

Руководство по созданию кроссплатформенной консольной игры «Морской бой» на Python

для школьников

Оглавление

Глава 1. Основы консольной разработки на Python.....	3
1.1. Рекомендуемые темы и материалы	3
1.2. Установка и настройка Python.....	5
1.3. Интегрированные среды разработки	9
1.4. Знакомство с первым консольным приложением	11
1.5. Запуск консольного приложения	12
Глава 2. Реализация кроссплатформенной игры Морской бой.....	14
2.1. Формирование правил и алгоритмов работы приложения.....	14
2.2. Создание консольного интерфейса	15
2.3. Реализация функциональной логики игры.....	18
2.4. Возможности по дальнейшему расширению функционала	21
2.5. Тестирование работоспособности.....	22
2.6. Размещение материалов в публичном репозитории	23
Список литературы.....	24

Глава 1. Основы консольной разработки на Python

1.1. Рекомендуемые темы и материалы

Если вы чувствуете себя уверенно в основах Python, можно смело пропустить эту главу.

В 2024 году языку Python версии 1.0 уже исполнилось 30 лет, но он по-прежнему не потерял свою актуальность и продолжает активно развиваться. Базовые конструкции и подходы в Python напоминают множество других известных языков высокого уровня, таких как Бейсик, Паскаль, Си, Java, JavaScript и их производных, таким образом, если у вас есть опыт использования любого из них, это поможет быстрее приступить к реализации практических задач. В тоже время, как показывает практика, изучения Python «с нуля» не вызывает особых проблем.

Для полноценного использования Python при реализации консольной игры, рекомендуется изучить следующие темы:

1. Основные конструкции языка – переменные, функции, числа, строки, арифметические операции, операторы сравнения, логические операторы, приоритеты операций, условные и циклические операторы, функции вывода на консоль и чтения с консоли, функции работы со строками и числами.
2. Структуры данных – списки, словари, множества, кортежи.
3. Объектно-ориентированное программирование (ООП) – классы, объекты и их создание, атрибуты, методы, магические методы (`__init__`, `__repr__`).
4. Исключения и их обработка.
5. Модули `random`, `os`, `enum` – методы генерации случайных чисел, способы определения текущего типа операционной системы (ОС), методы очищения экрана консоли.

В интернете есть большое количество материалов для людей, имеющих желание начать свой путь в программировании на языке Python – всевозможные курсы, книги и самоучители. Давайте рассмотрим некоторые из них, их плюсы и минусы.

Например, официальная документация Python [10].

Плюсы:

- Вся актуальная информация в одном месте, от разработчиков языка.
- Достаточно подробное описание, есть примеры кода.
- Удобно использовать как справочное пособие, глоссарий, выбор версии языка и функция быстрого поиска позволяют быстро найти требуемое.
- Можно подобрать свой темп обучения.

Минусы:

- Сайт только на английском языке.
- Нет задач для практики.
- Без плана обучения можно потеряться в обилии информации.

К сожалению, официальная документация не так хорошо подходит для изучения Python, как различные курсы или тренинги. Чаще всего, онлайн или очные курсы являются наиболее быстрым и эффективным способом изучения языка.

Плюсы курсов:

- Чёткий план обучения.
- Подробно объясняется материал на родном языке, как правило присутствуют консультации.
- Практические задания с проверкой.
- Есть индивидуальный подход.
- Выдаётся сертификат об окончании, что в дальнейшем может помочь с поступлением в высшее учебное заведение или устройством на работу.

Минусы курсов:

- Обучение занимает много времени.
- Не всегда бесплатно.

Из бесплатного, для школьников 8–11 классов, можно порекомендовать курсы по языку программирования Python в рамках проекта «Код будущего», который организован под эгидой федерального проекта «Развитие кадрового потенциала ИТ-отрасли» национальной программы «Цифровая экономика». В проекте участвует большинство регионов России, предоставляются как онлайн, так и очная формы обучения. Более подробную информацию можно получить на портале Госуслуги [5].

Также существует огромное количество книг по языку Python. Вот на которые стоит обратить внимание начинающим программистам:

- Доусон М. Програмируем на Python. 3-е изд. [1]. Руководство написано опытным разработчиком и преподавателем, научит фундаментальным принципам программирования на примере создания простых игр. В книге приведено множество примеров кода, а в конце каждой главы найдётся проект полноценной игры.
- Бэйдер Д. Знакомство с Python [2]. Книга построена по принципу 80/20: большую часть полезной информации можно усвоить, изучив несколько критически важных концепций. Освоив самые популярные команды и приемы, можно сразу сосредоточиться на решении реальных задач.

- Седер Наоми. Python. Экспресс-курс. 3-е изд. [3]. С помощью этой книги возможно быстро перейти от основ к управлению и структурам данных, чтобы создавать, тестировать и разворачивать полноценные приложения. Автор рассказывает не только об основных особенностях языка Python, но и его объектно-ориентированных возможностях, которые появились в Python 3.

1.2. Установка и настройка Python

Python можно установить на большое количество различных операционных систем (ОС), таких как Windows, Linux/UNIX, MacOS и т.д. Так как Windows пока ещё является наиболее популярной ОС, давайте рассмотрим пошаговую установку последней версии Python именно на эту платформу.

1. Откройте браузер и перейдите на официальный сайт Python (<https://www.python.org/>) в соответствии с рисунком 1.

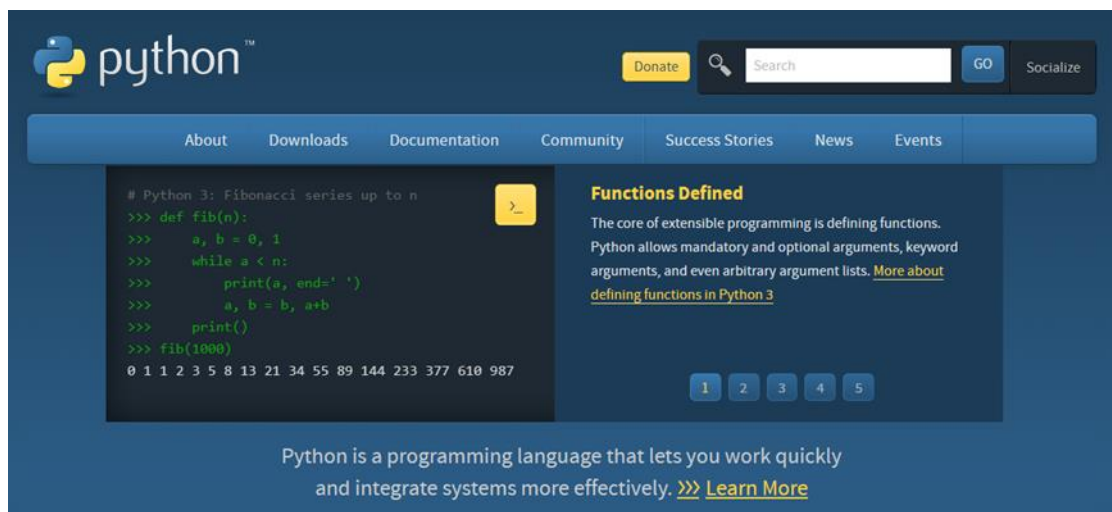


Рисунок 1. Официальный сайт Python

2. Наведите мышь на кнопку «Downloads» и выберите вашу ОС (в нашем случае Windows).



Рисунок 2. Меню выбора дистрибутива Python для скачивания

3. Нажмите на ссылку «Latest Python 3 Release».

