



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**"МИРЭА - Российский технологический университет"**

**РТУ МИРЭА**

---

**Институт Информационных Технологий**

**Кафедра Вычислительной Техники**

**Практические работы**

**по дисциплине**

**«Многоагентное моделирование»**

Студент группы: ИКБО-04-22

Арефьев А. М.  
(Ф.И.О. студента)

Преподаватель

Гололобов А.А.  
(Ф.И.О. преподавателя)

Москва 2024

## **СОДЕРЖАНИЕ**

ВВЕДЕНИЕ.....	3
ПОСТАНОВКА ЗАДАЧИ.....	4
РЕШЕНИЕ.....	5
ТЕСТИРОВАНИЕ.....	6
ЗАКЛЮЧЕНИЕ.....	10
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	11
ПРИЛОЖЕНИЯ.....	12

# **ВВЕДЕНИЕ**

Современные технологии и методы моделирования играют ключевую роль в изучении и понимании сложных систем и процессов. Много-агентное моделирование, в частности, становится всё более популярным инструментом для анализа и оптимизации систем, где множество взаимодействующих агентов влияет на общий результат.

В данной работе рассмотрено создание многоагентной системы «Переправа» (Практическая работа №9) на языке Python.

## ПОСТАНОВКА ЗАДАЧИ

Имеются объекты плот, водное пространство, правая и левая пристань, и люди, переправляющиеся на плоту с правой пристани на левую. В нашем примере реализации объект плот обозначен соответствующей надписью и произвольной заливкой, водное пространство обозначено синей заливкой, пристани черной, а люди обозначены символом смайлика ☺.

Если плот стоит у правой пристани, выполняется событие ЛюдиНаПристань, при этом случайное число прибывших людей на пристань не должно превысить 4 человек, вместе с людьми находящимися на пристани.

Если плот стоит у правой пристани, выполняется событие ЛюдиНаПлот, при этом случайное число людей перешедших на плот не должно превысить 4 человек, вместе с людьми находящимися на плоту [8].

Если есть люди на плоту, выполняется событие Влево, при этом плот передвигается влево пока он не достигнет левой пристани.

Если плот пристал к левой пристани, выполняется событие ЛюдиСПлота, при этом люди сначала сходя с плота, а потом уходят с левой пристани.

Если нет людей на плоту, выполняется событие Вправо, при этом плот передвигается вправо пока он не достигнет правой пристани. Далее события повторяются, пока не исчерпано заданное количество итераций.

## РЕШЕНИЕ

Для решения задачи было принято решение написать класс `DockRaftSimulation`, представляющий собой симуляцию. Класс имеет 3 основных метода:

1. Конструктор, в котором можно настроить максимальное количество людей на плоту или на пристани;

2. `display_state`, отображает текущее состояние и принимает текущий номер итерации.

3. `crossing_algorithm`, основной алгоритм, в каждой итерации, если на правом берегу людей меньше максимального количества, случайным образом добавляется от одного до нескольких человек.

- Затем, люди с правого берега по одному переходят на плот до заполнения плота.
- Плот переправляется на левый берег поэтапно, на каждом этапе обновляется визуализация.
- После переправы, люди с плота по одному переходят на левый берег.
- Левый берег очищается для следующей итерации.
- Если плот находится не на правом берегу, он возвращается на правый берег поэтапно.
- В конце всех итераций график отображается на экране.

Код представлен в приложении 1.

## **ТЕСТИРОВАНИЕ**

Программа выводит график для каждого действия, графики в совокупности образуют анимацию.

