

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  
**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**  
**МЕХАНИКИ И ОПТИКИ»**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**ДЕЦЕНТРАЛИЗОВАННЫЙ АЛГОРИТМ УПРАВЛЕНИЯ КОНВЕЙЕРНОЙ**  
**СИСТЕМОЙ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ МУЛЬТИАГЕНТНОГО**  
**ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ**

Автор: Мухутдинов Дмитрий Вадимович \_\_\_\_\_

Направление подготовки: 01.04.02 Прикладная  
математика и информатика

Квалификация: Магистр

Руководитель: Фильченков А.А., к.ф-м.н. \_\_\_\_\_

**К защите допустить**

Руководитель ОП Парфенов В.Г., проф., д.т.н. \_\_\_\_\_

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Санкт-Петербург, 2019 г.

Студент Мухутдинов Д.В.

Группа М4239 Факультет ИТиП

Направленность (профиль), специализация

Технологии разработки программного обеспечения

Консультанты:

а) Вяткин В.В., проф., д.т.н.

\_\_\_\_\_

ВКР принята «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г.

Оригинальность ВКР \_\_\_\_\_ %

ВКР выполнена с оценкой \_\_\_\_\_

Дата защиты «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г.

Секретарь ГЭК Павлова О.Н.

\_\_\_\_\_

Листов хранения \_\_\_\_\_

Демонстрационных материалов/Чертежей хранения \_\_\_\_\_

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  
**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**  
**МЕХАНИКИ И ОПТИКИ»**

**УТВЕРЖДАЮ**

Руководитель ОП

проф., д.т.н. Парфенов В.Г. \_\_\_\_\_

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

**ЗАДАНИЕ**  
**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

**Студент** Мухутдинов Д.В.

**Группа** М4239 **Факультет** ИТиП

**Руководитель** Фильченков А.А., к.ф.-м.н., кафедра КТ

**1 Наименование темы:** Децентрализованный алгоритм управления конвейерной системой с использованием методов мультиагентного обучения с подкреплением

**Направление подготовки (специальность):** 01.04.02 Прикладная математика и информатика

**Направленность (профиль):** Технологии разработки программного обеспечения

**Квалификация:** Магистр

**2 Срок сдачи студентом законченной работы:** «31» мая 2019 г.

**3 Техническое задание и исходные данные к работе**

Требуется разработать децентрализованный алгоритм управления конвейерной системы для транспортировки багажа. Алгоритм должен позволять контроллерам конвейерной сети динамически изменять свое поведение в целях адаптации под изменившиеся условия работы, такие как поломка одного из конвейеров или изменение интенсивности потока багажа. Алгоритм должен обеспечивать своевременную доставку багажных единиц до пунктов назначения, в то же время минимизируя энергопотребление всей системы в целом.

**4 Содержание выпускной работы (перечень подлежащих разработке вопросов)**

Пояснительная записка должна содержать обзор существующих результатов в сфере управления конвейерными системами, а также в сферах, имеющих непосредственное отношение к предложенному алгоритму (таких как сетевая маршрутизация и обучение с подкреплением). Также записка должна содержать подробное изложение предложенного алгоритма и данные экспериментального сравнения его производительности с производительностью существующих методов управления конвейерной системой, проведенного с помощью виртуальной имитационной модели конвейерной сети.

**5 Перечень графического материала (с указанием обязательного материала)**

Графические материалы и чертежи работой не предусмотрены

## **6 Исходные материалы и пособия**

- а) Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction. The MIT Press, 2012
- б) Mnih et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015.

## **7 Дата выдачи задания «01» сентября 2017 г.**

Руководитель ВКР \_\_\_\_\_

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

«01» сентября 2017 г.

