|  |
| --- |
| МБОУ Лицей №17 |
| Индивидуальный итоговый проект |
| Моделирование жизни одноклеточных организмов |
|  |
| **Подготовили: Белоконь Даниил, Соколов Артём** |
| **Под руководством: Виноградовой Юлии Николаевны** |

|  |
| --- |
|  |

**Кострома, 2020 год**

Оглавление

[Введение 3](#_Toc33607195)

[Концепция 3](#_Toc33607196)

[Актуальность 3](#_Toc33607197)

[Польза 3](#_Toc33607198)

[Цель 3](#_Toc33607199)

[Задачи 3](#_Toc33607200)

[Глава I. Теоретическая часть. 5](#_Toc33607201)

[Устройство клетки 5](#_Toc33607202)

[Деление клетки 6](#_Toc33607203)

[Выполнение действий 6](#_Toc33607204)

[Выбор языка программирования 7](#_Toc33607205)

# Введение

## Концепция

Идея проекта заключается в создании небольшого приложения, способного по заданной в её коде модели симулировать развитие одноклеточных организмов (ботов). Организмы живут по программе, прописанной в их «генном коде», массиве чисел, где каждое значение подразумевает то или иное действие, которое необходимо выполнить. В процессе размножения могут происходить случайные изменения в списке команд ботов – мутации. Вследствие постоянной смены поколений должны оставаться только те экземпляры, которые лучше всего подходят под данные условия и способны дать наибольшее количество потомков.

## Актуальность

Футуристы задумывались о подобных симуляциях достаточно давно, но технологии не позволяли обрабатывать большое количество информации, чтобы реализовать эти мечты. В наши дни до сих пор не представляется возможным предусмотреть все известные человеку процессы: малейшие округления способны привести к большим изменениям в предсказаниях учёных. Это называется эффектом бабочки или эффектом крыла бабочки[[1]](#footnote-1). Тем не менее, общие модели уже существуют и активно применяются практически во всех сферах деятельности человека.

## Польза

Данная идея при любой реализации способна стать наглядным примером эволюции, что при профессиональном исполнении может использоваться в качестве ознакомительной площадки для знакомства с упрощёнными принципами развития организмов. Добавление возможности редактирования стартовых условий пользователем позволит проводить домашние эксперименты с моделью мира. В свою очередь, это увеличит понимание изменчивости жизни, тонкость и сложность её настройки.

Для нас, как учеников 9 класса, данный проект является отличным опытом в сфере программирования, так как это большая работа по моделированию реальных объектов и их функций. Каждая пропущенная деталь ведёт к большому обобщению результатов работы программы и как следствие – ухудшению результата.

## Цель

Разработка и реализация упрощённой модели мира, одноклеточных организмов и принципов эволюции.

## Задачи

Поскольку работа по проектированию сложных биологических систем достаточно объёмна, учитывая то, что после создания концепции необходимо всё оформить в программном коде, было принято решение разделять работу по мере появления новых задач и совместной работой над ошибками. По итогу вышло так, как описано ниже.

Белоконь Даниил:

* Работа с самим проектом:
  + Структурирование файлов. Оформление. Создание документации.
  + Выведение результатов на экран.
* Моделирование мира:
  + Смена времён года.
  + Зонирование по уровням света.
  + Изменение условий при смене сезонов.
  + Организация хранения данных.
  + Смена ходов.
  + Получение энергии клетками.
  + Создание клетки и взаимодействий между ними:
    - Поедание.
    - Размножение.
    - Перемещение.
    - Передача энергии.
    - Осмотр территории.
    - Обновление состояния.
    - Мутации.
    - Выявление близкой родственности между ботами.
    - Обработка генов.
    - Смена направлений.
    - Проверка направлений и списка на возможность проведения операций.
    - Фотосинтез.

Соколов Артём:

* Работа с проектом:
  + Исправление ошибок, возникающих в коде.
  + Тестирование.
  + Работа с миром:
    - Разделение на зоны для получения минералов.
    - Передача минералов клеткам.
    - Работа с клетками:
      * Переработка минералов.
      * Переработка трупов ботов.

Кроме того ещё на моменте изучения теории было начато писание отдельных функций и частей кода, которые будут приведены в пример ниже. Это позволило сэкономить время, так как эти концепции были уже использованы в самой программе.