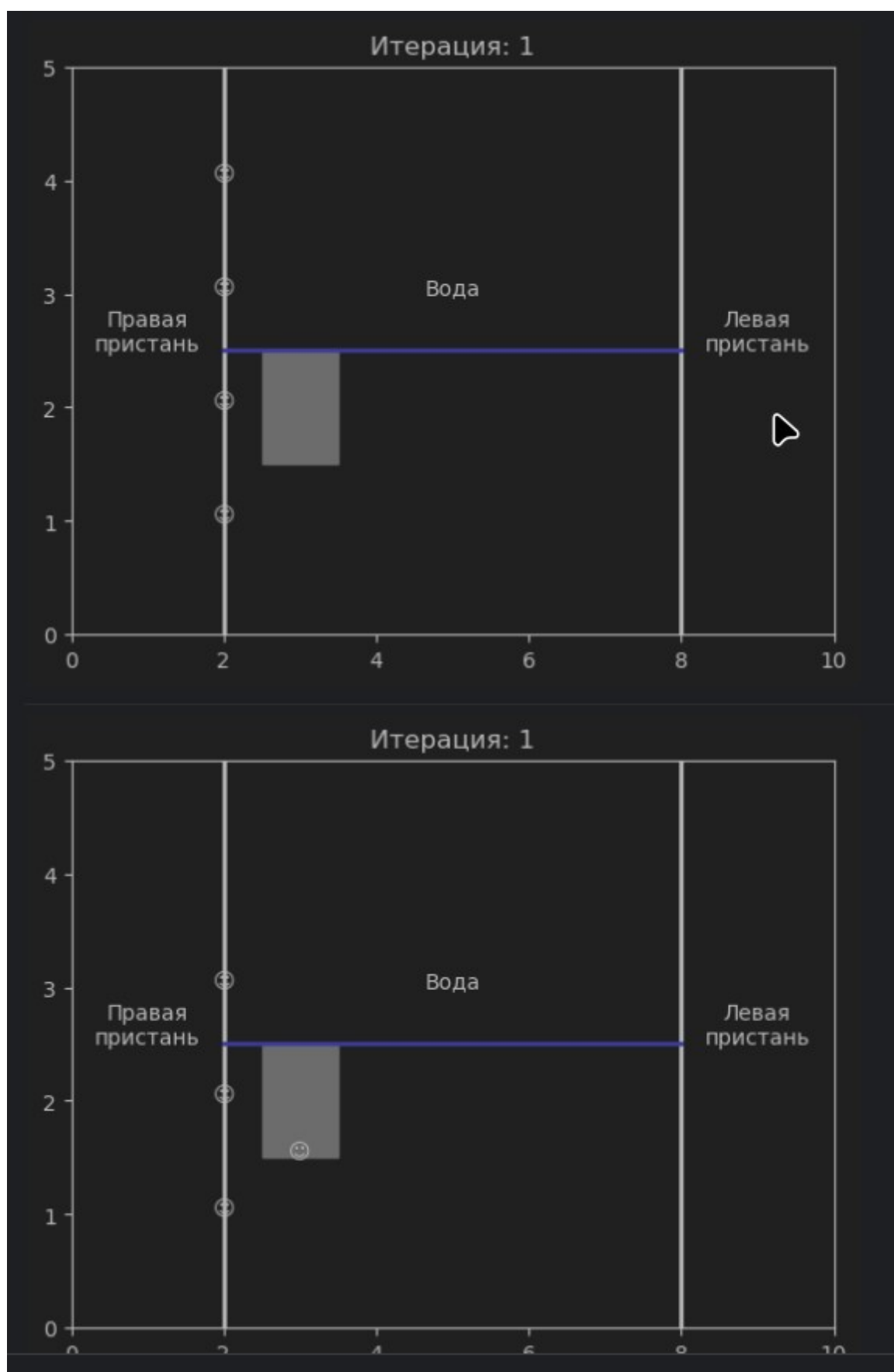


Рисунок 1 — Первые два графика анимации

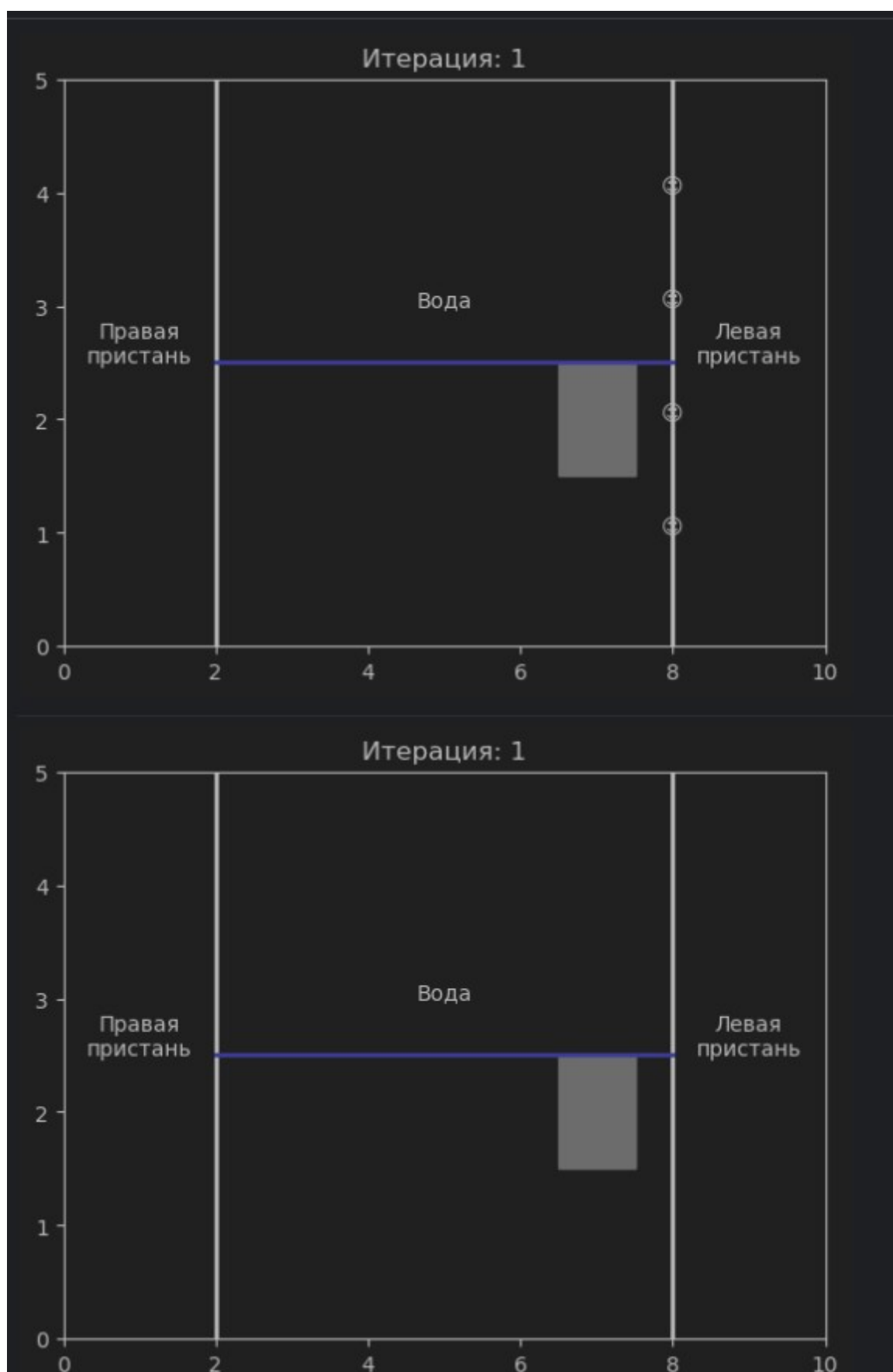


Рисунок 2 — Высадка людей