МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Балтийский федеральный университет имени Иммануила Канта»

ОНК «Институт высоких технологий»

ОТЧЁТ О ПРОХОЖДЕНИИ

УЧЕБНОЙ ТЕХНОЛОГИЧЕСКОЙ (ПРОЕКТНО-ТЕХНОЛОГИЧЕСКОЙ) ПРАКТИКИ

на базе Высшей школы компьютерных наук и искусственного интеллекта

Выполнил Третьяков Ричард Алексеевич

студент очной формы обучения 1 курса

направления подготовки 01.03.02 Прикладная математика и информатика

профиль обучения «Информатика и программирование»

Руководитель практики

Ст. преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Тарачков М.В.

г. Калининград 2024 г.

# ОГЛАВЛЕНИЕ

[**ВВЕДЕНИЕ 3**](#_Toc172103460)

[**ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ЗНАНИЯ 4**](#_Toc172103461)

[**Язык программирования Python и его особенности 4**](#_Toc172103462)

[**Объектно-ориентированное программирование 5**](#_Toc172103463)

[**Библиотека Pygame в Python 6**](#_Toc172103464)

[**Модуль random в Python 7**](#_Toc172103465)

[**ГЛАВА 2. ЗАДАНИЕ НА ПРАКТИКУ 8**](#_Toc172103466)

[**Задача игра «Snake» 8**](#_Toc172103467)

[**Реализация 8**](#_Toc172103468)

[**ГЛАВА 3. ВЫПОЛНЕНИЕ ЗАДАНИЯ НА ПРАКТИКУ 9**](#_Toc172103469)

[**Пункт 1. 9**](#_Toc172103470)

[**Пункт 2. 9**](#_Toc172103471)

[**Пункт 3. 10**](#_Toc172103472)

[**Пункт 4. 13**](#_Toc172103473)

[**Пункт 5. 14**](#_Toc172103474)

[**ЗАКЛЮЧЕНИЕ 15**](#_Toc172103479)

[**СПИСОК ЛИТЕРАТУРЫ 16**](#_Toc172103480)

[**ПРИЛОЖЕНИЕ 1 17**](#_Toc172103481)

# ВВЕДЕНИЕ

Вид практики – Учебная технологическая (проектно-технологическая) практика (далее Учебная практика).

Цель учебной практики: получение первичных профессиональных умений навыков.

Задачи учебной практики:

1. Закрепление и углубление теоретических знаний в области информационных технологий;
2. Приобретение и развитие первичных профессиональных навыков и

умений в области прикладной математики и информатики.

# ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ЗНАНИЯ

# Язык программирования Python и его особенности

Python – это высокоуровневый язык программирования, который был разработан в конце 1980-х годов. Его разработчик, Гвидо ван Россум, вложил в основу языка простоту и читабельность кода, что позволяет использовать Python для быстрой и эффективной разработки. Много популярных веб-сайтов, компьютерных игр и программ, написанных на Python.

Особенности Python:

* Простота и читаемость кода: Синтаксис Python интуитивно понятен и похож на английский язык, что делает его легким для изучения и использования.
* Интерпретируемость: Python является интерпретируемым языком, что означает, что код выполняется строчка за строчкой, что упрощает тестирование и отладку.
* Большая стандартная библиотека: Python имеет обширную стандартную библиотеку, которая предоставляет множество полезных функций и модулей для выполнения разнообразных задач.
* Кроссплатформенность: Python работает на различных операционных системах, таких как Windows, macOS и Linux.
* Поддержка различных парадигм программирования: Python поддерживает процедурное, объектно-ориентированное и функциональное программирование.

# Объектно-ориентированное программирование

Объектно-ориентированное программирование — это парадигма программирования, основанная на концепции "объектов", которые могут содержать данные и методы для обработки этих данных. В контексте игр, объекты могут представлять игровые сущности, такие как персонажи, враги, предметы и другие элементы.

Основные принципы ООП:

* Инкапсуляция: Объединение данных и методов, работающих с этими данными, в один объект. Это помогает скрыть внутреннюю реализацию и защитить данные от некорректного использования.
* Наследование: Возможность создавать новые классы на основе существующих, наследуя их свойства и методы. Это способствует повторному использованию кода.
* Полиморфизм: Способность использовать объекты разных классов через общий интерфейс, что позволяет создавать гибкие и расширяемые системы.
* Абстракция: Выделение значимых характеристик объекта и игнорирование незначительных деталей. Это упрощает понимание и использование объектов.

Программирование игр относится к объектно-ориентированному программированию (ООП), однако также часто использует другие концепции, такие как управление состоянием, события и асинхронное программирование для создания интерактивных и отзывчивых приложений.