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  
**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**  
**МЕХАНИКИ И ОПТИКИ»**

**АННОТАЦИЯ**  
**ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

**Студент:** Мухутдинов Дмитрий Вадимович

**Наименование темы ВКР:** Децентрализованный алгоритм управления конвейерной системой с использованием методов мультиагентного обучения с подкреплением

**Наименование организации, где выполнена ВКР:** Университет ИТМО

**ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

1 Цель исследования: Разработка децентрализованного алгоритма управления конвейерной системой, способного адаптироваться к гетерогенным изменениям в условиях работы.

2 Задачи, решаемые в ВКР:

- а) разработка виртуальной имитационной модели конвейерной сети, позволяющей проводить сравнительный анализ алгоритмов управления;
- б) реализация ряда известных алгоритмов управления конвейерной сетью (а также маршрутизации) в рамках разработанной имитационной модели;
- в) разработка алгоритма, соответствующего поставленным требованиям;
- г) экспериментальное сравнение разработанного алгоритма с существующими по качеству работы.

3 Число источников, использованных при составлении обзора: 3

4 Полное число источников, использованных в работе: 18

5 В том числе источников по годам:

Отечественных			Иностранных		
Последние 5 лет	От 5 до 10 лет	Более 10 лет	Последние 5 лет	От 5 до 10 лет	Более 10 лет
0	0	0	3	2	13

6 Использование информационных ресурсов Internet: нет

7 Использование современных пакетов компьютерных программ и технологий:

Пакеты компьютерных программ и технологий	Раздел работы
Язык программирования Python 3.6	
Библиотека для дискретно-событийного моделирования SimPy	
Библиотека для операций с графами NetworkX	
Математические пакеты NumPy и SciPy	
Библиотеки для машинного обучения scikit-learn и PyTorch	
Среда интерактивной разработки Jupyter Lab	
Библиотеки Matplotlib, Seaborn и Pandas для обработки и визуализации данных	

#### 8 Краткая характеристика полученных результатов

Разработан алгоритм управления конвейерной системой на основе глубокого мультиагентного обучения с подкреплением. В ходе экспериментального исследования было установлено, что разработанный алгоритм превосходит существующие по качеству работы и способен адаптироваться к изменениям во внешней среде.

#### 9 Гранты, полученные при выполнении работы

Государственное задание № 2.8866.2017/БЧ «Технология разработки программного обеспечения систем управления ответственными объектами на основе глубокого обучения и конечных автоматов»

#### 10 Наличие публикаций и выступлений на конференциях по теме работы

- 1 Multi-agent deep learning for simultaneous optimization for time and energy in distributed routing system / D. Mukhutdinov [et al.] // Future Generation Computer Systems. — 2019. — Vol. 94. — P. 587–600.

Студент            Мухутдинов Д.В. \_\_\_\_\_

Руководитель    Фильченков А.А. \_\_\_\_\_

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г.

ВВЕДЕНИЕ .....	5
1. Первая глава .....	9
1.1. Таблицы .....	9
1.2. Рисунки.....	9
1.3. Листинги .....	10
2. Проверка сквозной нумерации.....	11
Выводы по главе 2 .....	11
ЗАКЛЮЧЕНИЕ.....	12
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	13
ПРИЛОЖЕНИЕ А. Пример приложения.....	15
ПРИЛОЖЕНИЕ Б. Еще один пример приложения с невероятно длинным названием для тестирования переносов .	17
ПРИЛОЖЕНИЕ В. Пример огромного листинга.....	18

## ВВЕДЕНИЕ

Конвейерные системы широко применяются для автоматизированной транспортировки объектов и материалов. Они являются неотъемлемой частью комплексов промышленного оборудования, пунктов сортировки грузов, багажных систем в аэропортах и т. д.. Оптимизация работы таких системы имеет высокую практическую и экономическую значимость, что мотивирует поиск эффективных методов управления конвейерными системами.

Для управления большинством современных конвейерных систем применяются централизованные статические стратегии управления, специально разрабатываемые под конкретную систему одновременно с проектированием ее физической топологии в целях решения задач конкретного предприятия [5]. Преимуществами такого подхода являются высокая производительность работы и предсказуемость поведения системы. Недостатками такого подхода являются высокая стоимость и долгие сроки разработки кастомизированной стратегии управления, неспособность системы адаптироваться к изменениям во внешние условия, не учтенных во время проектирования (таким как неожиданный отказ отдельных элементов системы), а также наличие централизованного контроллера как критической точки отказа.