на левую пристань и их уход с пристани



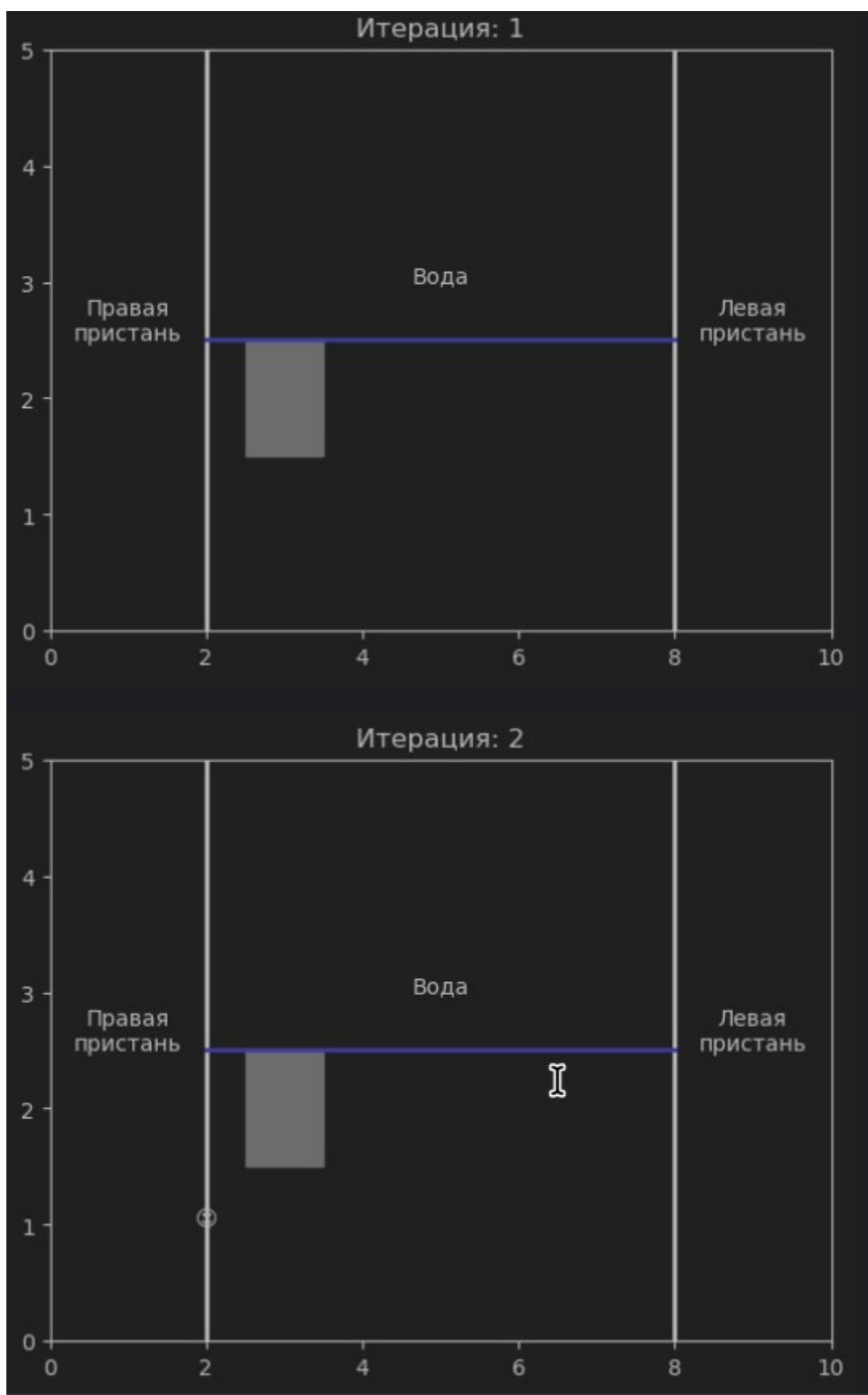


Рисунок 3 — Возврат плота на правую пристань и начало новой итерации.

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения практической работы я познакомился с созданием многоагентной модели на языке Python.