# Библиотека Pygame в Python

**Pygame** — это библиотека для разработки игр на языке программирования Python. Она предоставляет функциональные возможности для работы с графикой, звуком и вводом от пользователя, что делает её популярным выбором для создания 2D-игр и мультимедийных приложений.

Основные свойства Pygame:

* Кроссплатформенность: Pygame работает на различных операционных системах, включая Windows, macOS и Linux.
* Простота использования: Pygame имеет простой и интуитивно понятный API, что делает его доступным для начинающих разработчиков.
* Графика: Поддержка работы с изображениями, спрайтами, анимацией и текстом.

Основными виджетами Pygame являются:

* pygame.display – управление окном
* pygame.draw – рисование фигур
* pygame.image – загрузка изображений
* pygame.sprite – работа со спрайтами
* pygame.mixer – работа с музыкой
* pygame.font – работа со шрифтами
* pygame.event – обработка событий
* pygame.time – работа с таймером
* pygame.key – обработка ввода с клавиатуры

# Модуль random в Python

random — это стандартный модуль Python для генерации случайных чисел и выполнения операций с использованием случайности. Модуль предоставляет функции для генерации случайных чисел, выбора случайных элементов из последовательностей и перетасовки списков.

Основные функции модуля random:

* random() - Возвращает случайное число с плавающей точкой в диапазоне [0.0, 1.0).
* randint(a, b) - Возвращает случайное целое число N, такое что a<=N<= b.
* choice(seq) - Возвращает случайный элемент из непустой последовательности seq.
* shuffle(seq) - Перемешивает элементы последовательности seq на месте.
* sample(seq, k) - Возвращает список из k уникальных элементов, выбранных из последовательности seq.

Модуль random полезен в игровом программировании для создания случайных событий, генерации уровней, размещения объектов и других задач, связанных с случайностью.

# ГЛАВА 2. ЗАДАНИЕ НА ПРАКТИКУ

# Задача игра «Snake»

Реализовать игру «Snake», в функции которой будет входить:

* Передвижение по полю
* Сбор оранжевых яблок
* Учёт количества собранных яблок в отдельной плашке с названием «Cчёт»
* Завершение игры при столкновении со стенкой или при съедении змейки самой себя

# Реализация

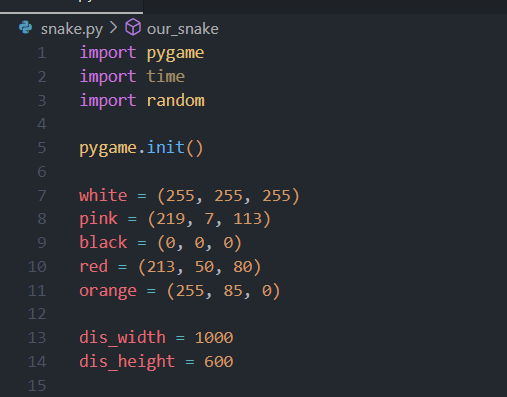
Разобьём задание на пункты для последовательного выполнения:

1. Импорт библиотек и инициализация Pygame
2. Определение вспомогательных функций
3. Создание основного цикла, обработка событий и состояние игры
4. Обновление положений змейки и проверка столкновений
5. Проверка на сбор еды и увеличение длины змейки

# ГЛАВА 3. ВЫПОЛНЕНИЕ ЗАДАНИЯ НА ПРАКТИКУ

# Пункт 1.

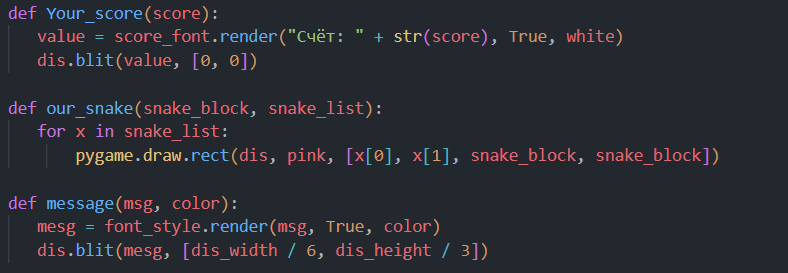
Первым делом были импортированы функции библиотеки pygame, time и модуль random для дальнейшей работы. Затем были заданы значения размера окна и установлены цвета для различных элементов игры.



**Пункт 2.**

В этом пункте идет определение вспомогательных функций, таких как:

1. your\_score – отображает текущий счет на экране
2. our\_snake – отрисовка змейки на экране
3. message – отображение сообщений на экране



# 

# Пункт 3.

Далее создаем основной цикл игры. Основной игровой цикл отвечает за управление всеми аспектами игры, включая инициализацию переменных, обработку событий, обновление состояния игры и отрисовку элементов на экране.

Начинается все с инициализации переменных в функции gameLoop().