В связи с обозначенными недостатками использования кастомизированных стратегий управления существует интерес к разработке обобщенных систематических подходов к управлению конвейерными системами. Наиболее популярным подходом из используемых является управление с прогнозируемыми моделями (*англ.* model predictive control, MPC) [13]. Управление с прогнозируемыми моделями изначально разрабатывалось для задач химической промышленности и нефтепереработки [6], и на данный момент широко используется в этих сферах. В последние несколько лет повысился интерес к применению данного подхода к управлению другими типами киберфизических систем, в том числе промышленными конвейерными линиями [3, 9] и системами распределения багажа на основе рельсовых тележек (*англ.* destination coded vehicles, DCVs) [15, 18]. Существующие алгоритмы на основе управления с прогнозируемыми моделями обобщаются на различные конкретные конфигурации физических систем, но все еще предусматривают наличие централизованного контроллера. Кроме того, модификация существующего алгоритма управления (например, добавления учета энергопотребления системы)



во многих случаях является нетривиальной задачей, так как оптимизируемая функция во фреймворке MPC должна выражаться как задача линейного (*англ.* linear programming, LP) или квадратичного программирования (*англ.* quadratic programming, QP). В связи с этим существует интерес к разработке иных, децентрализованных подходов к управлению конвейерными сетями, требующих менее строгих ограничений к характеру решаемой задачи.

При рассмотрении штучных конвейеров, перемещающих отдельные объекты, как в случае систем для перемещения багажа в аэропортах, задача управления конвейерной системой по большей части сводится к задаче пакетной маршрутизации (*англ.* packet routing). Задача пакетной маршрутизации — это задача поиска пути наименьшей стоимости в графе из текущей вершины  $n$  в вершину назначения  $d$ . В контексте конвейерных систем топология конвейерной сети может быть представлена в виде ориентированного графа, а перемещаемые по конвейерам объекты могут быть абстрагированы как «пакеты».

Задача пакетной маршрутизации впервые обрела актуальность с появлением компьютерных сетей. Первые алгоритмы сетевой маршрутизации появились в процессе разработки сети ARPANet. Именно тогда были изобретены такие подходы к пакетной маршрутизации, как дистанционно-векторный (*англ.* distance-vector) [11] и состояния каналов связи (*англ.* link-state) [10], которые и по сей день лежат в основе таких стандартных и широко применяемых алгоритмов сетевой маршрутизации, как Routing Information Protocol (RIP) [7] и Open Shortest Path First (OSPF) [12]. Преимуществами алгоритмов сетевой маршрутизации являются обусловленные характером задачи децентрализованность, низкая требовательность к вычислительным ресурсам и устойчивость к отказам маршрутизаторов и обрывам соединений. В работе [17] было продемонстрировано, что простой дистанционно-векторный алгоритм маршрутизации может быть напрямую применен к задаче управления конвейерной системой для транспортировки багажа, что позволяет достигнуть устойчивости системы к отказам отдельных конвейеров. Однако, в силу своей простоты, дистанционно-векторный алгоритм решает исключительно задачу направления перемещаемых объектов вдоль кратчайших путей в конвейерной сети, что ограничивает его применимость в тех случаях, когда необходимо учитывать дополнительные критерии оптимизации (такие как энергопотребление системы).

Существуют и другие подходы к решению задачи пакетной маршрутизации. Одним из таких подходов является подход на основе идеи обучения с подкреплением (*англ.* reinforcement learning, RL). Первым таким алгоритмом стал алгоритм Q-routing [2], основанный на методе Q-learning [16]. Этот алгоритм, как и его модификации [4, 8], благодаря обучению с подкреплением оказался способен лучше адаптироваться к изменениям в интенсивности сетевого трафика, чем алгоритмы, основанные на вычислении кратчайшего пути. Но из-за использования большого количества служебных сообщений, использующих те же каналы передачи данных, что и целевые пакеты, применение таких алгоритмов в реальных компьютерных сетях ограничено.