# cells-animal-plant-ways-nucleus-difference-organelles.jpg[[2]](#footnote-2) Глава I. Теоретическая часть.

Для реализации задуманной идеи понадобятся знания, как в области биологии, так и в сфере программирования. Прежде чем преступать к моделированию мира, необходимо изучить некоторые процессы в одноклеточных организмах.

## Устройство клетки

Во-первых, необходимо выяснить устройство клетки и решить, что именно нужно перенести в программируемую модель. Для начала стоит вспомнить то, что клетки бывают нескольких видов. В наше время выделяют основные 3: бактериальные, животные и растительные.

Животные и растительные клетки находятся в структурах многоклеточных организмов и не могут выполнять большое количество отдельных задач, будучи отделенными друг от друга. Поэтому к целям проекта подходят бактерии. Они имеют малые размеры и живут независимо друг от друга.

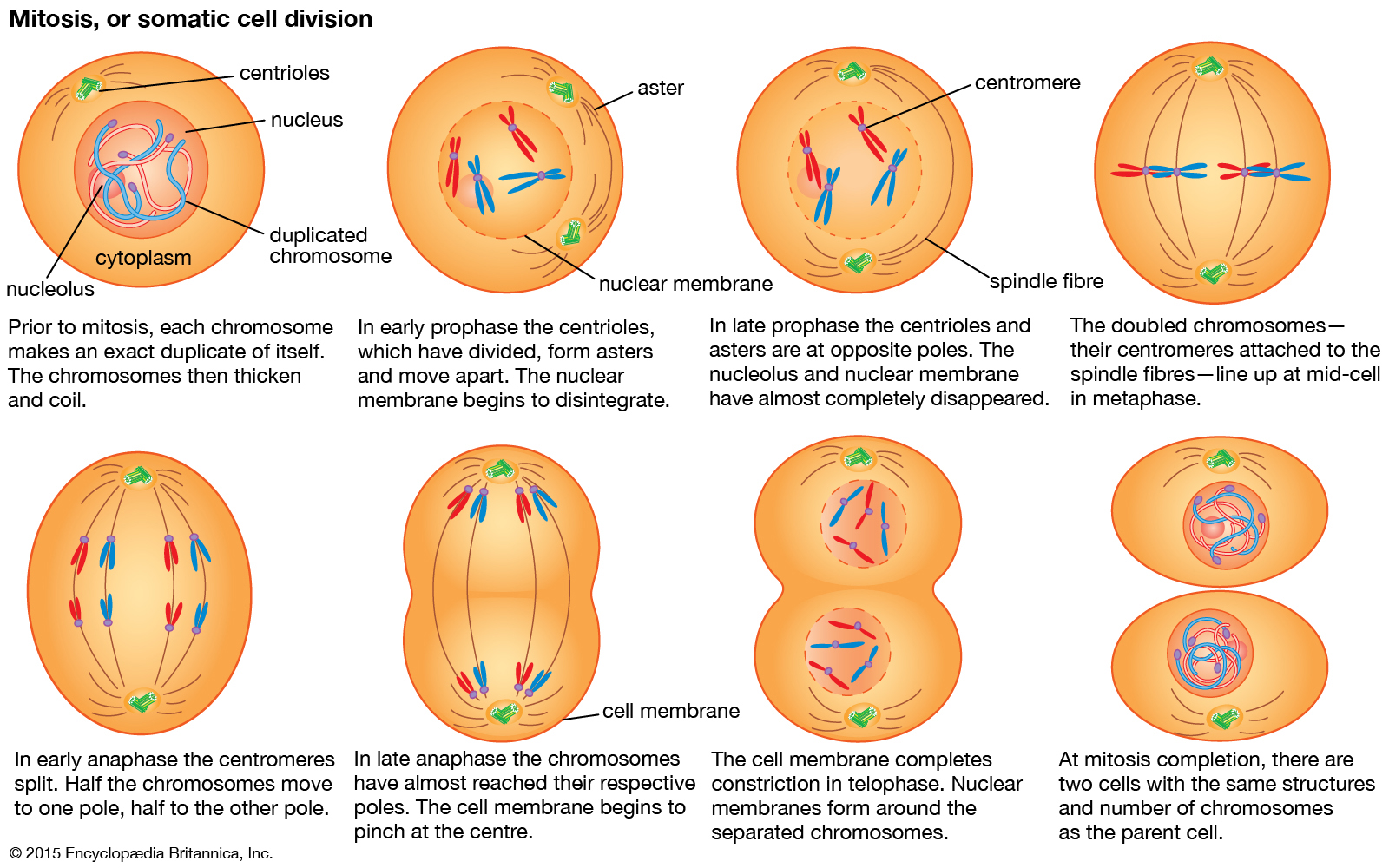
Моделирование и изменение инициализации клетки происходило на протяжении всей работы с кодом, в итоге получился подобный конечный вариант.