Изучил основы симуляции и концепции многоагентного моделирования.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Официальная сайт AnyLogic [Электронный ресурс] / Создание многоагентных моделей. – Режим доступа: <https://www.anylogic.ru/>
2. Практические работы РТУ МИРЭА по дисциплине Многоагентное моделирование [Электронный ресурс] / Создание многоагентных моделей. – Режим доступа: <https://online-edu.mirea.ru/course/view.php?id=6558>

## **ПРИЛОЖЕНИЯ**

Приложение А – Листинг 1

## Приложение А

*Листинг 1 — Код программы*

```
import matplotlib.pyplot as plt
import random
import time
from typing import List

class DockRaftSimulation:
    """
    Симуляция процесса переправы людей с одного берега на другой с
    использованием плота.

    Attributes:
        right_dock (List[str]): Список людей на правом берегу.
        left_dock (List[str]): Список людей на левом берегу.
        raft (List[str]): Список людей на плоту.
        max_people_on_dock (int): Максимальное количество людей на
        пристани.
        max_people_on_raft (int): Максимальное количество людей на
        плоту.
        raft_position (int): Текущая позиция плота.
    """

    def __init__(self, max_people_on_dock: int = 4,
max_people_on_raft: int = 4):
        """
        Args:
            max_people_on_dock (int): Максимальное количество людей
            на пристани.
            max_people_on_raft (int): Максимальное количество людей
```

на плоту.

```
    """

    self.right_dock: List[str] = []
    self.left_dock: List[str] = []
    self.raft: List[str] = []
    self.max_people_on_dock = max_people_on_dock
    self.max_people_on_raft = max_people_on_raft
    self.raft_position = 3

def display_state(self, iteration: int) -> None:
    """

    Отображает текущее состояние симуляции.

    Args:
        iteration (int): Текущий номер итерации.
    """

    plt.clf()
    plt.title(f"Итерация: {iteration}")
    plt.xlim(0, 10)
    plt.ylim(0, 5)
    plt.plot([2, 2], [0, 5], 'k-', lw=2)
    plt.plot([8, 8], [0, 5], 'k-', lw=2)
    plt.plot([2, 8], [2.5, 2.5], 'b-', lw=2)
    plt.text(1, 2.5, "Правая\nпристань", ha='center')
    plt.text(9, 2.5, "Левая\nпристань", ha='center')
    plt.text(5, 3, "Вода", ha='center')
```

```

        for i, person in enumerate(self.right_dock):
            plt.text(2, i + 1, person, ha='center')

        for i, person in enumerate(self.left_dock):
            plt.text(8, i + 1, person, ha='center')

        plt.gca().add_patch(plt.Rectangle((self.raft_position -
0.5, 1.5), 1, 1, color='gray'))

        for i, person in enumerate(self.raft):
            plt.text(self.raft_position, i + 1.5, person,
ha='center')

        plt.pause(0.5)

def crossing_algorithm(self, iterations: int) -> None:
    """
    Основной алгоритм симуляции процесса переправы.

    Args:
        iterations (int): Количество итераций симуляции.
    """
    for iteration in range(1, iterations + 1):
        if len(self.right_dock) < self.max_people_on_dock:
            num_people = random.randint(1,
self.max_people_on_dock - len(self.right_dock))
            self.right_dock.extend(['@'] * num_people)
            self.display_state(iteration)

        while self.right_dock and len(self.raft) <

```

```

self.max_people_on_raft:

    person = self.right_dock.pop(0)
    self.raft.append(person)
    self.display_state(iteration)

if self.raft:
    for _ in range(4):
        self.raft_position += 1
        self.display_state(iteration)
        time.sleep(0.5)

while self.raft:
    person = self.raft.pop(0)
    self.left_dock.append(person)
    self.display_state(iteration)

self.left_dock.clear()
self.display_state(iteration)

if self.raft_position > 2:
    for _ in range(4):
        self.raft_position -= 1
        self.display_state(iteration)
        time.sleep(0.5)

plt.show()

simulation = DockRaftSimulation()

```



```
simulation.crossing_algorithm(10)
```