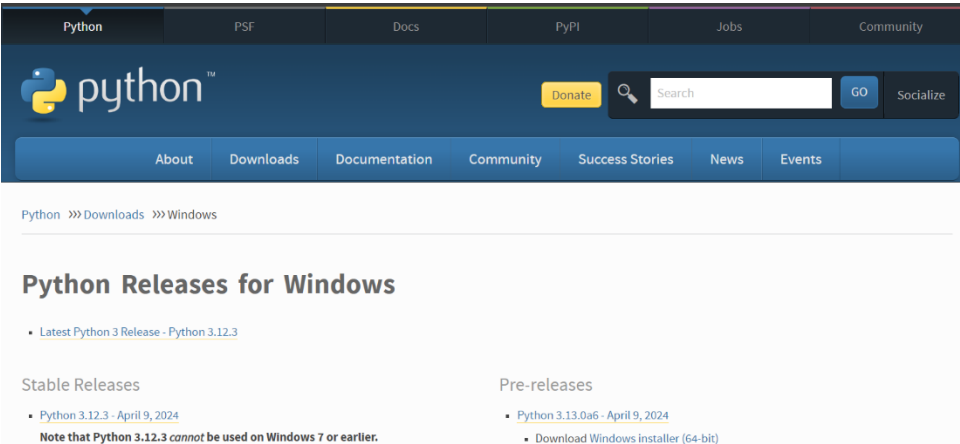


Рисунок 3. Страница для скачивания дистрибутив Python

4. Прокрутите страницу вниз и нажмите на нужную версию Python, чтобы началось скачивание установщика. Версия дистрибутива зависит от вашего центрального процессора, и в большинстве случаев это «Windows installer (64-bit)». Будьте внимательны, некоторые версии Python не будут работать на старых версиях Windows.

Files							
Version	Operating System	Description	MDS Sum	File Size	GPG	Sigstore	SBOM
Gzipped source tarball	Source release		3c5498a34d5226c9b746b1199f0bf2d9	25.9 MB	SIG	sigstore	SPDX
XZ compressed source tarball	Source release		8defb33f0c37aa4bdd3a38ba52abde4e	19.7 MB	SIG	sigstore	SPDX
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	6114a3bb9b288f23ab38dbb959be1bf	43.6 MB	SIG	sigstore	
Windows installer (64-bit)	Windows	Recommended	c86949710e0471a065db970290819489	25.5 MB	SIG	sigstore	
Windows installer (ARM64)	Windows	Experimental	ef016521b5a147f3ded730801d36a350	24.7 MB	SIG	sigstore	
Windows embeddable package (64-bit)	Windows		38cce2bf5b4de76db19a31f46a0720de	10.5 MB	SIG	sigstore	
Windows embeddable package (32-bit)	Windows		65d873c723db66d6746e9872df5a715e	9.4 MB	SIG	sigstore	
Windows embeddable package (ARM64)	Windows		3229271cac55f913aad1bb46ebc9931	9.8 MB	SIG	sigstore	
Windows installer (32-bit)	Windows		a95c4fbdce0b6a22ca7cfb450f57c616	24.2 MB	SIG	sigstore	

Рисунок 4. Страница для скачивания установщика Python

5. Запустите скачанный установщик и убедитесь, что окно установки похоже на картину ниже. Если у вас установка выглядит иначе, необходимо сперва удалить установленную ранее версию Python.

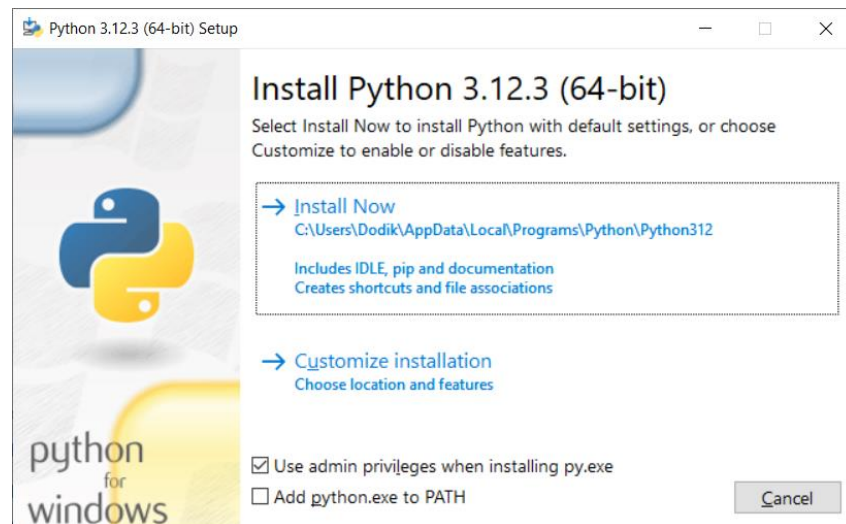


Рисунок 5. Установщик Python

6. Поставьте дополнительную галочку в пункте чекбокса «Add python.exe to PATH». Это необходимо, чтобы в консоли заработала команда python (запустится терминал языка Python), и другие приложения смогли корректно работать с установленным дистрибутивом.

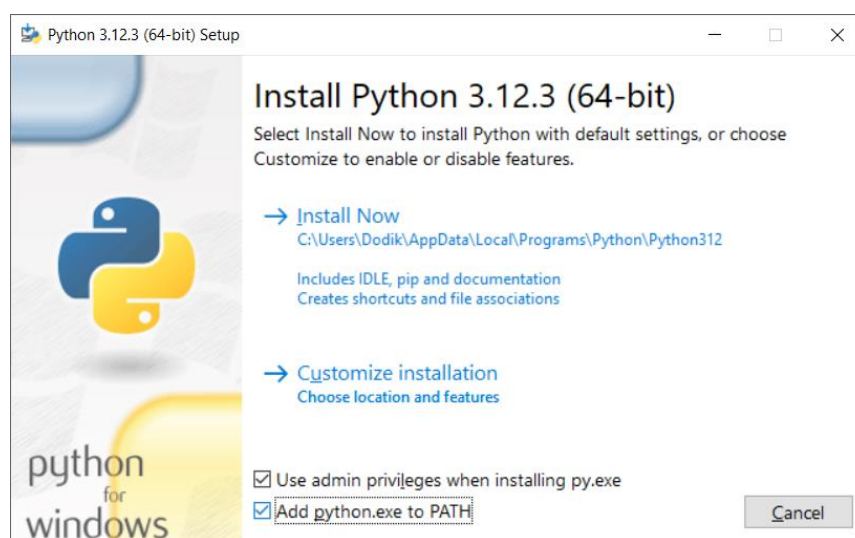


Рисунок 6. Чекбокс «Add python.exe to PATH»

7. Нажмите «Install Now» и дождитесь завершения работы инсталлятора.