Клетка в проекте обладает энергией, значение которой или заставляет клетку размножаться, или умереть при её недостатке; минералами, которые заменяют разные химические элементы, которыми клетки могут питаться; а также самим генетическим кодом, который представляет из себя массив чисел. Остальные параметры используются для работы самой программы и обеспечения процесса развития и действий.

## Деление клетки

Во-вторых, это само деление клетки. Если обобщить, то хромосомы дублируются, затем выстраиваются посередине родительской клетки, после чего расходятся в дочерние организмы. Именно на этих этапах возникают одни из самых серьёзных мутаций: происходят ошибки при копировании геномного кода или нарушается количество хромосом (в одной клетке хромосом становится больше, в другой – меньше). На картинке снизу изображено деление не бактериальной клетки, тем не менее, процесс достаточно схож.

[[3]](#footnote-3)

Для упрощения этого процесса было решено проводить деление таким образом: сначала из родительской клетки вычитается 150 энергии. Затем со случайной стороны создаётся базовая клетка, в которую передаётся половина минералов и энергии материнского бота, а затем копируется генный код. С шансом в 25% в генном коде дочерней клетки происходит мутация – просто случайное изменение в случайном месте массива. Функция размножения находится в приложении.

## Выполнение действий

В реальности каждый ген отвечает за выработку определённых белков, а действия совершаются уже вследствие их комбинаций и сложных химических процессов. Ради упрощения принципов работы ДНК её функция изменилась с содержания информации об отдельных белках до хранения чисел, каждое из которых означает одно из действий, которое необходимо выполнить.



При каждом обновлении состояния клетки (при смене хода) происходит выполнение действия, обозначенного курсором в массиве чисел. К некоторым (а в перспективе и ко всем) значениям приравнено выполнение определённых функций.

Некоторые являются простыми, и после них действие передаётся следующей клетке. Однако остальные взаимодействуют с последующими «генами». Они получают их значения и работают в зависимости от аргументов. К примеру, такие функции, как перемещение и осмотр местности делят следующее за генами их вызвавшими число на 8, берут остаток и взаимодействуют с той стороной, которая соответствует этому числу. Кроме того, каждая функция смещает курсор на определённое количество единиц.

## Выбор языка программирования

В мире существует большое количество не только естественных языков, но и языков программирования. Практически каждый создаётся с какой-то определённой целью и имеет свою специфику. Это необходимо для выполнения различных задач. К примеру, одним из самых низкоуровневых языков (т.е. находящихся ближе всего к физическому уровню) является ассемблер. Языками более высокого уровня является, к примеру, С (Си) и его «потомки»: С# и C++. Существует ещё очень много критериев разделения языков программирования: объектно-ориентированность, функциональность и т. д.

Для нашего проекта мы использовали высокоуровневый (близкий к пониманию человеком), интерпретируемый (выполняемый «пошагово») язык программирования Python (питон). Это связано с тем, что его достаточно просто изучить, он имеет лёгкий синтаксис. Вместе с тем, к нему написано большое количество библиотек (подключаемых модулей), которые позволяют легко работать с графической составляющей проекта.

Конкретный графический модуль, который был выбран – это Pygame. Его преимущество – возможность в несколько строчек кода выводить желаемое на экран, а затем работать с результатом.

Сам Python имеет возможность работать как с функциями и классами, так и с простым исполняемым кодом. Поэтому на нём можно запрограммировать как мир и клетку, так и отдельные куски кода, помогающие выполнять программу.