- Переменные:

- game\_over: Логическая переменная, указывающая, завершена ли игра. Изначально установлена в False.

- game\_close: Логическая переменная, указывающая, находится ли игра в состоянии завершения (например, когда змейка столкнулась с границей или с собой). Изначально установлена в False.

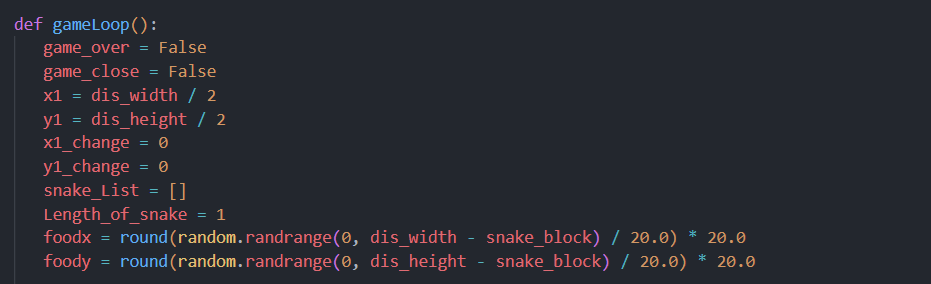
- x1 и y1: Начальные координаты головы змейки, установленные в центр экрана.

- x1\_change и y1\_change: Переменные для изменения координат головы змейки, которые будут использоваться для управления движением.

- snake\_List: Список для хранения координат всех сегментов змейки.

- Length\_of\_snake: Начальная длина змейки, установленная в 1.

- foodx и foody: Координаты еды, которые генерируются случайным образом в пределах игрового поля.



Следующим этапом идет создание основного цикла игры.

Основной цикл игры состоит из двух вложенных циклов while. Внешний цикл while not game\_over продолжает выполнение игры до тех пор, пока переменная game\_over не станет True.

- Вложенный цикл while game\_close == True:

- Этот цикл выполняется, когда игра находится в состоянии завершения (game\_close равно True).

- dis.fill(black): Очищает экран, заполняя его черным цветом.

- message("Вы проиграли! Нажмите Q для выхода или C для повторной игры", red): Отображает сообщение о проигрыше.

- Your\_score(Length\_of\_snake - 1): Отображает текущий счет.

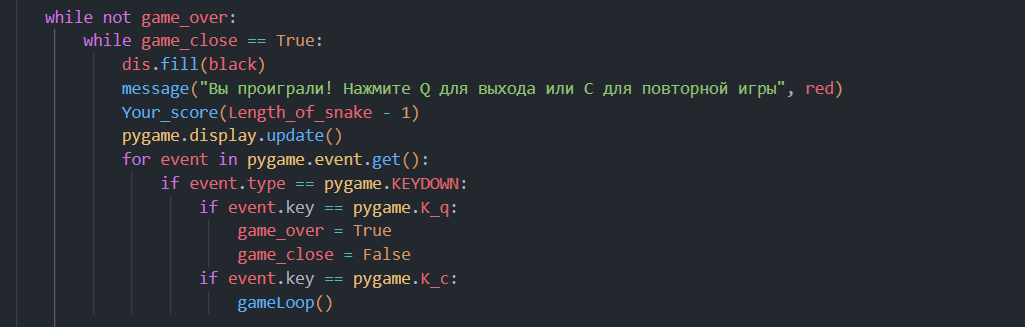
- pygame.display.update(): Обновляет экран, чтобы отобразить изменения.

- Обработка событий:

- Проходит по всем событиям, происходящим в Pygame.

- Если нажата клавиша Q, игра завершается (game\_over устанавливается в True).

- Если нажата клавиша C, игра перезапускается (вызывается gameLoop()).



Заключительным же этапом этого пункта является обработка событий и управление змейкой.

Обработка событий:

Проходит по всем событиям, происходящим в Pygame.

- Если событие типа pygame.QUIT, игра завершается (game\_over устанавливается в True).

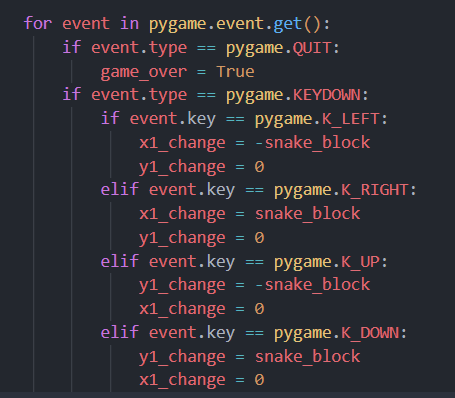
- Если событие типа pygame.KEYDOWN, проверяется, какая клавиша была нажата:

- Клавиша влево (K\_LEFT): Изменяет направление движения змейки влево.