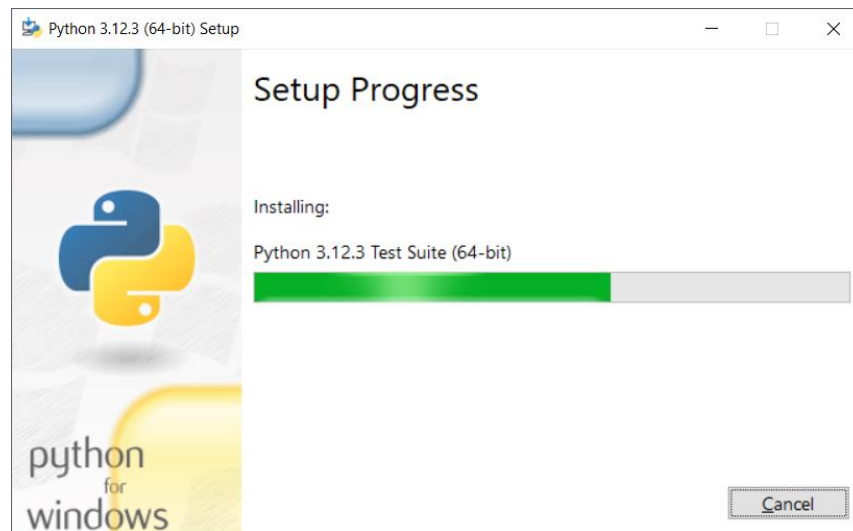


Рисунок 7. Процесс установки Python

8. После успешного завершения нажмите кнопку «Close».

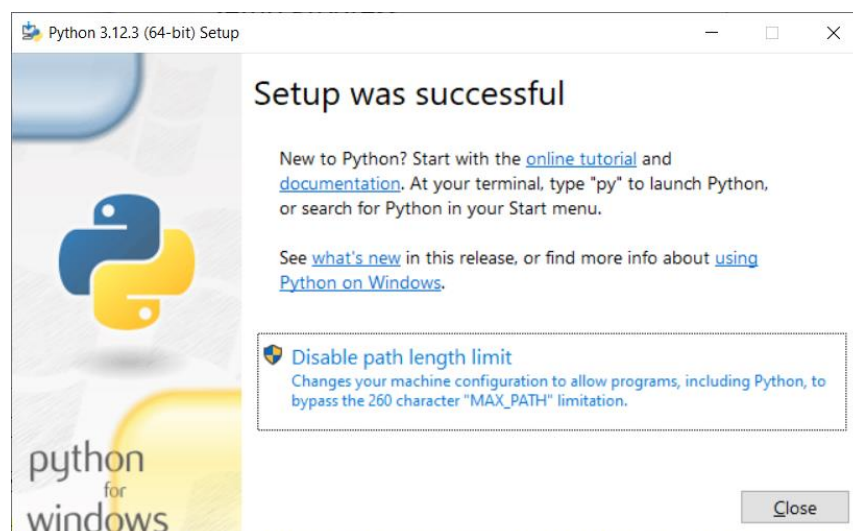


Рисунок 8. Окно успешного завершения установки Python

9. Для проверки правильности установки дистрибутива, откройте консоль Windows (команда cmd). Напишите в ней команду python. Запустится Python в интерактивном режиме (см. Рисунок 9).

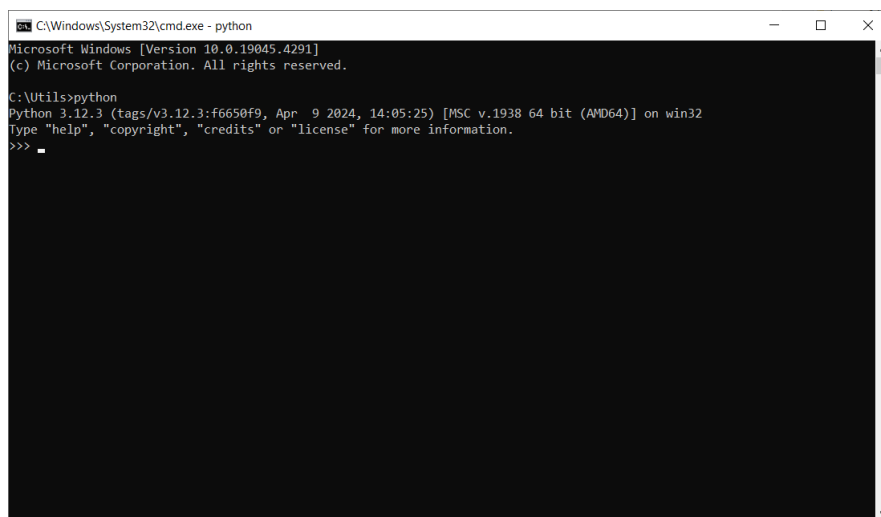


Рисунок 9. Терминал Python в интерактивном режиме

10. Введите команду `exit()`. Она завершит работу терминала Python.

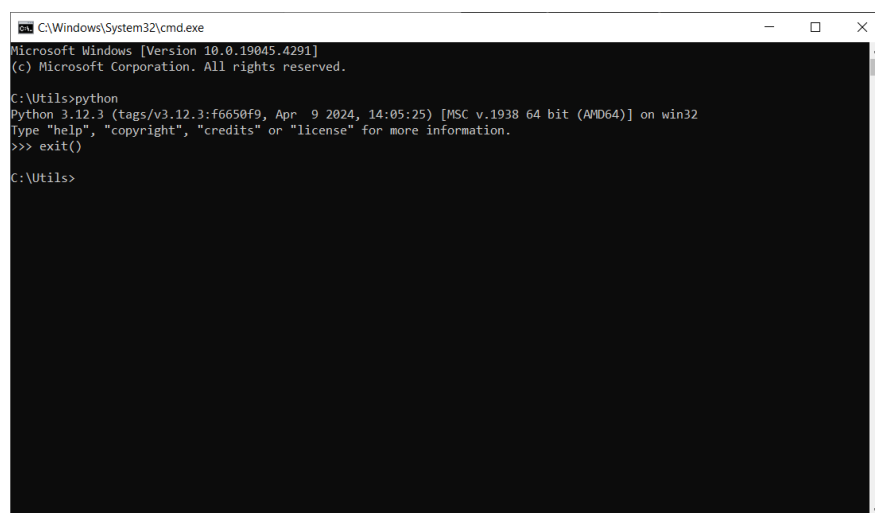


Рисунок 10. Завершение работы терминала Python

После успешной проверки, можно начинать использование языка Python.

1.3. Интегрированные среды разработки

Есть множество способов писать и запускать код на Python. Пожалуй, самый неудобный из них, это писать код в приложении «Блокнот» и запускать программу через консоль Windows. Гораздо удобнее использовать специализированные программы – Integrated Development Environment (IDE), или сайты для написания кода, которые подсветят синтаксис и помогут понять, где программист допустил ошибку. Давайте рассмотрим некоторые из них.

- Google Colab (<https://colab.research.google.com/>) – сервис от компании Google, позволяющий писать код прямо в браузере. Подходит для множества задач, вообще не нагружает ваш компьютер.

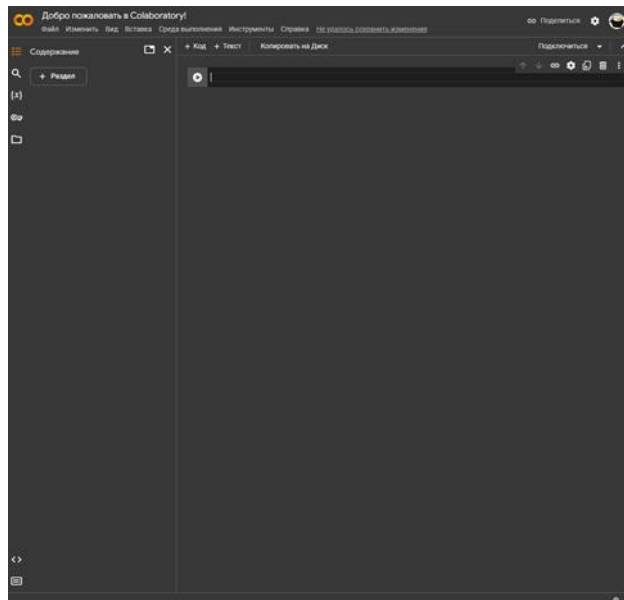


Рисунок 11. Google Colab IDE

- PyCharm (<https://www.jetbrains.com/ru-ru/pycharm/>) – одна из лучших IDE для написания кода на Python. Имеет две основные версии: PyCharm Community Edition – бесплатная версия, и PyCharm Professional Edition – платная профессиональная версия, с расширенным функционалом.