# Глава II. Реализация.

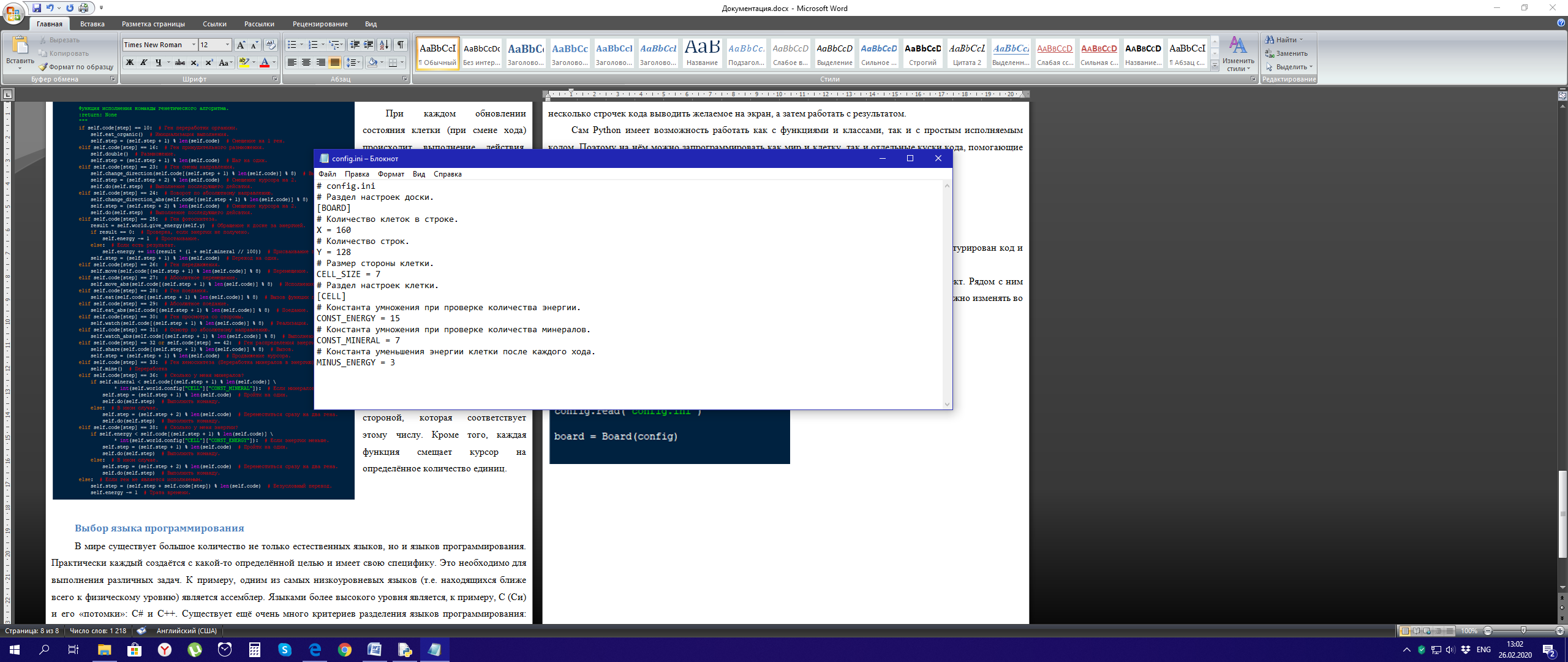
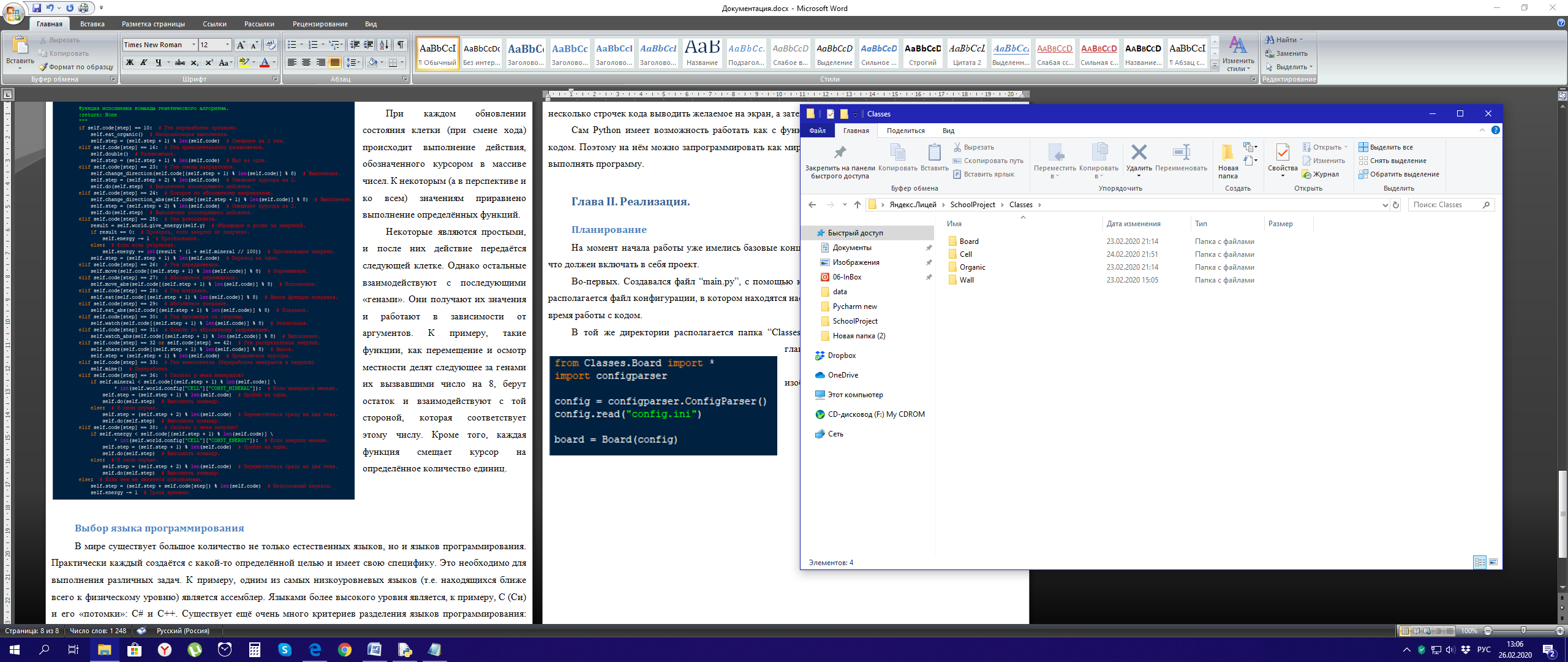
## Планирование

