|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |
| Институт кибербезопасности и цифровых технологий |
| Кафедра КБ-2 «Прикладные информационные технологии» |

**ПРАКТИЧЕСКАЯ РАБОТА №3 по дисциплине**

**«Технологии и методы программирования»**

***(наименование дисциплины)***

**Тема практической работы   
«Шаблонный метод. Стратегия»**

Студент группы: БИСО-03-20 Шаршов И.А.

*(шифр учебной группы) (Фамилия И.О)*

Преподаватель: Лесько С.А.

*(Фамилия И.О)*

Москва, 2023

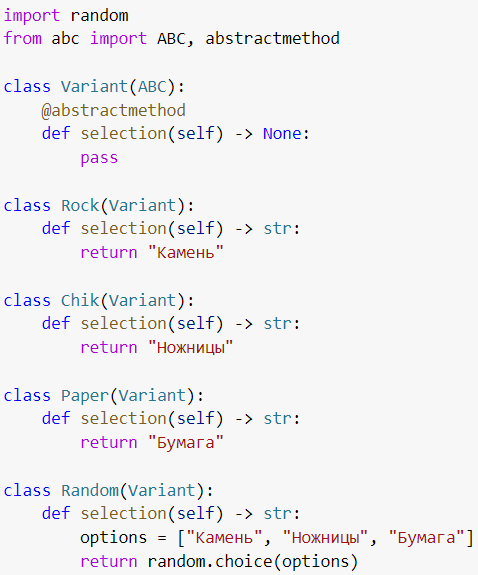
Практическая работа №3

[**Стратегия** 3](#_Toc134720515)

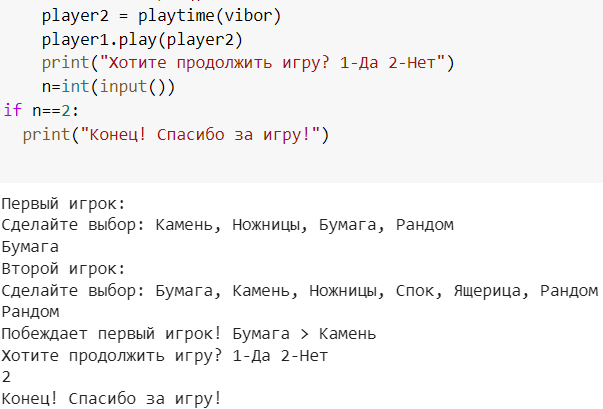
[**Шаблонный метод** 6](#_Toc134720516)

**Стратегия**

Листинг программы:







import random

from abc import ABC, abstractmethod

class Variant(ABC):

    @abstractmethod

    def selection(self) -> None:

        pass

class Rock(Variant):

    def selection(self) -> str:

        return "Камень"

class Chik(Variant):

    def selection(self) -> str:

        return "Ножницы"

class Paper(Variant):

    def selection(self) -> str:

        return "Бумага"

class Random(Variant):

    def selection(self) -> str:

        options = ["Камень", "Ножницы", "Бумага"]

        return random.choice(options)

class Game:

    strategy: Variant

    def \_\_init\_\_(self, strategy: Variant = None) -> None:

        if strategy is not None:

            self.strategy = strategy

        else:

            self.strategy = Random()

    def play(self, sec2) -> None:

        s1 = self.strategy.selection()

        s2 = sec2.strategy.selection()

        if s1 == s2:

            print("Ничья!")

        elif s1 == "Камень":

            if s2 == "Ножницы":

                print("Побеждает первый игрок!", s1, '>', s2)

            elif s2 == ("Бумага"):

                print("Побеждает второй игрок!", s1, '<', s2)

        elif s1 == "Бумага":

            if s2 == "Ножницы":

                print("Побеждает второй игрок!", s1, '<', s2)

            elif s2 == ("Камень"):

                print("Побеждает первый игрок!", s1, '>', s2)

        elif s1 == "Ножницы":

            if s2 == "Камень":

                print("Побеждает второй игрок!", s1, '<', s2)

            elif s2 == ("Бумага"):

                print("Побеждает первый игрок!", s1, '>', s2)

def playtime(vibor):

    if vibor == "Бумага":

        return Game(Paper())

    elif vibor == "Ножницы":

        return Game(Chik())

    elif vibor == "Камень":

        return Game(Rock())

    elif vibor == "Рандом":

        return Game(Random())

n=1

while n==1:

    print("Первый игрок:\nСделайте выбор: Камень, Ножницы, Бумага, Рандом")

    vibor=input()

    while vibor not in ("Бумага", "Камень", "Ножницы", "Рандом"):

        print("Повторите ввод")

        vibor=input()

    player1 = playtime(vibor)

    print("Второй игрок:\nСделайте выбор: Бумага, Камень, Ножницы, Спок, Ящерица, Рандом")

    vibor = input()

    while vibor not in ("Бумага", "Камень", "Ножницы", "Спок", "Ящерица", "Рандом"):

        print("Повторите ввод")

        vibor=input()

    player2 = playtime(vibor)

    player1.play(player2)

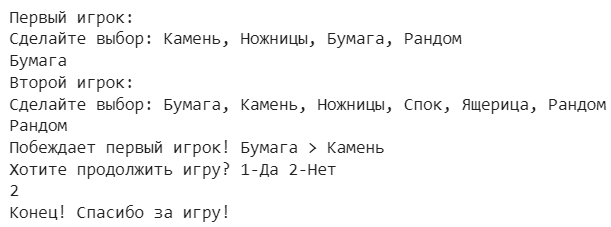
    print("Хотите продолжить игру? 1-Да 2-Нет")

    n=int(input())

if n==2:

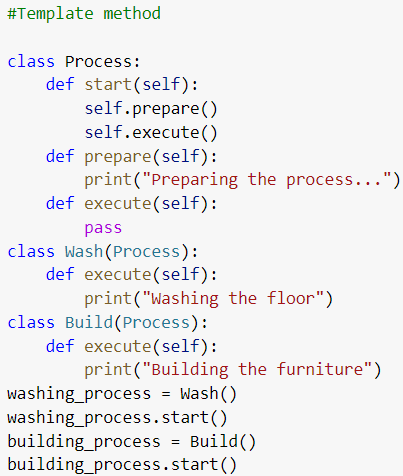
  print("Конец! Спасибо за игру!")

Результат работы алгоритма:



**Шаблонный метод**

Листинг программы:



#Template method

class Process:

    def start(self):

        self.prepare()

        self.execute()

    def prepare(self):

        print("Preparing the process...")

    def execute(self):

        pass

class Wash(Process):

    def execute(self):

        print("Washing the floor")

class Build(Process):

    def execute(self):

        print("Building the furniture")

washing\_process = Wash()

washing\_process.start()

building\_process = Build()

building\_process.start()

Результат работы алгоритма:

