985) или модель стохастической волатильности (Хестон, 1993).

Сейчас, спустя 15 лет после публикации книги Рубинштейна, можно выделить четвертый и пятый периоды, которые обеспечили появление и нынешнюю популярность Python в финансах.

- *Вычислительный период (2000–2020 годы)*. В это время произошел переход от теоретической направленности финансов к вычислительной, что обусловлено развитием программно-технических средств. Эту смену парадигмы хорошо иллюстрирует численный алгоритм для оценки американских опционов методом Монте-Карло, представленный в работе Лонгстаффа и Шварца в 2001 году. Алгоритм требует больших вычислительных ресурсов, поскольку для оценки стоимости всего одного опциона нужны сотни тысяч численных моделирований и анализ множества обычных регрессий методом наименьших квадратов (см.: Хилпиш, 2018).

- *Период использования искусственного интеллекта (после 2020 года).* Прогресс в области искусственного интеллекта (ИИ) и связанные с ним истории успеха подстегнули интерес к использованию возможностей ИИ в финансовой сфере. Несмотря на существование эффективных приложений в данной области (см.: Хилпиш, 2020), можно предположить, что с 2020 года парадигма планомерно смещается в сторону *финансовых систем, основанных на ИИ*. Таким образом, мы становимся свидетелями перехода от простых, в основном линейных, финансовых моделей к сложным моделям и использованию алгоритмов ИИ (глубоких нейронных сетей или обучения с подкреплением) для фиксации, описания и объяснения финансовых явлений.

Главные тенденции в области финансов

Как и многие другие предметы и отрасли, наука о финансах со временем стала формализованной научной дисциплиной. Этому прогрессу поспособствовали растущая популярность формальной математики, развитие технологий, доступность данных и усовершенствованные алгоритмы, например ИИ-алгоритмы. В целом в эволюции финансов можно выделить четыре основных направления.

- *Математика.* Начиная с 1950-х годов наука о финансах становится все более формализованной дисциплиной, в которой систематически используются знания из различных областей математики, например линейной алгебры и стохастического исчисления. Появление портфельной теории Марковица (1952) стало прорывом в количественном анализе и точкой перехода от древнего периода с его неформальными рассуждениями к классическому периоду.
- *Технологии.* Благодаря рабочим станциям, серверам и компьютерам, которые начали широко использоваться в конце 1980-х — начале 1990-х годов, технологии стали проникать и в финансовую отрасль. И если сначала техника имела довольно небольшую вычислительную мощность, то сейчас даже самые сложные финансовые задачи можно решить с помощью специальной программы, не прибегая к специальным моделям и методам, характерным для классического и современного периодов. Кредо стало таким: «Обновляйте оборудование и используйте современное программное обеспечение вместе с правильными вычислительными методами». При этом большинство современных компьютеров, предназначенных для

использования массовым потребителем, уже обладает мощностью, необходимой для высокопроизводительной работы (например, для параллельной обработки), что значительно упрощает изучение финансовой инженерии и взаимодействие с финансовыми системами, основанными на ИИ.

- *Данные.* Если в древний и классический периоды теории и практики черпали информацию, касающуюся финансов, в основном из специализированных печатных изданий (здесь можно вспомнить Wall Street Journal или Financial Times), то сегодня возросла доступность массива финансовых данных в электронном виде. Массивы высокочастотных внутридневных данных стали нормой и заменили цены на момент закрытия биржи в качестве основной базы для эмпирических исследований. Каждый торговый день одна акция может генерировать массивы внутридневных данных, содержащие более 100 000 значений, что приблизительно эквивалентно ценам на момент закрытия биржи для той же акции за 400 лет (250 торговых дней в году умножить на 400 лет). Вдобавок с недавнего времени наблюдается распространение открытых массивов данных, что также значительно снижает порог входа в финансовую инженерию, алгоритмическую торговлю или финансовую эконометрику.
- *Искусственный интеллект.* Увеличение объема финансовых данных (возникновение «больших финансовых данных») в наши дни делает применение алгоритмов искусственного интеллекта: алгоритмов машинного обучения, глубокого обучения или обучения с подкреплением (см.: Хилпиш, 2020) — не только возможным, но и во многих случаях необходимым подходом к работе с финансами. Традиционные статистические методы из финансовой эконометрики уже не подходят для решения большинства современных проблем, возникающих на финансовых рынках. В условиях нелинейной, многомерной, постоянно меняющейся финансовой среды алгоритмы на основе искусственного интеллекта нередко становятся единственным средством обнаружения релевантных взаимосвязей и закономерностей, получения важных данных и использования улучшенных возможностей прогнозирования.

В книге дается основная информация о финансовой математике и современных технологиях, используемых для реализации формальных финансовых моделей. С ее помощью можно приобрести навыки работы с массивами финансовых данных, наиболее часто встречающимися в финансовой сфере. Книга предназначена для того, чтобы подготовить читателя к более сложным темам финансовой инженерии и применения ИИ к финансовым расчетам.