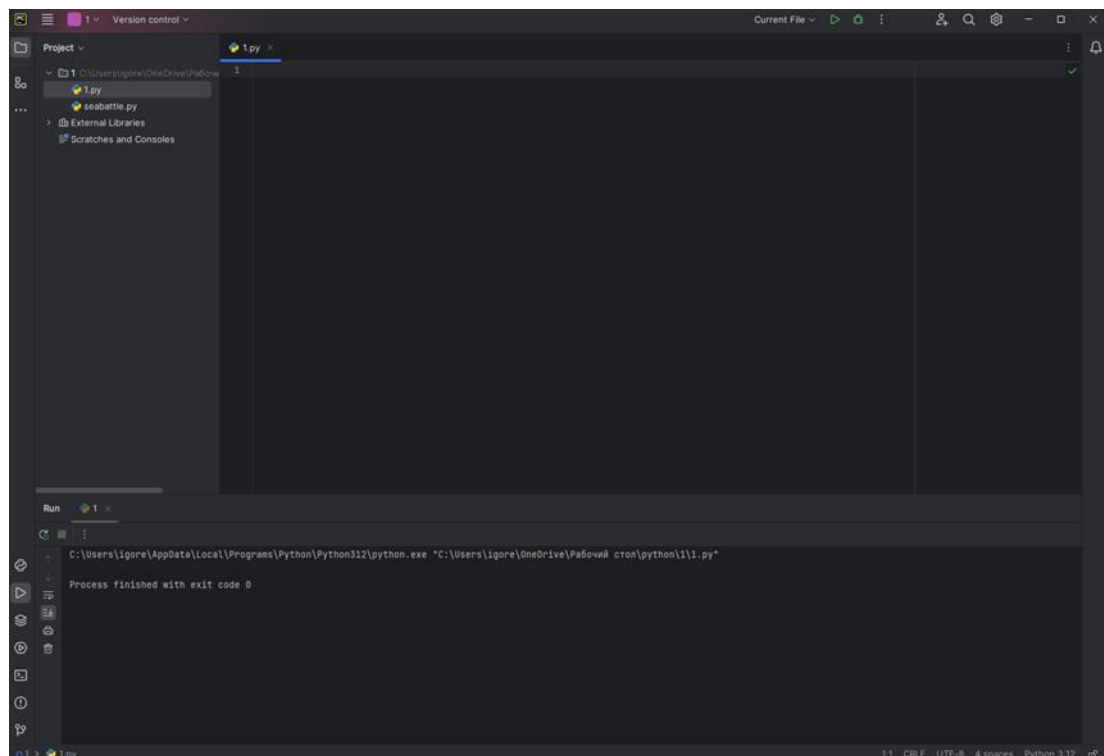


Рисунок 12. PyCharm IDE

- IDLE – Python’s Integrated Development and Learning Environment, входит в дистрибутив Python и устанавливается «из коробки» по умолчанию. Не такая удобная как PyCharm.

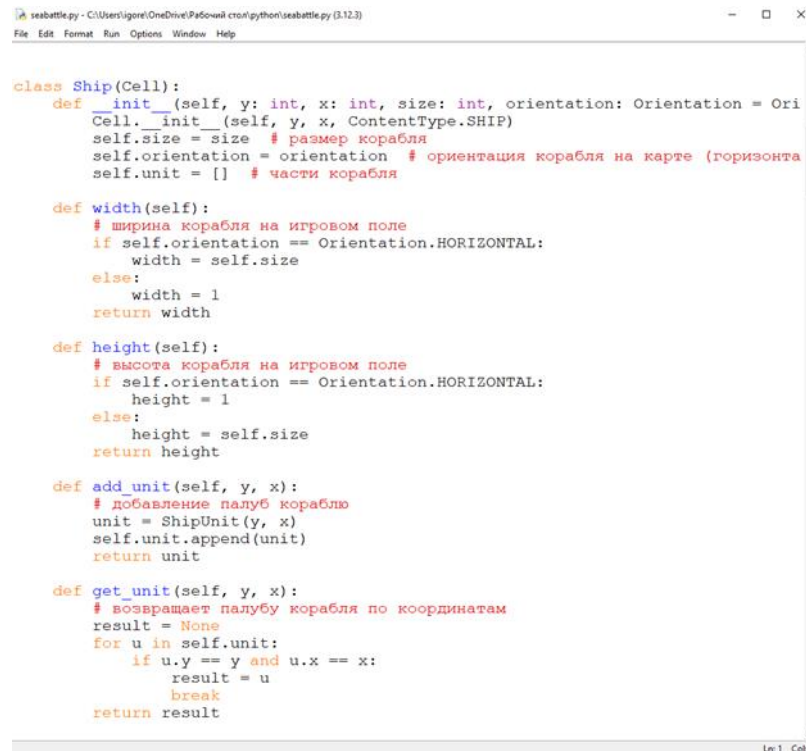


Рисунок 13. IDLE

1.4. Знакомство с первым консольным приложением

Так как в нашей игре будет задействовано объектно-ориентированное программирование (ООП), в качестве первого приложения возьмём простой, но вместе с тем, объектно-ориентированный код, который позволит вывести сообщение в консоль.

1. Для начала нужно создать основной класс приложения SayHello.

```

1 class SayHello:
2
3

```

Рисунок 14. Класс SayHello

2. Используя ключевое слово def, создадим метод hello(), который будет выводить текст в консоль.

```

1 class SayHello:
2     def hello(self):
3

```

Рисунок 15. Метод hello()

3. Используем функцию print() для вывода сообщения «Hello!» в консоль.

```

1 class SayHello:
2     def hello(self):
3         print('Hello!')

```

Рисунок 16. Функция print()

4. Создаём экземпляр класса SayHello и сохраняем полученный объект в глобальную переменную.

```

1 class SayHello:
2     1 usage
3     def hello(self):
4         print('Hello!')
5
6 h = SayHello()

```

Рисунок 17. Создание объекта от класса SayHello

5. Вызовем метод hello() у нашего объекта в переменной h.

```

1 class SayHello:
2     1 usage
3     def hello(self):
4         print('Hello!')
5
6 h = SayHello()
7 h.hello()

```

Рисунок 18. Вызовем метод hello()

1.5. Запуск консольного приложения

Мы уже знаем, как установить Python, какое IDE использовать и даже, что написать в своём первом приложении. Но как же его запустить?

Чтобы запустить код в PyCharm необходимо:

1. Выбрать Current File.

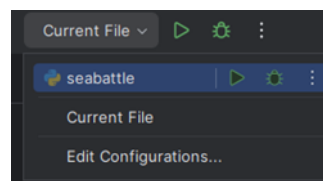


Рисунок 19. Выбор класса для запуска в PyCharm

2. Нажать на зелёную кнопку запуска или сочетание клавиш Shift+F10.