В контексте конвейерных сетей, однако, целевые «пакеты» являются физическими объектами (например, чемоданами), перемещаемыми по конвейерной ленте, в то время как служебные сообщения передаются по проводным соединениям между контроллерами. Таким образом, время распространения служебных сообщений по системе пренебрежимо мало по сравнению с временем перемещения целевых объектов, что нивелирует озвученный недостаток алгоритмов маршрутизации, основанных на обучении с подкреплением.

В данной работе будет предложено несколько модификаций децентрализованного алгоритма управления конвейерной системой, основанного на методе Q-routing, но использующего нейронную сеть в качестве обучающегося агента. Для демонстрации способности работы алгоритма в различных постановках задачи маршрутизации он будет исследован как в имитационной модели конвейерной системы, так и в имитационной модели абстрактной компьютерной сети.

В главе 1 будет сформулирована обобщенная постановка задачи маршрутизации в терминах мультиагентного обучения с подкреплением, к которой будет сведена задача управления конвейерной системой. Будут рассмотрены существующие алгоритмы маршрутизации, их сильные и слабые стороны. Также будут рассмотрены существующие методы обучения нейронных сетей с подкреплением, в том числе в мультиагентном случае.

В главе 2 будет рассмотрен предложенный алгоритм и его модификации и обоснованы решения, принятые в ходе его разработки.

В главе 3 будут приведены экспериментальные результаты работы алгоритма для задач пакетной маршрутизации в моделях компьютерной сети и

управления системой багажных конвейеров. Также будет проведено экспериментальное сравнение алгоритма с существующими алгоритмами маршрутизации и управления конвейерной системой.

## ГЛАВА 1. ПЕРВАЯ ГЛАВА

Пример ссылок в рамках обзора: [2, 14, 17]. Вне обзора: [1].

### 1.1. Таблицы

В качестве примера таблицы приведена таблица 1.

Таблица 1 – Таблица умножения (фрагмент)

–	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68

Есть еще такое окружение `tabularx`, его можно аккуратно растянуть на всю страницу. Приведем пример (таблица 2).

Таблица 2 – Таблица умножения с помощью `tabularx` (фрагмент)

–	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68

### 1.2. Рисунки

Пример рисунка (с помощью `TikZ`) приведен на рисунке 1. Под `pdflatex` можно также использовать `*.jpg`, `*.png` и даже `*.pdf`, под `latex` можно использовать `Metapost`. Последний можно использовать и под `pdflatex`, для чего в стилевике продекларированы номера картинок от 1 до 20.

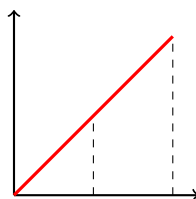


Рисунок 1 – Пример рисунка

### 1.3. Листинги

В работах студентов кафедры «Компьютерные технологии» часто встречаются листинги. Листинги бывают двух основных видов — исходный код и псевдокод. Первый оформляется с помощью окружения `lstlisting` из пакета `listings`, который уже включается в стилевике и немного настроен. Пример Hello World на Java приведен на листинге 1. Пример большого листинга — в приложении (листинг В.1).

Листинг 1 – Пример исходного кода на Java

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

Псевдокод можно оформлять с помощью разных пакетов. В данном стилевике включается пакет `algorithmicx`. Сам по себе он не генерирует флагов, поэтому для них используется пакет `algorithm`. Пример их совместного использования приведен на листинге 2.

Листинг 2 – Пример псевдокода

```
function IsPrime( $N$ )
  for  $t \leftarrow [2; \lfloor \sqrt{N} \rfloor]$  do
    if  $N \bmod t = 0$  then
      return false
    end if
  end for
  return true
end function
```

Наконец, листинги из `listings` тоже можно подвешивать с помощью `algorithm`, пример на листинге 3.