- Клавиша вправо (K\_RIGHT): Изменяет направление движения змейки вправо.

- Клавиша вверх (K\_UP): Изменяет направление движения змейки вверх.

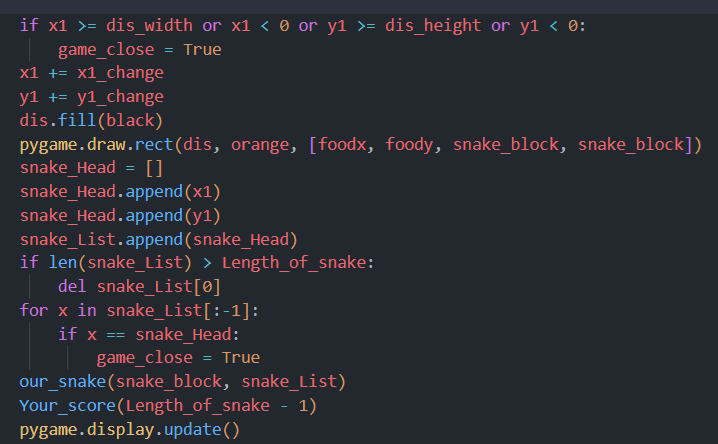
- Клавиша вниз (K\_DOWN): Изменяет направление движения змейки вниз.



# Пункт 4.

Этот пункт отвечает за обновление координат змейки, отрисовку её на экране и проверку на столкновения:

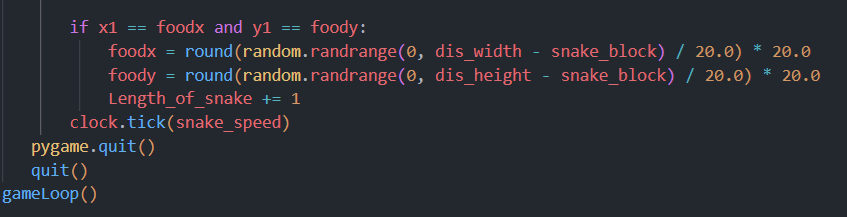
1. Проверка столкновений с границами
2. Обновление координат головы змейки
3. Очистка экрана и отрисовка еды
4. Обновление списка сегментов змейки
5. Удаление лишних сегментов
6. Проверка столкновений с собственным телом
7. Отрисовка змейки и обновление счета



# Пункт 5.

Заключительным же пунктом будет являться проверка сбора еды и увеличение длины змейки. В нее входят:

1. Проверка на сбор еды
2. Генерация новых координат для еды
3. Увеличение длины змейки
4. Управление временем
5. Завершение и перезапуск игры



# ЗАКЛЮЧЕНИЕ

В ходе практики была изучена концепция объектно-ориентированного программирования на языке Python, была изучена такая библиотека как **pygame**, был изучен модуль random и все его функции.

Задачи были направлены на закрепление изученного теоретического материала. Особое внимание уделялось работе со встроенной библиотекой виджетов pygame и объектно-ориентированному программированию, его принципам и особенностям.

Предлагалось создать классическую игру «Snake» с набором очков, удлинением хвоста змеи и перемещением по ограниченному полу действия игры и её компонентов.

В результате практики были усовершенствованы мои компетенции, закреплены теоретические знания и практические навыки.

В ходе учебной практики я научился: выполнять поставленную задачу, следуя условию; разделять главную задачу на подпункты и определять приоритетность этих пунктов; изменять готовое решение по мере нахождения ошибок, искать альтернативные пути решения задач.

Я изучил библиотеку **pygame** (расстановка виджетов: изменение их характеристик, приведение их в движение, взаимодействие друг с другом), модуль random (работа со случайными числами, генератор случайностей), принципы ООП.

По мере прохождения учебно-технологической практики я выполнил пять задач, в которых использовал полученные знания и закрепил навыки работы с ЯП Python.

В течение практики все задачи были выполнены, а цели достигнуты.

# СПИСОК ЛИТЕРАТУРЫ

1. Документация синтаксиса ЯП Python - <https://docs.python.org>
2. Документация библиотеки pygame - <https://docs.python.org/3/library/pygame.html>
3. Документация метода random - <https://docs.python.org/3/library/random.html>
4. Объяснение основных методов работы с ООП в Python - <https://cmp.phys.msu.su/sites/default/files/%D0%9E%D0%9E%D0%9F_%D0%BD%D0%B0_Python_%D0%A3%D1%87%D0%B5%D0%B1%D0%BD%D0%BE%D0%B5%20%D0%BF%D0%BE%D1%81%D0%BE%D0%B1%D0%B8%D0%B5_var7.pdf>

# ПРИЛОЖЕНИЕ 1

Код проекта, отчет, дневник - [https://github.com/heelmees/practice](https://github.com/heelmees/practicej)