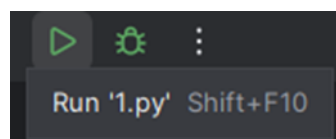


Рисунок 20. Кнопка запуска приложения в PyCharm

3. После того как вы проделали эти действия, должна открыться консоль и отобразить ошибку или вывод вашей программы.
4. Запустив программу, вы увидите в консоли сообщение «Hello!» или сообщение об ошибке, если что-то пошло не так.

```
Hello!  
Process finished with exit code 0
```

Рисунок 21. Консол вывода в PyCharm

Для запуска приложения в IDLE необходимо:

1. Сохранить открытый файл нажав на меню File> Save или комбинацию клавиш Ctrl+S.

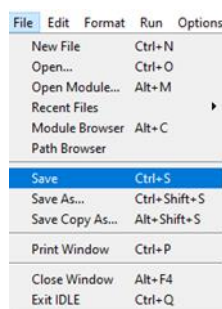


Рисунок 22. Меню сохранения файла в IDLE

2. Нажать меню Run> Run Module или клавишу F5.

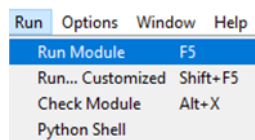


Рисунок 23. Меню запуска в IDLE

Глава 2. Реализация кроссплатформенный игры Морской бой

2.1. Формирование правил и алгоритмов работы приложения

Хотя правила игры «Морской бой» знакомы многим, стоит их вспомнить и формализовать, это поможет как при создании самого приложения, так и при его последующем тестировании.

Стандартные правила игры «Морской бой»:

- У каждого игрока есть одинаковый набор кораблей на поле 10 x 10.
- Набор кораблей включает в себя:
 - четырёхпалубный – 1 шт.,
 - трёхпалубный – 2 шт.,
 - двухпалубный – 3 шт.,
 - однопалубный – 4 шт.
- Корабли должны располагаться на расстоянии не менее одной клетки друг от друга т.е. не могут быть расположены встык.
- Палубы кораблей могут размещаться друг за другом, как по горизонтали, так и по вертикали. Диагональное размещение запрещено.
- Игроки ходят по очереди называя координаты клетки на поле противника, по которой осуществляется «выстрел».
- Если по названным координатам стоит корабль, игрок должен сообщить о результатах попадания – корабль подбит («ранен») или потоплен («убит»).
- При промахе ход переходит к другому игроку.
- Игра продолжается до тех пор, пока все корабли у одного из игроков не будут потоплены.

Правила вспомнили, перейдём к функционалу приложения и попытаемся сформировать основное техническое задание.

Программа должна:

1. Автоматически расставлять корабли из набора двух игроков на игровом поле 10 x 10 согласно вышеописанным правилам игры.
2. Выводить игровое поле в консоль.
3. Запрашивать координаты клетки «выстрела» игрока.
4. Регистрировать попадания до первого промаха.
5. Выводить координаты, по которым бьёт компьютер до первого промаха, т.к. в этой версии будет реализована игра только против компьютера/бота.
6. Выполнять шаги 2–5 пока все корабли игрока/компьютера не будут потоплены.

7. Поздравить победителя.

Для повышения удобства пользования игрой и упрощения тестирования, введём дополнительные требования. Программа должна:

- Корректно обрабатывать ввод пользователя в формате

<столбец><строка>

где <столбец> – следующие 10 букв русского алфавита не зависимо от их регистра:

А, Б, В, Г, Д, Е, Ж, З, И, К.

<строка> – номер строки от 1 до 10.

Примеры корректного ввода:

- Б1 – клетка с координатами (1, 1);
- в5 – клетка с координатами (3, 5).
- Выводить сообщение о некорректных данных, в случае ошибок игрока.
- Завершаться досрочно, в случае ввода букв В или Q не зависимо от регистра.
- Отображать имена игроков и общий счёт игры.
- Дополнительно помечать ячейки, где, согласно правилам игры, не может стоять корабль. Например, рядом с уже подбитым кораблём.

2.2. Создание консольного интерфейса

Интерфейс – это одна из основополагающих частей любого приложения, предполагающего взаимодействие с пользователем, поэтому невозможно просто оставить её без внимания. Сформулируем общие критерии к нашему интерфейсу:

- не должен быть перегружен;
- должен быть понятен и удобен;
- должен быть в едином стиле.

Итоговый результат отображён на рисунке ниже.

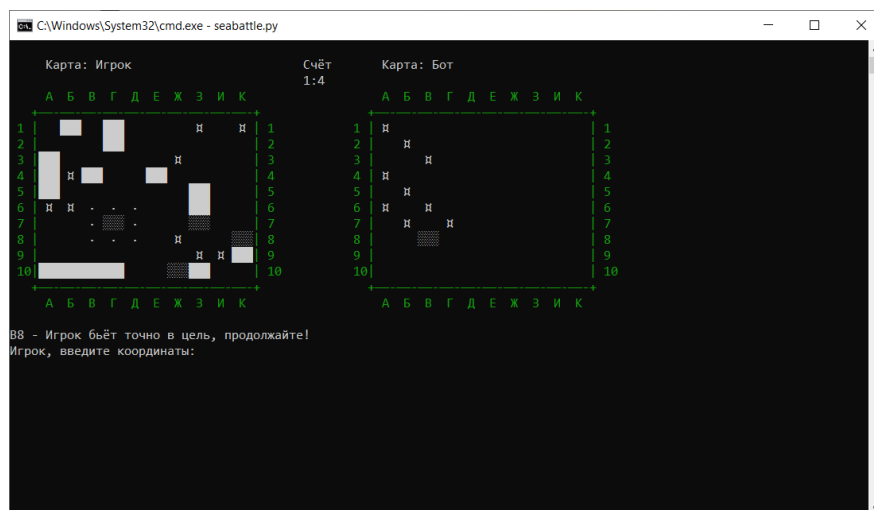


Рисунок 24. Интерфейс игры

Рассмотрим более подробно основные функциональные элементы получившегося интерфейса:

- По столбцам и строкам игровые поля размечены буквами и цифрами соответственно, чтобы пользователю было удобно ориентироваться.

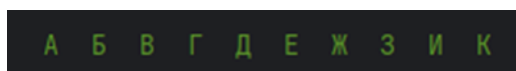


Рисунок 25. Разметка столбцов игрового поля

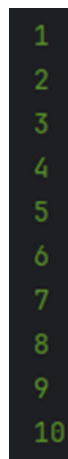


Рисунок 26. Разметка строк игрового поля

- Используются специальные символы и псевдографика для отображения «живых» и подбитых кораблей, границ поля и клеток, куда уже производился выстрел, или где не может стоять ещё один корабль (см. Рисунок 27, Рисунок 28).

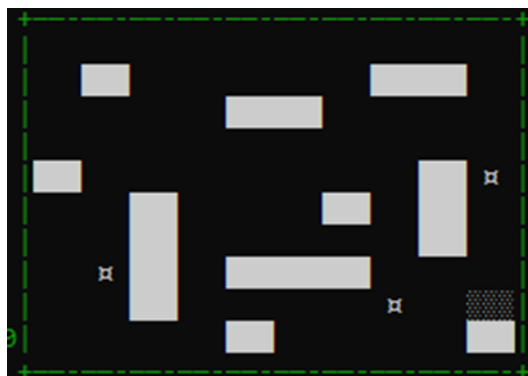


Рисунок 27. Отображение «живых» и подбитых кораблей

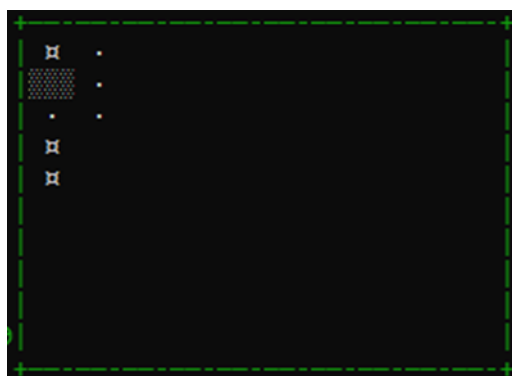


Рисунок 28. Отображение клеток с промахами и где не может стоять корабль

- Над игровым полем показан счёт подбитых кораблей, подписаны кому принадлежит какая из карт.