Листинг 3 – Исходный код и флаги `algorithm`

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

## ГЛАВА 2. ПРОВЕРКА СКВОЗНОЙ НУМЕРАЦИИ

Листинг 4 должен иметь номер 4.

Листинг 4 – Исходный код и флюид `algorithm`

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

Рисунок 2 должен иметь номер 2.

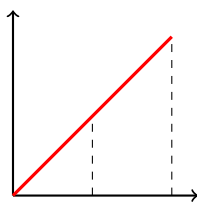


Рисунок 2 – Пример рисунка

Таблица 3 должна иметь номер 3.

Таблица 3 – Таблица умножения с помощью `tabularx` (фрагмент)

–	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68

### Выводы по главе 2

В конце каждой главы желательно делать выводы. Вывод по данной главе — нумерация работает корректно, ура!

## **ЗАКЛЮЧЕНИЕ**

В данном разделе размещается заключение.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 *Bellman R.* On a routing problem // Quarterly of Applied Mathematics. — 1958. — No. 16. — P. 87–90.
- 2 *Boyan J. A., Littman M. L.* Packet routing in dynamically changing networks: a reinforcement learning approach // Advances in Neural Information Processing Systems. — 1994. — No. 6. — P. 671–678.
- 3 *Cataldo A., Scattolini R.* Dynamic pallet routing in a manufacturing transport line with model predictive control // IEEE Transactions on control systems technology. — 2016. — Vol. 24, no. 5. — P. 1812–1819.
- 4 *Choi S. P. . M., Yeung D.-Y.* Predictive Q-Routing: A Memory-based Reinforcement Learning Approach to Adaptive Traffic Control // Advances in Neural Information Processing Systems. — 1996. — No. 8. — P. 945–951.
- 5 *De Neufville R.* The baggage system at Denver: prospects and lessons // Journal of Air Transport Management. — 1994. — Vol. 1, no. 4. — P. 229–236.
- 6 *Eaton J. W., Rawlings J. B.* Model-predictive control of chemical processes // Chemical Engineering Science. — 1992. — Vol. 47, no. 4. — P. 705–720.
- 7 *Hendrick C.* Routing Information Protocol : RFC. — 06/1988. — No. 1058.
- 8 *Kumar S., Miikkulainen R.* Dual reinforcement Q-routing: An on-line adaptive routing algorithm // Artificial Neural Networks in Engineering. — 1997. — No. 7. — P. 231–238.
- 9 *Luo J., Huang W., Zhang S.* Energy cost optimal operation of belt conveyors using model predictive control methodology // Journal of Cleaner Production. — 2015. — Vol. 105. — P. 196–205.
- 10 *McQuillan J. M., Richer I., Rosen E. C.* The New Routing Algorithm for the ARPANet // IEEE Trans. on Comm. — 1980. — Vol. 28, no. 5. — P. 711–719.
- 11 *McQuillan J. M., Walden D. C.* The ARPA network design decisions // Computer Networks. — 1977. — Vol. 1, no. 5. — P. 243–289.
- 12 *Moy J.* OSPF Version 2 : RFC. — 04/1998. — No. 1058.
- 13 *Qin S. J., Badgwell T. A.* A survey of industrial model predictive control technology // Control engineering practice. — 2003. — Vol. 11, no. 7. — P. 733–764.