На момент начала работы уже имелись базовые концепции того, как должен быть структурирован код и что должен включать в себя проект.

Во-первых, был создан файл “main.py”, с помощью которого запускалась остальная программа. Рядом с ним располагается файл конфигурации, в котором находятся настройки для программы, которые можно изменять во время работы с кодом.

В той же директории располагается папка “Classes” где хранятся классы, которые импортируются в главный файл или в другие объекты (между собой).

В итоге вышло так, как показано на данных изображениях.



В начале работы с кодом были выработаны правила оформления, которые должны соблюдаться на протяжении написания всего проекта. А именно: комментирование каждой функции и каждого класса, пояснение каждой строчки, старание уместиться в 120 символов в одной линии.

## Написание кода

Затем был реализован класс доски, включающий в себя вывод изображений, смену ходов и времён года, передачу энергии и минералов, а также массив данных, в котором находятся клетки, стены и органика (мёртвые организмы).

В процессе написания этого класса использовалась библиотека Pygame. По мере необходимости приходилось обращаться к официальной документации[[4]](#footnote-4). В проекте использовалась лишь небольшая часть того, что может в целом данный модуль.

Класс клетки является самым трудоёмким, так как именно в нём происходят все процессы, ведущие к размножению, мутациям и как следствие – эволюции. Поэтому после оформления задумок разработка началась именно с него, а функции в остальных файлах создавались лишь по мере необходимости.

Органика и стена были добавлены в процессе написания самой клетки. Органика означает умерших ботов и постепенно падает вниз. Стена – это ограничение по перемещению, которые наносят урон организмам, если не попытаются их съесть.

1. Эффект бабочки или эффект крыла бабочки – свойство некоторых систем, в которых малейшие изменения в переменных могут приводить к большим изменениям в результатах. Исторические заметки на эту тему: https://www.wolframscience.com/reference/notes/971c [↑](#footnote-ref-1)
2. Картинка и информация были взяты из статьи: <https://www.britannica.com/science/cell-biology> [↑](#footnote-ref-2)
3. Иллюстрации к делению клетки и сам материал были взяты из образовательной статьи: <https://www.britannica.com/science/cell-biology/Meiosis> [↑](#footnote-ref-3)
4. Документация по Pygame: https://www.pygame.org/docs/ [↑](#footnote-ref-4)