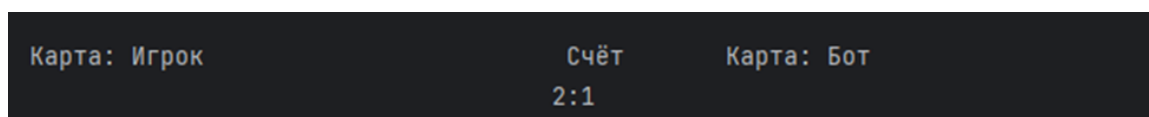


Рисунок 29. Счёт игры

- Под игровым полем находится текстовый лог промахов/попаданий и поле для ввода координат очередной ячейки.

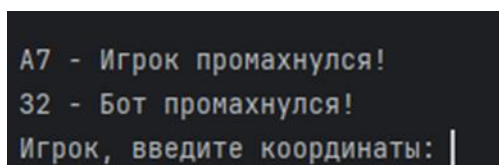


Рисунок 30. Лог игры

- Для облегчения тестирования, в алгоритм был добавлен специальный режим EASY позволяющий просматривать корабли противника.



Рисунок 31. EASY режим, позволяющий просматривать корабли противника

2.3. Реализация функциональной логики игры

При разработке приложения использовалось объектно-ориентированное программирование (ООП). Рассмотрим более подробно предназначение основных классов, их атрибутов и методов, а также работу программы в целом. Общая диаграмма классов отображена на рисунке ниже (см. Рисунок 32).

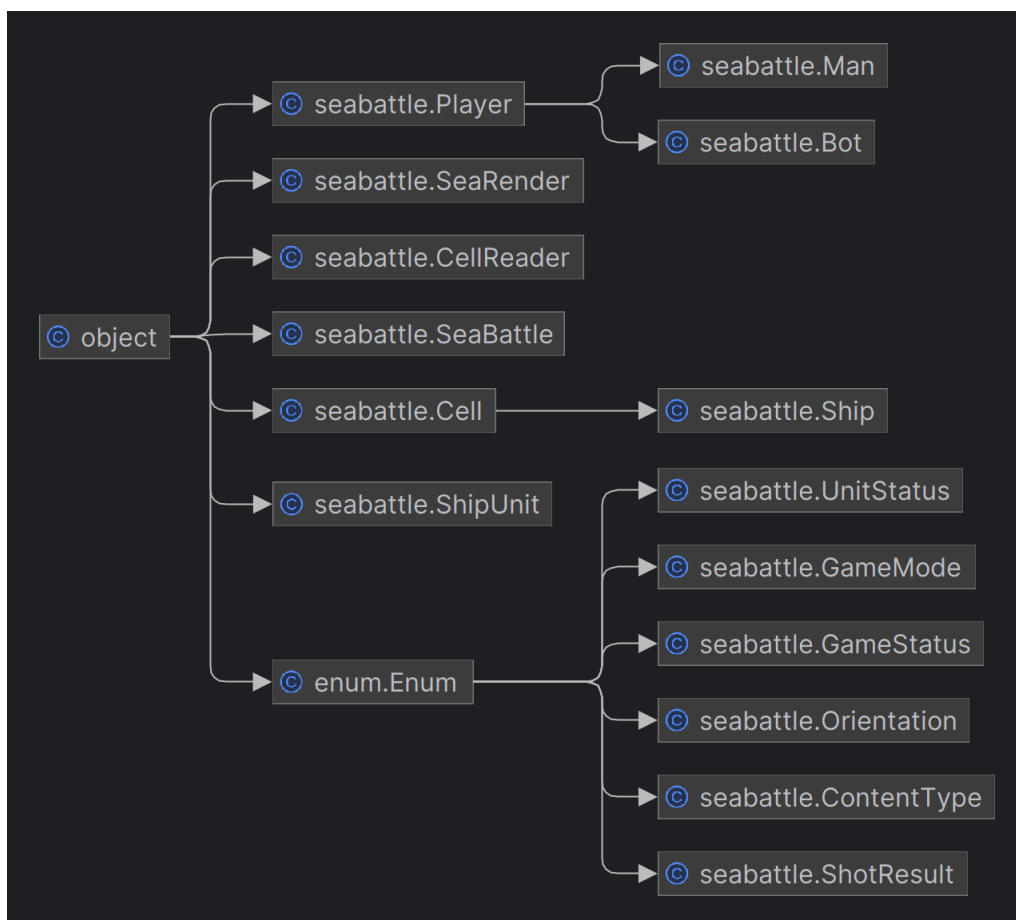


Рисунок 32. Диаграмма классов приложения

Основной класс приложения SeaBattle (см. Рисунок 33) отвечает за игру в целом. Класс содержит информацию о размерах игрового поля, возможных размерах и количестве кораблей, формирует игровое поле, создаёт и расставляет корабли каждого типа, регистрирует выстрелы, добавляет сообщения в лог игры, управляет процессом рендеринга игрового поля. Экземпляр этого класса создаётся сразу при запуске игры, при инициализации задаётся режим игры при помощи класса GameMode (см. Рисунок 34). Используя метод add_player() в игру добавляются объекты игроков, экземпляры классов Man и Bot. Таким образом можно создать игру, где между собой будут сражаться либо люди, либо боты. Подобную возможность хорошо иллюстрирует код запуска игры (см. Рисунок 35)

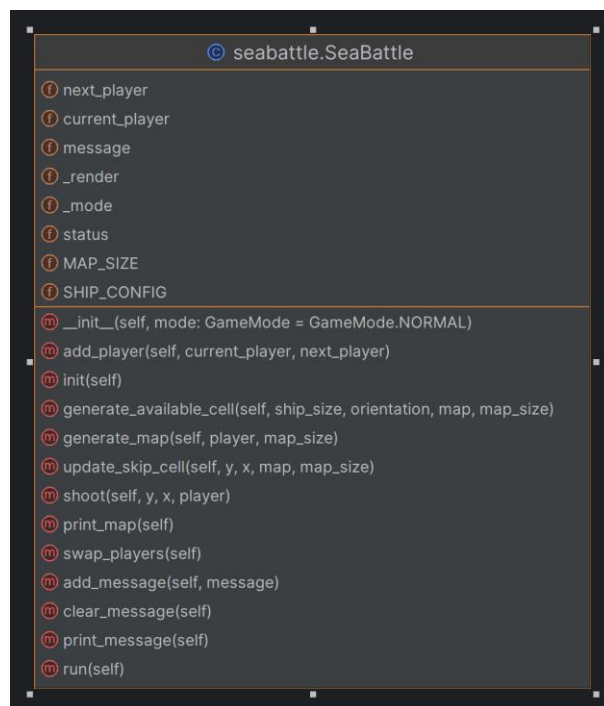


Рисунок 33. Диаграмма основного класса приложения SeaBattle

```

6 class GameMode(Enum):
7     # режим сложности
8     EASY = 1 # видны все детали - где расположены корабли противника, куда стрелял, что подбил
9     NORMAL = 2 # видны частичные детали - куда стрелял, что подбил
  
```

Рисунок 34. Класс GameMode, режимы игры

```

484 # запуск игры
485 if __name__ == '__main__':
486     game = SeaBattle(GameMode.EASY)
487     game.add_player(Man('Игрок'), Bot('Бот'))
488     game.init()
489     game.run()
490
  
```

Рисунок 35. Код запуска игры

Классы `Man` и `Bot` являются наследниками класса `Player` и отвечают за поведение игроков, человека и бота соответственно (см. Рисунок 36). Классы содержат информацию о количестве подбитых кораблей противника, список своих кораблей, имени игрока. Метод `get_coordinate()` отвечает за получение координат очередного выстрела, в случае с человеком это будет ввод из консоли, в случае с ботом – случайная ячейка на игровом поле. Класс бота можно модернизировать и реализовать более сложную стратегию, вместо хаотичных выстрелов по всему полю.