- 14 *Tan M.* Multi-agent reinforcement learning: Independent vs. cooperative agents // Proceedings of the tenth international conference on machine learning. — 1993. — P. 330–337.
- 15 *Tarau A. N., De Schutter B., Hellendoorn H.* Model-based control for route choice in automated baggage handling systems // IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). — 2010. — Vol. 40, no. 3. — P. 341–351.
- 16 *Watking C.* Learning from Delayed Rewards : PhD thesis / Watking C. — Cambridge : King's College, 1989.
- 17 *Yan J., Vyatkin V.* Distributed Software Architecture Enabling Peer to Peer Communicating Controllers // IEEE Transactions on Industrial Informatics. — 2013. — Vol. 9, no. 4. — P. 2200–2209.
- 18 *Zeinaly Y., De Schutter B., Hellendoorn H.* An integrated model predictive scheme for baggage-handling systems: Routing, line balancing, and empty-cart management // IEEE Transactions on control Systems technology. — 2015. — Vol. 23, no. 4. — P. 1536–1545.

## ПРИЛОЖЕНИЕ А. ПРИМЕР ПРИЛОЖЕНИЯ

В приложениях рисунки, таблицы и другие подобные элементы нумеруются по приложениям с соответствующим префиксом. Проверим это.

Листинг А.1 должен иметь номер А.1.

Листинг А.1 – Исходный код и флоат `algorithm`

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

Рисунок А.1 должен иметь номер А.1.

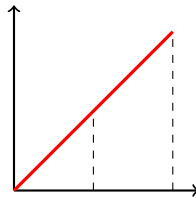


Рисунок А.1 – Пример рисунка

Таблица А.1 должна иметь номер А.1.

Таблица А.1 – Таблица умножения с помощью `tabularx` (фрагмент)

–	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68

Заодно проверим нумерованные и ненумерованные перечисления. Ненумерованные:

- пункт А;
- пункт Б;
- пункт В.

Нумерованные списки нескольких уровней:

- а) первый элемент;
- б) второй элемент с подэлементами:
  - 1) первый подэлемент;

2) второй подэлемент;

3) третий подэлемент.

в) третий элемент;

г) четвертый элемент;

д) пятый элемент;

е) шестой элемент;

ж) седьмой элемент;

и) восьмой элемент;

к) девятый элемент;

л) десятый элемент.

**ПРИЛОЖЕНИЕ Б. ЕЩЕ ОДИН ПРИМЕР ПРИЛОЖЕНИЯ С  
НЕИМОВЕРНО ДЛИННЮЩИМ НАЗВАНИЕМ ДЛЯ  
ТЕСТИРОВАНИЯ ПЕРЕНОСОВ**

Проверим на примере таблиц, что нумерация в приложениях — по приложениям. Таблица Б.1 должна иметь номер Б.1.

Таблица Б.1 – Таблица умножения с помощью `tabularx` (фрагмент)

–	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68

## ПРИЛОЖЕНИЕ В. ПРИМЕР ОГРОМНОГО ЛИСТИНГА

Листинг В.1 – Пример большого листинга

```
import java.util.*;

public class Example {
    static int[] restoreOutgoing(int[] g, int[] outgoing,
                                int vertex, int mask) {
        int[] rv = new int[1 + Integer.bitCount(mask)];
        int n = g.length;
        int current = rv.length - 1;
        while (true) {
            rv[current] = vertex;
            if (current == 0) {
                if (vertex != 0) {
                    throw new AssertionError();
                }
                return rv;
            }
            mask ^= 1 << (vertex - 1);
            int prevMask = outgoing[mask] & g[vertex];
            if (prevMask == 0) {
                throw new AssertionError();
            }
            vertex = Integer.numberOfTrailingZeros(prevMask);
            --current;
        }
    }

    static int[] restoreIncoming(int[] g, int[] incoming,
                                int vertex, int mask) {
        int[] rv = new int[1 + Integer.bitCount(mask)];
        int n = g.length;
        int current = 0;
        while (true) {
            rv[current] = vertex;
            if (current == rv.length - 1) {
                if (vertex != 0) {
                    throw new AssertionError();
                }
                return rv;
            }
        }
    }
}
```

```

mask ^= 1 << (vertex - 1);
int nextMask = incoming[mask] & g[vertex];
if (nextMask == 0) {
    throw new AssertionError();
}
vertex = Integer.numberOfTrailingZeros(nextMask);
++current;
}
}
}

```