Python и финансы

Все чаще основой финансовых расчетов становятся алгоритмы, требующие больших вычислительных ресурсов, постоянно растущая доступность данных и искусственный интеллект. Python оказался подходящим языком программирования и технологической платформой для удовлетворения потребностей и решения задач, обусловленных тенденциями развития финансовой сферы.

Четырехязычная сфера

На фоне перечисленных выше тенденций финансы стали сферой применения четырех языков.

- *Естественный язык.* Сегодня основным языком публикуемых финансовых исследований, книг, статей и новостей является английский.
- *Финансовый язык.* Как и в любой другой области, в финансах есть свои технические термины, понятия и выражения, описывающие определенные финансовые явления или понятия.
- *Математический язык.* Математика — самый удобный языковой инструмент для формализации финансовых понятий и концепций.
- *Язык программирования.* Как отмечается в эпиграфе к введению, Python — лучший из существующих языков программирования для работы в финансовой индустрии.

Таким образом, 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, Python. Здесь подразумевается, что при ограниченном количестве времени, выделенном на изучение финансов, наряду с финансовой математикой лучше сосредоточить свои силы на Python, а не на другом языке программирования.

Структура книги

Как в издании совмещается использование всех этих четырех языков? С естественным языком все понятно: можно либо читать книгу в оригинале, либо пользоваться переводом на родной язык. Осталось решить вопрос с остальными.

К примеру, книга не может подробно объяснить каждый отдельный раздел математики, необходимый в финансах, или представить все концепции программирования (Python), использующиеся в финансовой инженерии. Тем не менее в ней была сделана попытка связать понятия из финансов, математики и программирования между собой.

Начиная с главы 2, информация в книге будет подаваться через введение финансового понятия (или концепции) вместе с его математическим представлением и реализацией на Python. К примеру, в таблице из главы 3 перечислены финансовые понятия, о которых будет идти речь, соответствующие им основные математические элементы и основная структура данных Python, используемая для реализации финансовой математики.

Финансы	Математика	Python
Неопределенность	Вероятностное пространство	<code>ndarray</code>
Финансовые активы	Векторы, матрицы	<code>ndarray</code>
Достижимые условные требования	Линейная оболочка векторов, базис векторного пространства	<code>ndarray</code>

Чтобы проиллюстрировать общую структуру книги, далее приведу пример описания финансового понятия, который будет рассматриваться подробнее в следующих главах.

Возьмем из таблицы концепцию *неопределенности в финансах*. Под неопределенностью понимается невозможность заранее узнать состояние модельной экономики в будущем. Будущее состояние экономики важно, например, для определения выплаты по европейскому колл-опциону (то есть по опциону на покупку). В условиях дискретности мы имеем дело с конечным числом таких состояний, например двумя, тремя или более. В самом простом случае, когда рассматриваются только два будущих состояния, выплата по европейскому колл-опциону математически представляется как *случайная величина*, которая, в свою очередь, формально может быть представлена вектором $\mathbf{v}$, являющимся элементом *векторного пространства* $\mathbb{R}^2$. Векторное пространство — это набор объектов (векторов), для которых определены сложение и скалярное умножение. Формально вектор $\mathbf{v}$ можно представить следующим образом:

$$\mathbf{v} = \begin{pmatrix} v^u \\ v^d \end{pmatrix} \in \mathbb{R}_{\geq 0}^2$$

Здесь предполагается, что оба векторных элемента являются неотрицательными вещественными (действительными) числами $v^u, v^d \in \mathbb{R}_{\geq 0}$. Другими словами, если неопределенная, зависящая от состояния экономики цена акций, на которые выписан европейский колл-опцион, определяется как:

$$S = \begin{pmatrix} 20 \\ 5 \end{pmatrix} \in \mathbb{R}_{\geq 0}^2,$$

а цена исполнения опциона $K = 15$, то выплата C по европейскому колл-опциону будет равна:

$$C = \max(S - K, 0) = \begin{pmatrix} \max(20 - 15, 0) \\ \max(5 - 15, 0) \end{pmatrix} = \begin{pmatrix} 5 \\ 0 \end{pmatrix} \in \mathbb{R}_{\geq 0}^2.$$

Вот так понятия *неопределенной цены акции и зависящей от состояния экономики выплаты по европейскому опциону* могут быть математически смоделированы в виде вектора. Дисциплина, занимающаяся векторами и векторными пространствами в математике, называется *линейной алгеброй*.

Как же все это перевести на Python? Во-первых, вещественные числа в нем представлены как *числа с плавающей запятой* или объекты `float`:

```
In [1]: vu = 1.5 ❶
```

```
In [2]: vd = 3.75 ❷
```

```
In [3]: type(vu) ❸
```

```
Out[3]: float
```

```
In [4]: vu + vd ❹
```

```
Out[4]: 5.25
```

❶ Определение переменной `vu` и ее значения `1.5`.

❷ Определение переменной `vd` и ее значения `3.75`.

❸ Поиск и вывод типа объекта `vu` (`float`).

❹ Сложение значений `vu` и `vd`.

Во-вторых, наборы объектов одного типа в программировании обычно называются *массивами*. Поддержку таких структур данных в Python обеспечивает библиотека NumPy (<http://numpy.org/>). Основной структурой данных в ней