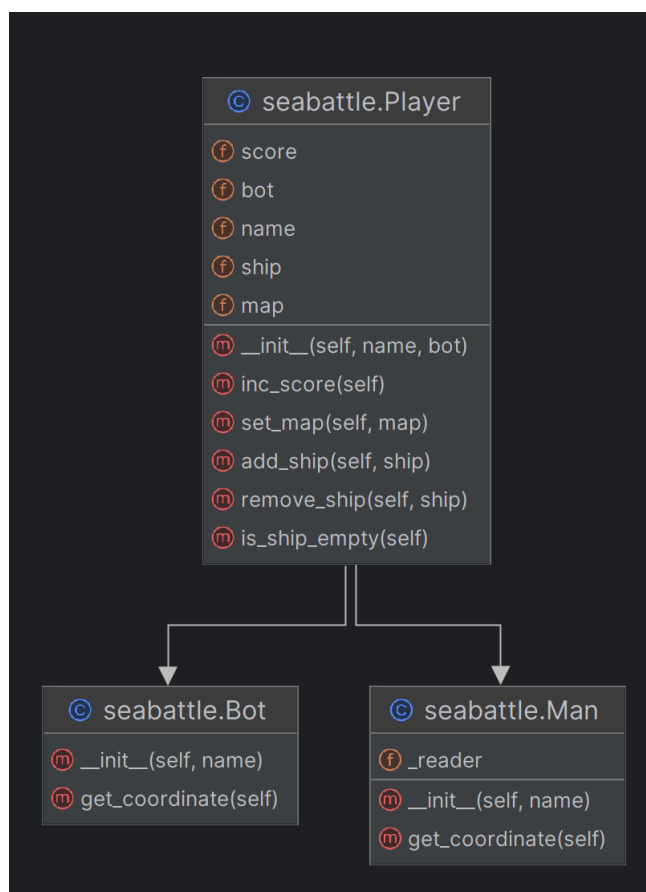


Рисунок 36. Диаграмма классов `Player`, `Bot`, `Man`

Классы `Cell` и `Ship` отвечают за содержание ячеек на игровом поле (см. Рисунок 37). Классы содержат информацию о координатах ячейки на игровом поле и её содержимом, которое описывается классом `ContentType` (см. Рисунок 38). Класс `Ship` содержит дополнительную информацию о корабле: количество палуб, ориентация корабля на карте (горизонтальная, вертикальная) и список палуб, описываемых классом `ShipUnit`.

Класс `ShipUnit` отвечает за палубы корабля, содержит информацию о координатах палубы на игровом поле и состоянии палубы: цела, подбита.

Так же проект содержит вспомогательные классы:

- `SeaRender` – отвечает за отрисовку в консоли игрового поля, текущего счёта и имени игроков.

- CellReader – отвечает за получение координат выстрела из консоли и преобразует их в координаты матрицы игрового поля.
- ShotResult – описывает результат выстрела по игровому полю: промахнулся, корабль подбит, корабль уничтожен, попытку нужно повторить т.к. в эту ячейку уже был выстрел ранее, либо нет смысла стрелять т.к. рядом расположен подбитый корабль.

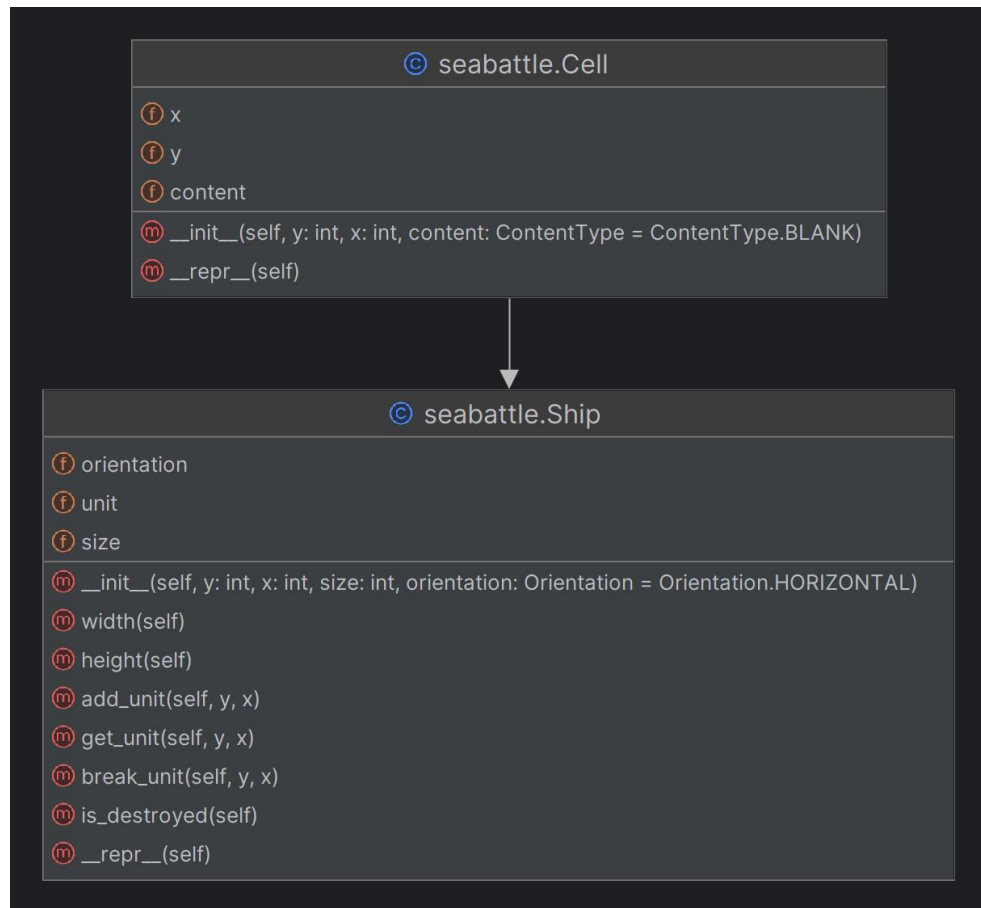


Рисунок 37. Диаграмма классов Cell и Ship

```

12 class ContentType(Enum):
13     # содержимое ячеек карты
14     BLANK = 1 # пустота, водная гладь
15     CRATER = 2 # сюда был выстрел, появился кратер
16     SHIP = 3 # корабль
17     SKIP = 4 # сюда нет смысла стрелять, ячейка рядом с подбитым кораблём
18
  
```

Рисунок 38. Код класса ContentType

2.4. Возможности по дальнейшему расширению функционала

Язык программирования Python имеет богатый арсенал подключаемых пакетов и модулей, которые позволяют значительно расширить функциональность проекта. Использование ООП в союзе с совершенствующимся языком, позволяют не останавливаться на достигнутом и дают нам огромное поле для творчества.

При разработке игры, появилось несколько идей по её дальнейшему улучшению:

- Перейти от консольного интерфейса к полноценному графическому. Для этого можно использовать различные библиотеки, например PyGame.
- Улучшить бота, добавив уровни сложности и полноценные тактики игры вплоть до использования ИИ и нейросетей. В этом могут помочь библиотеки TensorFlow и Keras.
- Добавить локальный мультиплеер по Bluetooth и глобальную многопользовательскую игру через сеть интернет. В этом могут помочь библиотека PyBluez и уже встроенная в дистрибутив The Python Standard Library.
- Создать игрового Telegram-бота, например с помощью библиотеки PyTelegramBotAPI.
- Создать полноценное мобильное приложение.

2.5. Тестирование работоспособности

Чтобы удостовериться, что наша игра работает правильно, т.е. соответствует всем описанным требованиям, воспользуемся следующими способами:

- Проведём альфа-тестирование – попросим своих друзей, знакомых и родственников поиграть в наше приложение и дать развёрнутую обратную связь.
- Проведём функциональное тестирование – составим чек-лист с основными требованиями и планомерно будем проверять работоспособность каждого пункта просто играя в игру и пытаясь смоделировать требуемую ситуацию. В этом нам может помочь специальный EASY режим, позволяющий просматривать корабли противника (см. Рисунок 31). Для более сложных случаев можем воспользоваться режимом отладки в IDE PyCharm. Останавливая работу программы в нужных точках и меняя значения переменных, можно смоделировать нужную ситуацию (см. Рисунок 39).

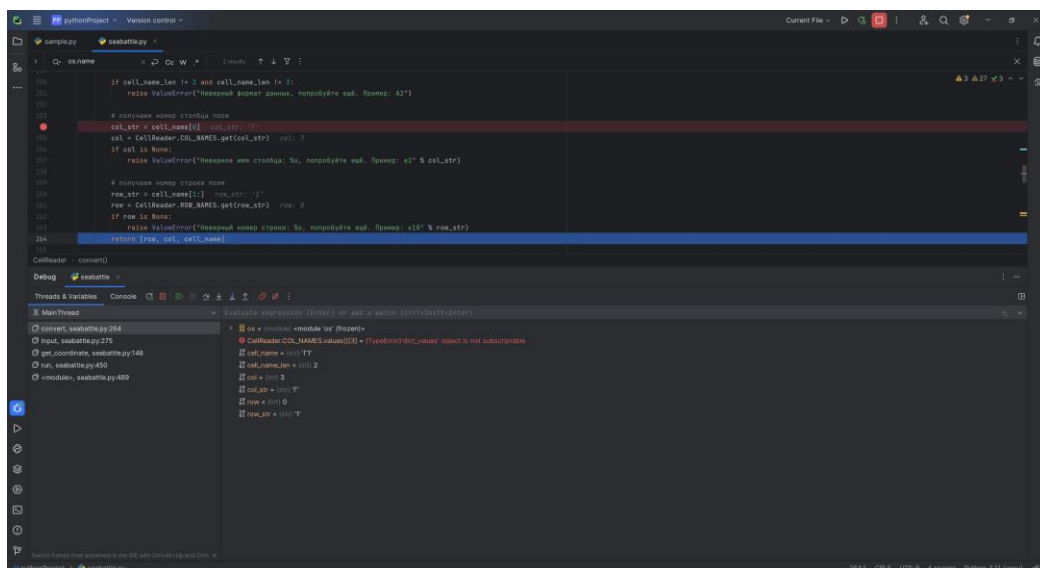


Рисунок 39. Режим отладки в IDE PyCharm

2.6. Размещение материалов в публичном репозитории

Чтобы дать доступ к созданным продуктам (исходники игры, руководство по созданию), воспользуемся популярным сервисом GitHub – крупнейший хостинг ИТ-проектов и их совместной разработки. Это поможет не только открыть доступ к изучению материалов для всех желающих, но и поспособствует созданию комьюнити по дальнейшей разработке приложения.

Разработанные продукты размещены в публичном репозитории по адресу <https://github.com/seabattle44/seabattle>

Чтобы привлечь внимание и быстро распространять наш адрес в соцсетях и мессенджерах, зашифруем его в QR-код, для этого воспользуемся бесплатным онлайн генератором QR кодов (см. Рисунок 40).



Рисунок 40. QR-код с адресом на публичный репозиторий

Список литературы

1. Доусон М. Програмируем на Python. 3-е изд. – СПб.: Питер, 2022. – 416 с. // Руководство написано опытным разработчиком и преподавателем, научит фундаментальным принципам программирования на примере создания простых игр. В книге приведено множество примеров кода, а в конце каждой главы найдётся проект полноценной игры.
2. Бэйдер Д. Знакомство с Python. – СПб.: Питер, 2023. – 512 с. // Книга построена по принципу 80/20: большую часть полезной информации можно усвоить, изучив несколько критически важных концепций. Освоив самые популярные команды и приемы, можно сразу сосредоточиться на решении реальных задач.
3. Седер Наоми. Python. Экспресс-курс. 3-е изд. – СПб.: Питер, 2021. – 480 с. // С помощью этой книги возможно быстро перейти от основ к управлению и структурам данных, чтобы создавать, тестировать и развертывать полноценные приложения. Автор рассказывает не только об основных особенностях языка Python, но и его объектно-ориентированных возможностях, которые появились в Python 3.
4. Моржакова М. (2021) Цифровые мозги в дефиците. ИТ-отрасль остро нуждается в кадрах // Официальный сайт Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации (<https://digital.gov.ru/ru/events/41208/>). Просмотрено: 16.04.2024.
5. Проект «Код будущего» (2024) Программирование для школьников, студентов колледжей и техникумов // Справочно-информационный интернет-портал РФ Госуслуги (<https://www.gosuslugi.ru/futurecode>). Просмотрено: 23.04.2024.
6. Index TIOBE (2024) // Индекс, оценивающий популярность языков программирования, на основе подсчёта результатов поисковых запросов в наиболее посещаемых порталах: Google, Blogger, Wikipedia, YouTube, Baidu, Yahoo!, Bing, Amazon и т.п. (<https://www.tiobe.com/tiobe-index/>). Просмотрено: 16.04.2024.
7. Stephens R. (2024) RedMonk Top 20 Languages Over Time: January 2024 // Рейтинг языков программирования на базе Amazon Web Services (<https://redmonk.com/rstephens/2024/03/08/top20-jan2024/>). Просмотрено: 16.04.2024.
8. Stack Overflow Annual Developer Survey (2023) // Рейтинг наиболее популярных технологий на базе результатов ежегодного опроса более 90 000 разработчиков от Stack Overflow (<https://survey.stackoverflow.co/2023/#technology>). Просмотрено: 16.04.2024.
9. Cass S. (2023) The Top Programming Languages 2023 // Ежегодный рейтинг языков программирования от института инженеров электротехники и электроники (IEEE

- Spectrum), который использует 11 критериев, включая популярность языка в поисковых запросах, число репозиторий, количество разработчиков, зарплаты специалистов, упоминания в вакансиях и учебных курсах, технологические тренды (<https://spectrum.ieee.org/the-top-programming-languages-2023>). Просмотрено: 16.04.2024.
10. The Python Tutorial (2024) // Официальная документация Python 3 (<https://docs.python.org/3/tutorial/index.html>). Просмотрено: 22.04.2024.